Fusing Visual Odometry and Depth Completion

by

Guilherme Venturelli Cavalheiro

B.S., Aeronautics Institute of Technology (2016)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2019

Author ..... Signature redacted .....
Department of Aeronautics and Astronautics
Jan 31, 2019

Signature redacted

Certified by.....
Sertac Karaman
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Signature redacted

Accepted by .....
Sertac Karaman
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Fusing Visual Odometry and Depth Completion

by

Guilherme Venturelli Cavalheiro

Submitted to the Department of Aeronautics and Astronautics
on Jan 31, 2019, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Recent advances in technology indicate that autonomous vehicles and self-driving cats in particular may become commonplace in the near future. This thesis contributes to that scenario by studying the problem of depth perception based on sequences of camera images. We start by presenting a sensor fusion framework that achieves state-of-the-art performance when completing depth from sparse LiDAR measurements and a camera. Then, we study how the system performs under a variety of modifications of the sparse input until we ultimately replace LiDAR measurements with triangulations from a typical sparse visual odometry pipeline. We are then able to achieve a small improvement over the single image baseline and chart guidelines to assist in designing a system with even more substantial gains.

Thesis Supervisor: Sertac Karaman
Title: Associate Professor of Aeronautics and Astronautics

Acknowledgments

I would like to thank Fangchang Ma and Varun Murali for collaborating with me in projects that directly contributed to this thesis. I am also very grateful for the continued support and advice that Sertac Karaman has given me, especially given my tendency to propose crazy ideas. A special thanks to all my professors and the staff for providing a great learning experience. I would also like to thank my coworkers and colleagues that helped my in some way with whom I shared great moments: Amado Antonini, Dave McCoy, Ezra Tal, Gilhyun Ryou, Igor Spasojevic, Jasper Arnenberg, Jin Gao, John Aleman, Sebastian Quilter, Thomas Sayre-McCord, Murat Bronz, Oscar Mickelin, Winter Guerra. Last but not least, a huge thanks to my family for always being in my mind and always supporting me wherever I go.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

1.1.1 The human baseline

The human body is equipped with sensory organs that are similar to some common robotic sensors: the eyes work in a similar manner to a stereo camera setup, the vestibular system is not unlike an IMU and both machines and humans are able to detect sounds waves [31]. Yet, even with the availability of radar, sonar, LiDAR and other types of sensors, modern robotic solutions still fall short of human performance when it comes to tasks like autonomous driving. Part of the gap can be attributed to planning and decision making challenges, but perception is arguably the greatest roadblock. This thesis aims at helping close the perception gap by studying how temporal visual information, which is well studied in the context of classical computer vision, can be integrated with recent advances in depth perception provided by neural networks.

A popular approach in self-driving technologies involves using laser based sensors to measure distances to the environment. These sensors provide accurate measurements at thousands of points, but at significant costs, especially for the high resolution variants (see Table 1.1). However, they are not fundamentally necessary as humans are able to drive without lasers or similar sensors.

One of the mechanisms used by the human body to perceive depth information is through its binocular vision, which can be comparably achieved by using multiple cameras in a stereo setup. Another factor, prior knowledge about the environment, is not as easily integrated, but plays an important role in estimating scales. For example, a car is expected to measure a few meters in length, while buildings may assume several meters in height with a somewhat constant expectation of a couple of meters per level. Since that knowledge is intractably vast to be formulated analytically, neural networks and other learning approaches are often used to try and incorporate some prior information.

Although humans are remarkably capable in some tasks, one might wonder the value in trying to replicate their capabilities with robotics system by limiting the types of sensors used when the availability of other devices could allow for even better performance. Indeed, additional sensors should be used whenever beneficial, but humans represent a lower bound on the potential of a solution and indicate room for improvement, even when adopting more complex approaches. Furthermore, this baseline, however capable, is still identified as the critical reason for 94% of the car crashes in the US [47], so any improvement provided by robotics systems would translate into fewer accidents, not to mentions quality of life and economic benefits.

1.1.2 Towards tighter integration

From a conceptual standpoint, there is more information in a sequence of images than a single image, so it's to be expected that using the former would yield better results than using the latter. However, most research in depth estimation is focused on single image analysis, which is valid on itself as a fundamental problem, but is ultimately limited. The works that do combine sequential images and depth estimation often do so for the purpose of unsupervised training, to provide scale or initialization to monocular odometry, rarely to enhance how distances are estimated.

Another popular source of depth information is stereo vision, which is a valuable information in itself, but we argue that it's not a complete replacement to temporally related image data. Besides increased hardware costs, the range of stereo is limited by

a fixed setup baseline distance, while landmark triangulation can work with a much larger baselines as long as there is sufficient movement and successful tracking.

1.1.3 Affordability

Equipping a vehicle with LiDAR , however beneficial, can be very costly. Table 1.1 [59] compares different products offered by Velodyne, which are commonly used in robotics experiments, including a dataset used in this work. New technologies and savings by scale may reduce the actual costs in a production environment, but it's likely that the traditional bulky and high-resolution rotating scanner will be replaced by alternate solutions (e.g. solid state LiDAR) that have reduced sensing capabilities and thus the necessity of sensor fusion becomes even greater.

Table 1.1: Comparison of different LiDAR products offered by Velodyne.

Product	Scanlines	Power (W)	Price ($)
HDL-64	64	60	75,000
HDL-32	32	12	30,000
VLP-16	16	8	8,000

1.2 Overview

Chapter 2 is dedicated to studying the problem of estimating depth from colored images and in potential combination with other sensors such as LiDAR and stereo cameras with the help of neural networks.

The third chapter presents aspects of visual odometry and robot localization.

The contents from the previous chapter is integrated in chapter 4, which aims at using the neural network based depth completion in tandem with localization and mapping.

Future directions and concluding remarks are introduced in chapter 5.

Chapter 2

Neural networks for autonomy and depth perception

2.1 Introduction

In this chapter we are going to discuss some of uses of neural networks for depth perception in the context of autonomous vehicles. This problem fits into a larger context of navigation as estimating distances is intimately linked to the processing of mapping and localization, but it's also associated with other tasks such as semantic labeling [28].

After providing a basic understanding of neural networks, we will explore related articles in the literature and then present the base architecture used for other chapter of this work. This neural network and related experiments were published in [32] and part of those are reproduced in this thesis. The first set of results is related to the network itself, which achieved state-of-the-art results on a depth completion challenge, while other discussions involve ablation studies. Finally, we explore the effects of having different distributions of sparse inputs: by reducing the number of scanlines in the LiDAR sensor and by using random sampling, an analysis strongly motivated by practical interests of self-driving cars.

2.2 Background

2.2.1 Neural networks

The study of artificial neural networks dates back to 1943 [34] and since then their popularity in academia has oscillated considerably. In 2012, the works of [24] have attracted attention back to the field as they were able to surpass ever other method in the ImageNet ILSRVC-2012 competition by a large margin using neural networks. More precisely, they were able to reduce the top-5 error rate for an image classification task from 26.2% to 15.3%. In the following years, a considerable amount of research effort was dedicated to such techniques and currently neural networks achieve impressive results in other domains such as natural language processing and computer games [35].

For the purpose of this dissertation, we are going to focus on computer vision applications and on modern considerations. An artificial neural network (sometimes simply referred to as "neural network") is an arrangement of mathematical operations centered on the notion of a "neuron", which is often a simple scalar function of a single variable ($f : \mathcal{R} \rightarrow \mathcal{R}$). Layers are a combination of a linear transformations to a multi-dimensional signal followed by an element-wise application of the neuron nonlinearity. Mathematically, given an input signal $x_i \in \mathcal{R}^m$, a weight matrix $M_i \in \mathcal{R}x\mathcal{R}^n$ and bias vector $b \in \mathcal{R}^n$ defining a linear transformation, the output y_{i+i} of the layer i is given by

$$z_i = W_i x_i + b_i \tag{2.1}$$

$$y_i = f(z_i) \tag{2.2}$$

where some common choices for f are the rectified linear unit ($f(x) = x$ if $x > 0$ and $f(x) = 0$ otherwise, also known as ReLU), the sigmoid ($f(x) = \frac{1}{1+e^{-x}}$).

The weights and biases are often determined by stochastic gradient descent optimization (or variants such as ADAM [23]), or by evolutionary strategies [39][49], although the latter is more common in reinforcement learning settings and when

differentiability is an issue, while the former is predominant for supervised image processing problems. For many tasks, the best performing networks require several layers and a considerable amount of data, so the learning of weights is of central importance. The ImageNet dataset, for example, a common source of data for image classification, contains millions of high-resolution images [8] and some popular network architectures have more than a hundred layers [18]. These extra considerations motivate the use of the term "deep learning" and "deep neural networks".

In the most simple situation, the input x_{i+1} to the next layer is y_i, but it's sometimes advantageous to combine other operations to compose this input. A very popular construct is the residual connection [18], originally given by replacing z_{i+2} with $z_{i+2} + x_2$ when the dimensions are appropriate. Other works [19] suggest variations, like settings y_{i+2} as $y_{i+2} + x_i$ instead in order to preserve a path of identity mapping. A proper understanding of when and why these techniques are helpful is still a topic of active research, but common hypothesis suggests that these connections provide better starting points and allow for easier conditions for learning as they provide shortcuts for information and gradient information to flow.

Another important tool to facilitate learning is given by batch normalization [20], which involves normalizing intermediary values along the network. During training, this is done by using batch statistics (that is, mean and standard deviation information along groups of training samples), while during evaluation fixed learned parameters are used. This difference in behavior should be negligible when batch sizes are large, but can be considerable otherwise. When only small batch sizes are possible (which is the case of this work, considering the necessity for high image resolution), one can try alternatives like instance normalization [52]. However, even though these techniques became widely popular and demonstrated effectiveness under a variety of situations, the reason for their success is still a topic of research [40].

For tasks related to image processing, the weights used in Equation 2.1 usually take a special form given by spatial convolutions with small kernels, a strategy motivated by the fact that in natural images information that is close in the spatial domain is correlated. Mathematically, given a $3D$ tensor X with dimensions (h, w, c) and the

$4D$ convolution kernel K of dimensions (k_x, k_y, c, d), the output Y of the convolution is a $3D$ tensor given by (h, w, d)

$$Y(i,j,k) = \sum_{d_x,\ d_y,\ d_c} K(d_x, d_y, d_c) X(i + d_x s_x, j + d_y s_y, d_c, k) \tag{2.3}$$

where the values of X are usually extended beyond its original domain by means of padding and some sort of spatial shifting is also applied, and s_x and s_y represent strides. Quite often, the kernel is rectangular ($k_x = k_y = k_l$) and of odd length, such that the shifting operation becomes simplified to the center of k_l. Other common practices involves small kernel sizes (e.g. $k_l = 3$) and a number of filters that is a power of 2, both practices motivated by practical considerations.

A complementary operation is the deconvolution or more aptly named the transposed convolution. A regular convolution will result in reductions of the spatial dimensions when the strides are larger than unity, so the transposed convolution is often used to expand and up-sample instead. This is achieved by expanding the original data into a larger a matrix and filling intermediary and border values with 0. Note that with appropriate parameters and weights one can reproduce bilinear up-sampling and that's a common initialization procedure. The interested reader is referred to [10] for more details.

2.2.2 depth perception

Computing distances between objects is an important step for autonomous vehicles or for other robotics applications as it can potentially be beneficial to localization, collision avoidance, object detection and related tasks. Depth measurements fall into such category as they capture the distance from a sensor to other objects.

Some sensors are able compute distances directly, but no solution is complete enough to cover some common requirements. Structured light sensors require an active energy source and have limited range; LiDAR sensor are bulky, expensive and only produce sparse measurements; stereo vision requires disparity, is limited

in range and is not very accurate. Consequently, it's common to fuse measurements from different sensor modalities to complement their weaknesses. Besides these direct measurements, one can also extract information indirectly from other devices, like cameras.

The problem of inferring depth from a single camera image is know as depth estimation. That's a challenging problem as it's ill-posed and in general a camera image could have an arbitrary depth associated with it. However, in practice we are mostly interested with the types of images one finds the real world and thus some sort for regularity is assumed: for instance, we expect a cars to have dimensions close to a few meters and assume certain specific shape. This allows the use of neural networks or other techniques in order to attempt learning these regularities.

A similar problem is the so called depth completion, which is composed of interpolating depth measurements in a dense setting given sparse information. Sometimes the task is complemented with camera images, which is a reasonable approach since in practice many robotics system are already equipped with cameras.

Common choices for studying this problem include the NYU Depth dataset [46] and the KITTI depth completion and depth prediction dataset [51]. The first one is composed of indoors scenes with associated camera and structured light sensor measurements, while the second one contains a multi-camera setup, a 64 scan line LiDAR and the ground truth is computed by fusing information at several timestamps. Other options include the indoor 3D scene reconstructions provided by ScanNet [7] or outdoors scenes available in the Make3D dataset [42].

2.3 Related work

Depth prediction, the problem of estimating depth from a single image, started with works such as [41], that used handcrafted features and Markov Random Fields. More recent works make use of deep learning techniques [11] and also explore unsupervised learning in order to avoid costly dataset generation procedures [63]. .

Depth completion is a more generic denomination that usually involves completing

missing entries in a depth map, but we are interested in the problem of recovering depth in the presence of highly sparse samples and a camera image. Works like [33], for example, apply neural networks to complete sparse depth measurements on both on outdoor and indoor situations, while [] encodes images with wavelets and contourlets in order to generate depth map reconstructions.

2.4 Proposed method

2.4.1 Architecture

We present a neural network architecture to tackle to problem of depth completion under two variants: using only a set of sparse depth measurements or also using a color image. The proposed solution is a fully convolutional neural network [30] illustrated in Figure 2-1, is inspired by [37] and is an evolution of [33]. It can be decomposed into an encoder and a decoder part, where the first section of the encoder merges different sensor modalities (when they are available). This fusion is accomplished by an independent set of convolutions prior to a filter concatenation, an approach that was adopted given that both inputs have widely different distribution. Following that, a set of residual blocks is applied using the structure of a ResNet-34 [18] with 64 filters in the first one, which allows the potential use of pretrained parameters. Afterwards, a sequence of transposed convolutions is applied to recover spatial resolution in conjunction with skip connections to allow for the flow of finer details from earlier parts. Finally, the filters are collapsed into a single channel by 1×1 convolutions to generate predictions. During training, dropout can be applied before that layer and during inference a post processing is applied to clamp distances above a minimal threshold. Following standard practices, we use ReLU nonlinearities and batch normalization.

The described architecture is slightly modified when only the sparse depth is available: the operations exclusive to the camera input are removed and the number of filters of the other branch are adjusted accordingly. We also reduce the number of

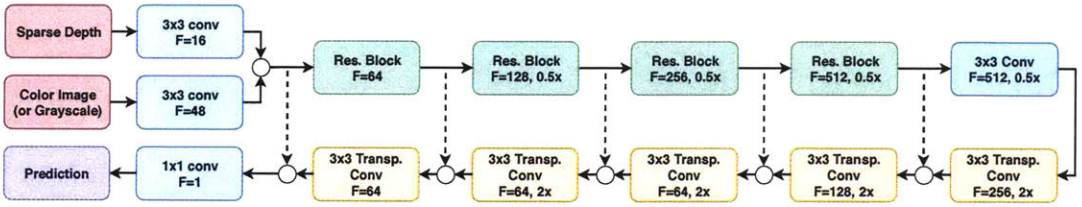


Figure 2-1: Our deep regression network for depth completion, with both sparse depth and RGB as input. Skip connections are denoted by dashed lines and circles represent concatenation of channels.

channels by a half, as explored in the ablation study in subsection 2.5.2.

2.4.2 training strategy

We train the network in a supervised fashion using the L_2 loss over all points for which there is a ground truth. In other words, the it's defined by considering all valid points of the image together as opposed to an alternative based on the average of the error per a single image. Explicitly ,let B be $B = 1, 2, \ldots, b$ with b the batch size and $I = I^1, I^2, \ldots, I^b$ the set of valid points for each image, we have the used L_2 loss as

$$L_2^{points} = \frac{1}{\sum_{k \in B} |I^k|} \sum_{k \in B} \sum_{i \in I^k} \left(y_i^{pred} - y_i^{true} \right)^2 \tag{2.4}$$

while another reasonable alternative would be

$$L_2^{image} = \frac{1}{|B|} \sum_{k \in B} \frac{1}{|I^k|} \sum_{i \in I^k} \left(y_i^{pred} - y_i^{true} \right)^2 \tag{2.5}$$

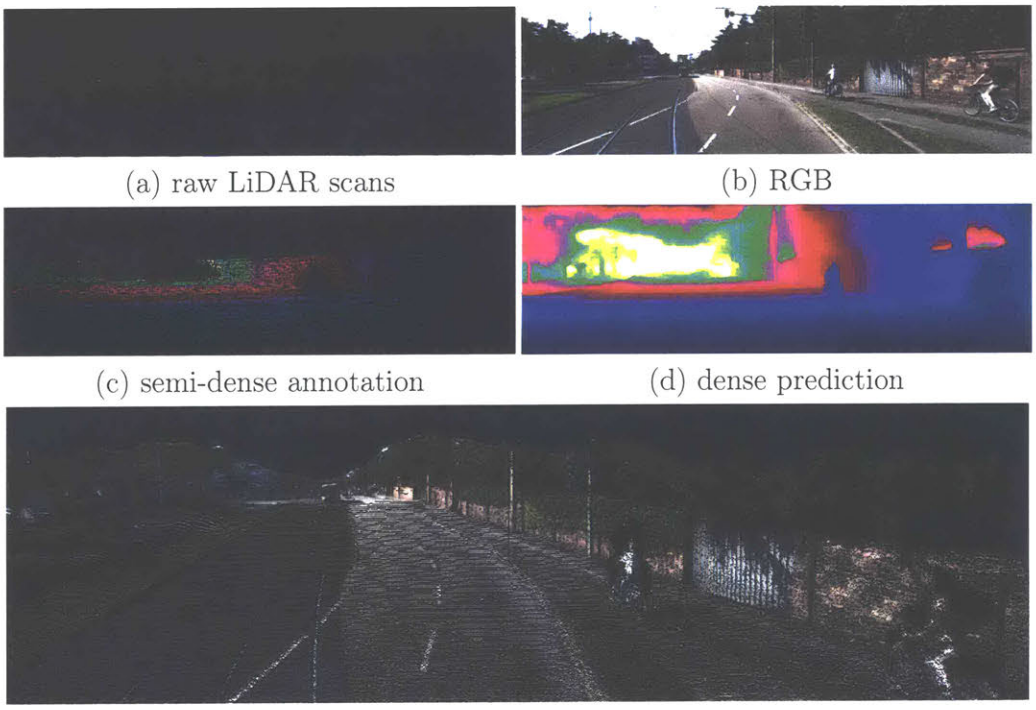
The difference is subtle, but noticeable in terms of performance. Conceptually, one could argue that each labeled point holds the same amount of information regardless of the amount of valid sample in their respective image.

We briefly mention that he proposed solution is also effective in the absence of ground truth by using self-supervision, as is explored in [32]. Namely, the photometric loss between a pair of adjacent images projected into the same frame can be used as a training signal and, if available, LiDAR measurements also provide additional helpful information if one takes it as a ground truth (despite being sparser and potentially

incorrect at some points due to reflection and other sensor issues). Additionally, smoothness losses can also be considered.

2.5 Experiments

We evaluate the proposed method mainly on the KITTI dataset, which has an associated depth completion challenge that allows for a rigorous comparison between different methods. An illustration of input, ground truth and predictions is given in Figure 2-2. There are 85897 images from 138 sequences in the training set, 6852 from 13 sequences in the validation (but we use a provided selection of 1000 images from that set) and another 1000 test images whose associated ground truth is not publicly available, but who can be evaluated in an external server a limited number of times. In consequence, for additional experiments we freeze hyperparameters and use the validation set as reference.



(a) raw LiDAR scans (b) RGB

(c) semi-dense annotation (d) dense prediction

(e) dense prediction (as point cloud)

Figure 2-2: Illustration of the developed deep regressional network for *depth completion.*

The method was also evaluated on the NYU dataset [46] for completeness and was shown to also perform adequately, as presented inAppendix A. This is done to assess the generality of the method as we argue that the KITTI dataset is a superior choice, at least for the interest of this work. Firstly, the labeling procedure of the outdoor dataset involves outlier rejection and sensor fusion, while the indoor one uses raw measurements. Secondly, there is no consistent test procedure to allow for a rigorous comparison between different methods, as authors use different resolutions, number of points and the stochastic nature of random sampling causes inputs to be different. Finally, the maximum range of 5m for the Kinect sensor is not respected as measurements exceeding 9.5m are used, while a rigorous analysis of the error suggests it grows quadratically and recommends the use of 3m as a maximum range for mapping applications [22].

Training is performed in a DGX-1 with a batch size of 8 and using the ADAM optimizer [23] for 20 epochs, a process that takes less than a day. The learning rate starts at $1e^{-5}$ and is decayed by a factor of 10 every 5 epochs. Random horizontal flipping is used and in the presence of camera images a small color jitter is added. Additionally, the topmost regions are cropped out (since they practically do not contains points with know or valid depths) to reach the final resolution of 1216×352, which is the same as the one used for the validation and test sets.

2.5.1 Depth completion challenge

The results of the proposed method with and without the help of a color image (respectively indicated as `d` and `RGBd`) are displayed in Table 2.1 (at time of first related publication) and in Table 2.2 (most recent results). Further illustration is provided in Appendix A. At the time of submission and for more than 6 months afterwards, our neural network achieved the top result in the ranking, including unpublished results, thus reaching state-of-the-art and demonstrating its effectiveness.

As evaluation metric the root mean square error (RMSE) is used as ranking metric for the competition and is also our main criteria for comparison. In this case, the error is taken per image and not considering the whole set of points, unlike the L_2

Table 2.1: Comparison against state-of-the-art algorithms on the test set at time of first publication.

Method	Input	RMSE [mm]	MAE [mm]	iRMSE [1/km]	iMAE [1/km]
NadarayaW [51]	d	1852.60	416.77	6.34	1.84
SparseConvs [51]	d	1601.33	481.27	4.94	1.78
ADNN [6]	d	1325.37	439.48	59.39	3.19
IP-Basic [25]	d	1288.46	302.60	3.78	**1.29**
NConv-CNN [13]	d	1268.22	360.28	4.67	1.52
NN+CNN2 [51]	d	1208.87	317.76	12.80	1.43
Ours-d	d	**954.36**	**288.64**	**3.21**	1.35
SGDU [44]	RGBd	2312.57	605.47	7.38	2.05
Ours-RGBd	RGBd	**814.73**	**249.95**	**2.80**	**1.21**

Table 2.2: Updated comparison against state-of-the-art algorithms on the test set.

Method	Input	RMSE [mm]	MAE [mm]	iRMSE [1/km]	iMAE [1/km]
Spade-sD [21]	d	1035.29	248.32	2.60	0.98
Ours-d	d	**954.36**	**288.64**	**3.21**	1.35
Morph-Net [9]	RGBd	1045.45	310.49	3.84	1.57
Spade-RGBsD [21]	RGBd	917.64	234.81	**2.17**	0.95
NConv-CNN-L1 [12]	RGBd	859.22	**207.77**	2.52	**0.92**
NConv-CNN-L2 [12]	RGBd	829.98	233.26	2.60	1.03
Ours-RGBd	RGBd	**814.73**	249.95	2.8	1.21

loss used in training, as is discussed in subsection 2.4.2. Other metrics, however, are also insightful as they prioritize different aspects of the problem and ultimately there is no clear choice for the best one. For instance, errors such as the inverse root mean square error (iRMSE) prioritizes nearby points, which may be more relevant due to their proximity, but one could also argue that distant points are more useful for planning and that other approaches (such as stereo and radio) are already effective at close range. To deal with these issues, [2] suggests a different parametrization of depth coined *proximity*, which is given by

$$p = \frac{a}{d + a} \tag{2.6}$$

where a is a some normalization factor (e.g. the average) for the depth d, thus retaining characteristics of both regular depth and inverse depth, though there is no systematic study of the consequence of using such approach.

2.5.2 Ablation study

To better understand the importance of the proposed components and associated design flexibility, we perform an ablation study by comparing the network performance under different modifications around the proposed baseline, as displayed in Table 2.3. Note that one must take into account the stochastic nature of any such experiment and the computational limitations that hinder additional statistical analyses.

Table 2.3: Ablation study of the network architecture for depth input. Empty cells indicate the same value as the first row of each section.

image	fusion split	loss	ResNet depth	with skip	reduced filters	pre-trained	Nº pairs	down-sample	dropout & weight decay	rmse [mm]
None	-	L_2	34	Yes	2x (F_1=32)	No	5	No	No	**991.35**
		L_1								1170.58
			18							1003.78
				No						1060.64
					1x (F_1=64)					992.663
					1x (F_1=64)	Yes				1058.218
					4x (F_1=16)					1015.204
							4			996.024
							3			1005.935
								Yes		1045.062
									Yes	1002.431
Gray	16/48	L_2	34	Yes	1x (F_1=64)	No	5	No	Yes	**856.754**
RGB										859.528
	32/32									868.969
			18							875.477
				No						1070.789
	8/24				2x (F_1=32)					887.472
							4			857.154
							3			857.448
								Yes		859.528

The results indicate that a considerable margin for architecture modifications that would still allow for reasonable performance. Most notably, the reduction in the number of filters by a half (which nominally reduces computational requirements by four times) does not drastically affect prediction quality, a result that motivates simplifications for other experiments in this dissertation. Other remarkable observations

include the superiority of the L_2 loss over the L_1 one (unlike some previous findings [33]), the ineffectiveness of dropout and weight decay and robustness to the reduction of encoder-decoder pairs. Assigning more filters to the image branch is conjectured to be beneficial because features from natural images are harder to process and understand in comparison to LiDAR to measurements, which directly represent the desired quantity.

2.5.3 Sparse input distribution

Although the baseline LiDAR sensor data provided in the KITTI dataset contains 64 lines, such a setup might not be desirable in terms of hardware cost, so it becomes attractive to explore how the sensor fusion is affected by the number of laser measurements of the sensor, which is correlated to its price. We then process the raw sensor data in order to identify said scanlines (they are not differentiated beforehand) and discard some of them in order to emulate the output of lower resolution sensors, even exploring configurations not commonly used, such as a single line scanner. Additionally, we also downsample depth measurements by uniformly randomly sampling them for comparison purposes.

The scanline identification was done based on computed angles of the points with respect to the horizontal plane and their assumed sequential nature in the raw data due to the device's operating principle. We note that the x and y coordinates in the projected camera image are not convenient features to identify each line as these values are also dependent on the measured depth and other factors, so grouping becomes harder, and that this approach is simpler than separating the scanlines from the depth images for similar reasons. These issues are illustrated in Figure A-1, where several artifacts appear on what should ideally be a set of horizontal lines. Finally, the selection of which lines are to be used aimed at retaining symmetry around the middle one .

In sequence, movement compensation is added and the measurements are projected into camera frame. The resulting reference images match the original ones from the KITTI depth dataset (whose generating code is not publicly available) with

the exception of a few occasional points off by a pixel and consequently we take the original coordinates and depth if they are close (i.e. at most a pixel off in x or y) to the generated reference image in order to get the final downsampled depth image. An illustration of the final result is given in Figure 2-3. Note that for the dataset the line sampling is expected to contain more points since the topmost regions of the image often contain fewer samples due to the geometry of the scenes, so the center and bottom lines contain more points than the average of roughly 300 points.

Figure 2-3: Illustration of the two types of downsampled sparse depth image

The network is trained for downsampling factors of 2, 4, 8, 16, 32 and 64 (down to a single scanline) and the results are displayed in logarithm scale in Figure 2-4. The superiority of uniformly random sampling is evident, despite that fact that it

contains slightly fewer points, which suggests that ensuring a well distributed center might be more beneficial than simply adding more samples, an act that is shown to have diminishing returns given the apparent linearity of the four plots in the graph (thus indicating a monomial trend). This trend is slightly off for for the full resolutions case in comparison with scanline downsampling, a result that we attribute to the fact that the hyperparameter optimization was performed for the former case and due to potentially discarding additional points when downsampling. The benefit of a camera is also clear, most specially as the density of the sparse samples is reduced and also considering their relative low prices in comparison to LiDAR systems.

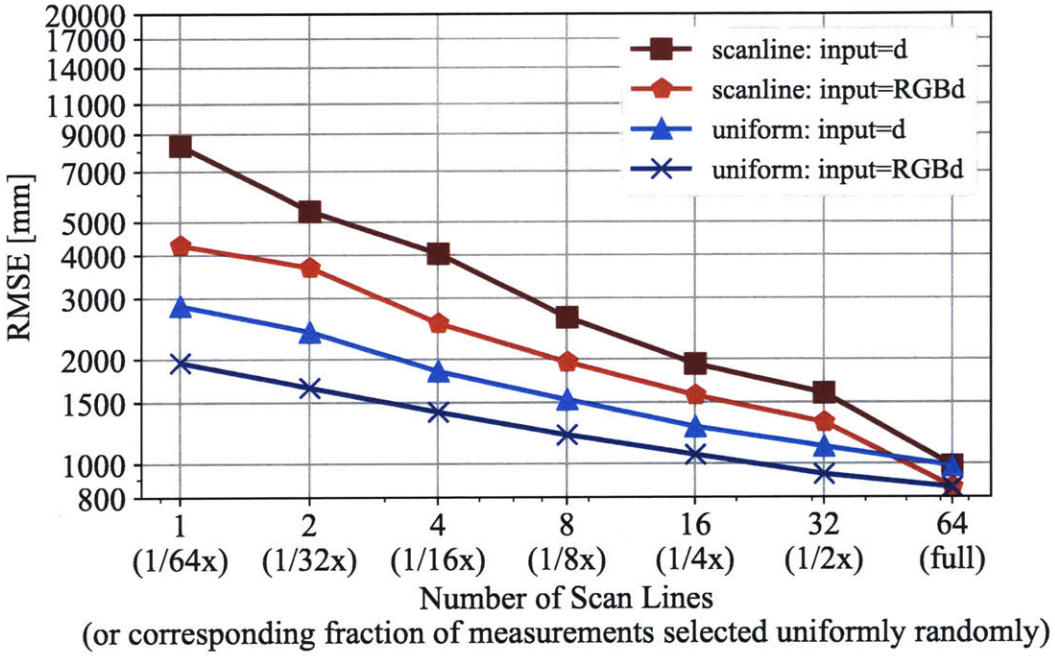


Figure 2-4: Prediction error against number of input sparse depth samples for both types of distributions (uniform random sampling and LiDAR scan lines selection) and both with and without camera images

Chapter 3

Visual Inertial Odometry Considerations

3.1 Introduction

In this chapter we are going to discuss some of the most common methods for visual inertial odometry (VIO) and simultaneous localization and mapping (SLAM) before they can be integrated with depth estimation approaches. The goal is not to provide a comprehensive overview of the topic, but to outline aspects that are of interest for the rest of this thesis and introduce aspects of the solution used.

3.2 Background

3.2.1 Problem formulation

One of the most fundamental tasks for a robotic system is to estimate its internal states based on sensor information. This is often done while also generating a map of the environment, even when that is not explicitly necessary, as sensor readings commonly depend on both. The extent upon which the map is used (by, for instance, performing loop closure and ensuring global consistency) sets apart VIO and SLAM, but this difference is of little importance to this dissertation, so we drop the distinction

between the two. A more comprehensive analysis of SLAM algorithms is given by [3], which incidentally argues that visual-inertial navigation can be seen as a "reduced" SLAM and is also the main reference for this chapter.

Mathematically, let x_k and z_k be the systems states and the measurements, respectively, at time k, which also determine the sets X and Z. The goal of SLAM in a maximum a posteriori formulation is to find estimates of the states $\hat{X} = \hat{x}_k$ by finding the most likely states given the measurements, which translates to

$$\hat{X} = argmax_X \; P(X|Z) = argmax_X \; P(Z|X)P(X) \tag{3.1}$$

where $P(X)$ encodes prior knowledge about X and $P(Z|X)$ encodes the probability of the measurements over the states, which assume a noisy observation function given by $z_k = h_k(X) + e_k$, for some random noise e_k and observation model h_k. Under appropriate conditions, the problem turns into a nonlinear least square minimization of the form

$$\hat{X} = argmin_X \sum_k \rho_k(h_k(X) - z_k)^2 \tag{3.2}$$

where ρ_k is a cost function that depends on noise assumptions (e.g. under gaussian noise it becomes a quadratic norm whose weight depends on the inverse of the noise covariance matrix). Solutions are often found with the use of successive linearization procedure such as Levenberg-Marquardt or Gauss-Newton.

It is convenient to reformulate the problem in a graphical model known as factor graph [29], which allows for a generic and easy to visualize formulation. It also facilitates the use of message passing algorithms such as the sum-product algorithm in order to infer marginal distributions or other quantities of interest.

3.2.2 Practical considerations

For the interest of robotics systems, it's common for SLAM problems to involve locally connected and sparse factor graphs. Intuitively, a vehicle traveling through

an environment will only detect a fraction of the available landmarks and in under sufficiently small displacements a feature that is visualized at time t will also be visible at nearby times, but not along the whole trajectory. This allows for improved computational efficiency as fast and memory-efficient sparse solvers can be employed and also facilities incremental solutions, both important considerations for online operation.

Another practical aspect particularly important in visual systems is the potential necessity to extract important feature or adopt other strategies for intractably large or difficult to model measurements. Cameras can typically have hundreds of thousands of pixels per frame and analytically relating these values to a compact state representation may not be feasible. Although not impossible to do it [48], it's common to instead use a small relevant subset of measurements by selecting corners, lines (or other "features") or some other sampling strategy in order to facilitate associating measurements to landmark locations and states. This part of the processes is called the front-end, while the factor graph optimization is denominated the front-end.

There are many variations in front-end systems, but [14] classifies them according to two properties: sparse versus dense and direct versus indirect. Direct approaches work on the intensity values as opposed to using derived information such as feature positions and their associated geometric errors. As such, they can potentially be more precise by taking into account information from featureless regions, but they are usually more brittle and sensitive to calibration. The other category, as the name suggests, refers to the number of points used and consequently involve different assumptions: dense method work with large connected regions and thus tend to favor smoothness, while sparse method select a subset of points without the neighborhood considerations. It's worth noting that a method can fall into any combination of categories and can be an hybrid solution that involves, for example, both indirect and direct approaches [15].

3.3 Implementation and acknowledgment

The framework used on this thesis is an indirect sparse odometry solution based on tracking "good features to track" points [45] across frames and optimizing them in a small window using GTSAM . It was already used successfully in a real-time embedded VIO setting [43] and was adapted to work on the used datasets and in the absence of inertial measurements with major assistance from Varun Murali. We note that the number of RANSAC iterations was set to a large value to compensate for the loss of pose priors given by a gyroscope.

Chapter 4

Fusing temporal information with depth

This chapter is dedicated to the discussion of results pertaining the use of VIO triangulations as a source of sparse depth measurements from temporal information to allow for neural network assisted depth completion.

Some experiments will be performed as a comparative analysis with LiDAR points by trying to isolate the characteristics that are most helpful in each sensor. Properties such as scale, distribution and range will be taken into account and finally the method will be applied with actual visual odometry data.

4.1 Related work

The problem of extracting useful information from a set of images is a central topic in computer vision and appears on a variety of situations that can't be comprehensibly analyzed in this thesis. Multi-view stereo, structure from motion and SLAM are all problems that end up computing some notion of distance from an observer to the environment, although not always as a central quantity of interest. Regardless, these methods provide useful guidance as the computation of depth is related to other quantities such as camera poses and optical flow.

Combining visual odometry and depth prediction is a recent endeavor in the aca-

demic literature, although several works focus only on how depth prediction can improve SLAM and not the opposite. The works of [60] attempt to improve existing monocular direct sparse odometry approaches, which are sensitive to initialization and (as any purely monocular VO) is ambiguous in scale. This is done by using a neural network to generate virtual disparity maps and then utilizing them in a stereo odometry pipeline. The resulting algorithm is then potentially able to provide a learned scale and improved stability in comparison to the original DSO (direct sparse odometry) formulation [14]. The authors of [50] also combine depth prediction and monocular SLAM to achieve better 3D reconstruction and semantic labeling in indoor scenes, which allows them to illustrate the stability of depth prediction under purely rotational motion, a challenging situation for monocular odometry. Following a more end-to-end approach, [61] replaces the conventional geometric odometry approaches with a neural network for depth prediction, another for flow computation and another to combine depth and optical flow to generate pose estimates and create dense 3d maps, thus fusing the depth maps after they were predicted independently.

Perhaps one of the first works to successfully use information from more than a single image as input to a neural network to compute depth is [53], where several networks are used to computed depth and egomotion from a pair of images with an intermediate computation of optical flow. These auxiliary values seem necessary to integrate multi-view information as the trivial approach of concatenating images as input to a neural networks was reported as ineffective by [63]. A similarly end-to-end solution is presented in [62], that makes use of virtual keyframe generation and a series of different networks to achieve both tracking and depth map generation. They also combine images with a cost volume computed from several frames in an iterative application of a refinement network. Cost volumes are also used with neural networks and a geometric regularization scheme by [55] to combine multi-frame information into depth maps.

Another study [4] tries to combine sparse depth measurements with camera information by using edge-preserving filters, thus avoiding the necessity of training a neural network. Their analysis on KITTI is actually solely based on subsampling

LiDAR depth maps either uniformly randmomly or selecting them based on feature points (and as such is more closely related to the problem of depth completion), while temporal information is only used for data collected indoors by the authors and for which groundtruth is available through a stereo setup, so direct comparison is not straightforward. Nevertheless, from an algorithm standpoint, they also explore the use of sparse VIO points to obtain dense depth, although their SLAM pipeline already produces dense representations [27].

Long Short-Term Memory units are used by [56] as an alternative to handle temporal data and achieve improved depth estimation and pose prediction in challenging situations. These recurrent solutions are not as common when depth is of interest, but recent works are exploring RNN for visual odometry [57]. Although these types of networks are designed to work with temporal data, it's not clear whether recurrent networks are superior to simple convolutional ones (at least in sequence modeling tasks) [1], so further exploration is required.

An approach based on variational auto encoders is explored in [2], in which features are extracted from an image and are fed into the decoder part of an auto encoder, alongside a small encoding vector. The multi-view coupling is given by the fact that the code is optimized by a costly structure from motion minimization framework based on photometric and geometric minimization. One remarkable insight from their work is that a small code size (they use a vector with 128 elements) is enough to capture information to significantly improve the otherwise single image depth prediction network, suggesting that not a lot of information needs to be transferred from one frame to the other.

A different type of integration is given by [63] or [54], where the goal of odometry is to provide correspondences between frames and thus allow unsupervised training through the minimization of photometric or other type of losses. The inherent lack of scale still needs to be compensated by other methods, potentially with additional sensors. Although we focus on supervised training for this chapter, it's worth mentioning such results as they provide further motivation to integrate odometry and pose estimation into depth estimation, even when there is no explicit need for localization.

4.2 Attributes of triangulated points

In order to compare the sparse depth points computed by VIO with the ones measured from other sources (e.g. from a LiDAR sensor), it's important to characterize in how they differ, so we can properly define relevant experiments.

- scale

  Due to the inherent scale ambiguity of purely visual odometric methods, their associated sparse depth measurements are only correct up to a constant factor. In general this deficiency is not present in other sensors and can be circumvented in practice by the addition of inertial sensors or of a stereo setup. Nevertheless, some datasets do not have such information available and so we also explore whether unscaled sparse points can provide useful information. Note that that's not an error or uncertainty, but an inherent characteristic of the sensor.

- spatial distribution

  Triangulated points usually come from locations with features, especially for indirect methods, and thus its spatial distribution is largely dependent on the environment. LiDAR sensors have a more characteristic distribution of points based on separate scan-lines.

- quantity

  The number of sparse depth measurements clearly affects the difficulty of the problem. The previous chapter already analyzed the effects of reducing the number of LiDAR scan-lines, as denser sensors can be drastically more expensive, but have diminishing returns. Indirect visual tracking methods are more limited in the sense that they rely on existence of salient features on view and can usually only triangulate a few hundred of those. Direct methods are potentially advantageous since they may track more points and can work on feature-less regions, but contains other drawbacks, such as requiring additional calibration and lacking the same robustness [14].

- consistency

  A metric that is related to quantity, consistency measures the deviations in the number of measurements with relation to a nominal value. This is specially relevant when using odometric measurements as the number of tracked points depends heavily on the environment and tends to drop between each keyframe (when new features are added, depending on the method).

- observability

  In order to properly triangulate a landmark, it needs to be visible for several frames and (due to how SLAM algorithms are usually designed) also needs to be static. These limitations can be relevant in some applications as moving objects may require more precise estimation (e.g. another car in a driving a scenario). Other observability considerations also apply to different sensors as for example certain absorptive or transparent surfaces prevent laser correct laser or sonar measurements.

- range:

  Both maximal and minimal range define the envelope for which a sensor measurement is valid. In general, for VIO it's typical to triangulate points that are close since they have more disparity and including far away points requires introducing long term dependencies, which turns the triangulate more computationally demanding, besides requiring to track the same feature for longer.

4.3 Proposed approach

In order to combine both depth completion and visual odometry, we use the neural network developed in chapter 2 in combination with an indirect visual odometry pipeline chapter 3. The triangulated positions of landmarks are transformed into a sparse depth map and used as inputs to the neural network, which then combines that information with camera images.

The used neural network is based on the one presented section 2.4, which achieved state-of-art results on the KITTI depth completion challenge, and was slightly modified to reduce computational requirements given that the ablation studies indicate that such modifications are expected to have minimal effect. Namely, the number of filters was reduced by a half.

For the purpose of isolating the effects of different attributes of the sparse measurements, we also perform different experiments where the sparse measurements are instead taken from another source, such as a real or simulated LiDAR or where some transformation is applied to these points, such as normalizing them with the intent of removing their scale.

4.4 Experiments

4.4.1 Datasets

We use both KITTI [17] and Synthia [38] datasets to evaluate the proposed methods. The first one is a real dataset widely used in the contexts of odometry, depth completion and depth estimation, although rarely combining more than one modality. The second is a synthetic dataset that allows the circumvention of several issues of the former. Namely, its ground truth is completely dense and is not biased towards the values that can be computed sensor post-processing technique proposed by the dataset authors. To have a similar comparison, we limit the maximum and minimum depth in the synthetic dataset to $2m$ and $90m$ respectively based on data from the real one.

With respect to the KITTI dataset, we use two different splits and ground truths. The older split was proposed in [11] and is often referred to as the Eigen split in reference to its author, while the newer one comes from [51] and is most commonly associated with depth completion and depth prediction challenges. We will refer to former as the "old" split and the latter as the "new" one, a complication that is necessary since the older split is more commonly used, but the other one contains

more data and has a denser and more precise ground truth, since it was generated by combining information from several sensors at different times.

4.4.2 Lack of scale and normalization

On a purely monocular visual odometry problem, the choice of scale is inherent ambiguous, which in practical terms indicate that that any triangulation in such settings (when it's not an outlier or other erroneous results) will only be correct up to a constant multiplicative factor. That is, if the triangulated depths in frame j are $d_i^j \in D_j$, then the correct values $\bar{d}_i^j$ are such that $d_i = \lambda^j \bar{d}_i$ for some unknown λ^j. Depending on the choice of normalization, it's possible to have $\lambda_j = \lambda$ as a constant for a givens sequence.

To the best of the author's knowledge, the problem of integrating this scaleless information into a neural network for depth completion has not been addressed in the literature. To explore this question, we suggest different normalization schemes and compare them with correctly-scaled data. They all follow a simple structure

$$\tilde{d}_i = \frac{d - M(D)}{N(D)} \tag{4.1}$$

where $M(D)$ and $N(D)$ are respectively a mean and a norm dependent on the normalization scheme. We note that this is only applied sparsely to the points with known depth, so it's similar to instance norm [52], except that it's sparse and different strategies are used. Additionally, no batch or instance norm is applied after the first convolution, since the data is already normalized. The results are in Table 4.1. We note that this problem is mostly present due to the lack of inertial information in commonly used datasets and that such inexpensive device would greatly simplify the problem both with the recovery of scale and as a source of initialization for the pose optimization.

There is a slight decrease in performance for all schemes in comparison to using the correct scale. These results indicate that even when scale is unknown it's still possible to take advantage of the available information to improve depth perception.

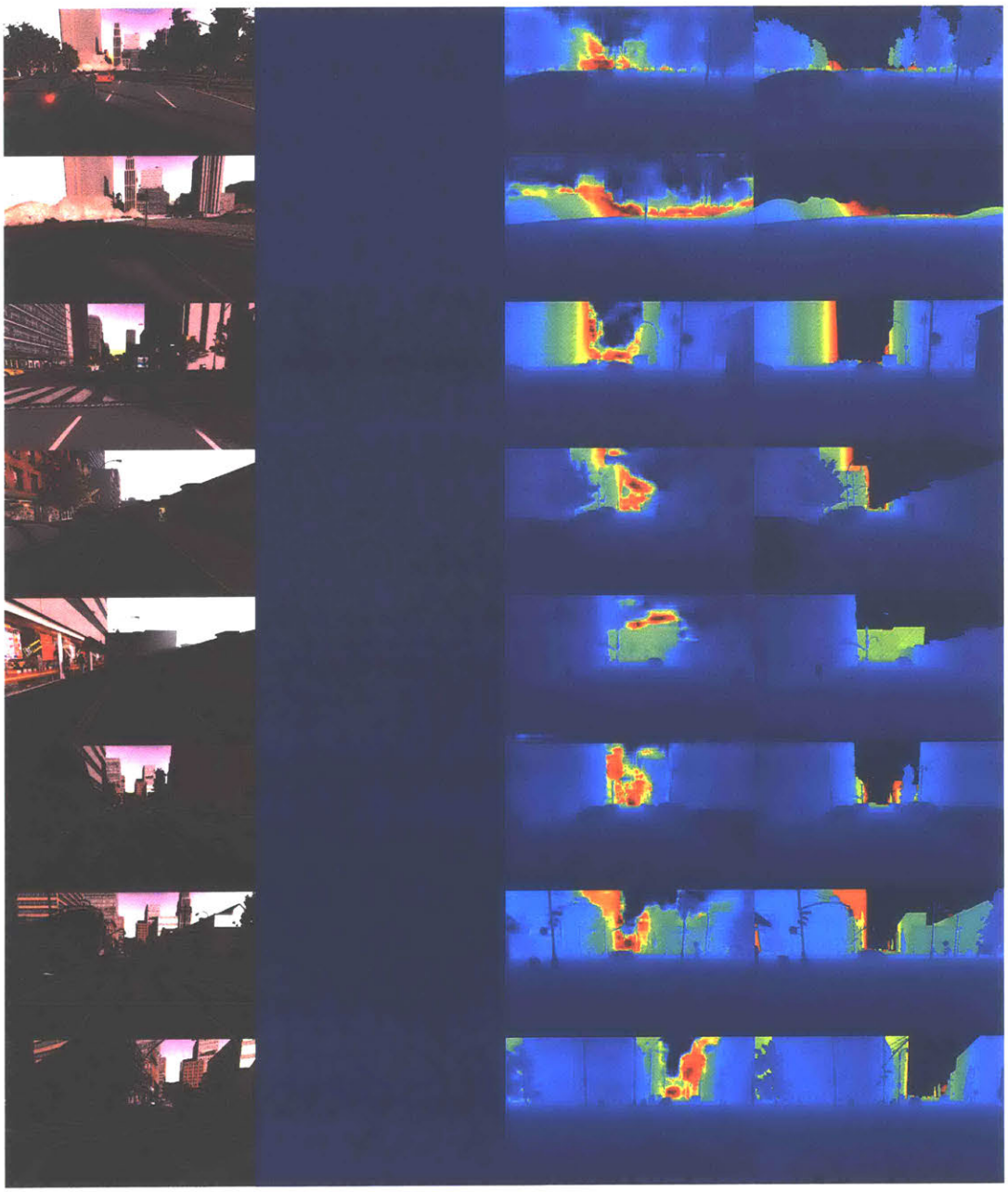


Figure 4-1: Illustration on Synthia dataset.

Table 4.1: Comparison of different sparse normalizations schemes.

Strategy	RMSE [mm]	$M(D)$	$N(D)$
none (true)	887	-	-
med	1071	0	median(D)
nrm	1025	mean(D)	std(D-mean(D))
max	937	0	max(D)

It's also worth mentioning that the distribution of sparse points in this case (coming from a LiDAR sensor) is relatively uniform and consistent and as such it's possible that network is in practice learning to undo the normalization given some typical expected values. For instance, in the case of the "max" normalization, if most frames contain at least a single point with depth close to the maximum range d_{max}, multiplying that normalized measurements $\tilde{d}_i$ by d_{max} would approximately recover the true scale. However, the existence of outliers and a less uniform distribution may hinder such approaches for different sources of information. Consequently, we attempt to use both the "nrm" and the "max" normalization. We also report that results failed to generalize correctly when a naive batch normalization was applied, which is to be expected since in the used encoding the null value indicates invalid depth measurements and not the actual value of zero.

4.4.3 Range

One important question when it comes to characterizing the difference between LiDAR measurements and triangulated points is their typical range. Depth can only be perceived if there is enough baseline motion to cause disparity, so although it's possible to triangulate faraway points given enough movement, this is will only happen if tracking is performed correctly along several frames and even so it may complicated the underlying optimization problem by adding long-range dependencies between frames. A distinction between close and far points is made in [36], where it's argued that the former are more beneficial to translation and scale information, while the latter contribute mostly to orientation estimation.

In light of these considerations, we performed experiments on the KITTI dataset

using two different proposed splits and constraining the sparse inputs (but not the ground truth) to contain either only close points ($d < 20$) or far points ($d > 20$). The sparse points were uniformly randomly sampled from the LiDAR input in order to obtain a 64 times downsampling factor or 300 points on average. The results are in Table 4.2

Table 4.2: Analysis of the effect of maximum and minimum range.

Split	Distance	RMSE	MAE	iRMSE	iMAE	silog
New	Close	2975	1188	7.174	3.94	9.3
New	Far	2411	1195	11.78	6.83	11.1
Old	Close	2070	991	10.26	5.22	8.8
Old	Far	2116	1126	13.71	7.39	11.01
Old	All	1976	926	10.77	5.34	8.5

The values indicate that for the specific network evaluated the relative importance of close or faraway points is not entirely conclusive as each one split indicate larger error for close points, while the other indicates the opposite (in terms of RMSE). Additionally, as would be expected metrics that prioritize closer points are improved when close sparse measurements are provided.

4.4.4 Depth at feature points

In order to gauge the effects of having samples in a distribution similar to a typical VO pipeline, we perform comparison experiments using depth points sampled from feature points, most specifically using the Good Features To Track (GFTT) approach [45]. For the Synthia dataset, these features are extracted from the colored image and the depth values are taken from the groundtruth, while for the KITTI dataset both the features and depth are extracted from the groundtuth, since it's sparse and some regions do not have any truth values nearby (so we refer to it in quote symbols). For the purpose of experiments, the features have a minimum distance of 10 pixels and a quality factor of $1e-3$, which effectively causes the total number of detected features to not be exactly the number of requested ones.

The results from Table 4.3 point to the fact that if the number of features used

Table 4.3: Comparison of different sparse measurement distributions

Dataset	Images	Features	Norm.	RMSE [mm]
Synthia	RGB	-	-	4179
Synthia	Gray	-	-	4283
Synthia	Gray	200 GFTT	-	3815
Synthia	Gray	200 GFTT	max	4144
Synthia	Gray	2000 GFTT	-	3000
KITTI	Gray	200 "GFTT"	-	1536
KITTI	Gray	300 random LiDAR	max	2116

is small enough, then using normalized features achieves similar performance to not using any feature at all. Naturally, the proposed method relies on distributing how many filter come from the convolution that acts on the sparse depth image and how many come from the color image, so if the information of one of the sources is poor, the overall performance can be reduced (this experiment is performed in the next subsection). However, a more noticeable improvement is verified when the scale is correct and when there are more points available. Another interesting result is the verified superiority of using color image as input as opposed to its gray version for the Synthia dataset, a trend that is opposite to what was verified in KITTI (at least for for LiDAR depth completion) and may be associated with the simplicity of simulated environment in comparison to the real world.

4.4.5 Integrating with visual odometry

In this subsection we present results of experiments involving the proposed depth completion framework with normalized sparse depth measurements computed by using actual visual odometry solutions (see chapter 2 for a more detailed description). For the Synthia dataset, we use the *dawn*, *summer*, *spring* and *fall* sequences and for the KITTI dataset we remove sequences for which there is almost no movement and the first frame of each sequence (since they have no associated triangulation).

A summary of experiments on the KITTI dataset is displayed in Table 4.4 considering both common utilized splits and the two sources of ground truth (see subsection 4.4.1), where "g" indicates using a gray image and "g+d" refers to using both

Table 4.4: Comparison of the combined system under dataset variations

Split	GT	Input	RMSE [m]
New	New	g	3815
New	New	g+d (null)	4590
New	New	g+d (max norm)	3793
New	New	g+d (nrm norm)	3865
Old	New	g	2423
Old	New	g+d (null)	2566
Old	New	g+d (max norm)	2426
Old	New	g+d (nrm norm)	2434
Old	Old	g	3115
Old	Old	g+d (max norm)	3156
Old	Old	g+d (nrm norm)	3148

gray image and VO triangulations with some normalization ("max" or "nrm"). The "null" mark indicates that the sparse measurements were set to 0, but the architecture remained the same, a procedure that was done in order to identify the consequences of removing filters from the branch that takes the intensity image as input. The results indicate that indeed the sparse measurements help the architecture that was modified to accommodate them, but overall shows similar performance to using the one specialized to using images only. This is in accordance with subsection 4.4.4, which indicated more substantial improvements when using the true scale or more sparse points.

To the best of the author's knowledge, there isn't a competition like the KITTI depth completion or depth estimation challenges [51] applied to temporal data in driving situations. As such, for the sake of comparison we use the old split provided by Eigen [11] and LiDAR information as true depth values. We also use the "nrm" normalization for the multi image procedure, as is displayed in Table 4.5. We note that [56] claims an improvement to 2538 RMSE error when evaluating on a randomly selected set of continuous sequences taken from the same ones as the test set, but a fair comparison is impossible without using the same set of images. For similar reasons the second part of the table indicate results for the test set when removing the sequences for which there is little movement (effectively reducing the number of

samples from 652 to 577).

Table 4.5: Comparison with other methods

Method	Abs rel	Sq rel	RMSE	RMSE-log
DenseSLAMNet [56]	0.129	0.704	4743	0.199
Eigen et al. [11]	0.190	1.515	7156	0.270
Liu et al. [28]	0.217	1.841	6986	0.289
DORN (ResNet) [16]	0.072	0.307	2727	0.120
Kuznietsov et al. [26]	0.113	0.741	4621	0.189
Ours (single)	0.163	1.008	4761	0.228
Ours (multi)	0.167	1.036	4753	0.229
Ours (single)	0.157	0.958	4691	0.221
Ours (multi)	0.157	0.952	4666	0.221

The comparison indicates that there is a small improvement in some metrics when using the sparse points, most noticeably when the stationary points are removed (and for which null sparse measurements were given). The overall performance is in line with recent state-of-the-art methods, with the exception of DORN, which achieves significant improvements over all other approaches.

4.4.6 Oversampling at stationary poses

In an attempt to reduce the potential bias created by static situations and the multiple similar images they generate, we proposed and tested a weighting scheme that aimed at prioritizing points that were far from each other and therefore expected to be different. This was done by using the groundtruth trajectory (when no groundtruth is available, one could potentially use the estimated trajectory instead) for each sequence in the Synthia dataset as measuring the similarity of two images is not trivial, so the camera poses served as a surrogate metric for dissimilarity. Mathematically, for the N training images in a frame we compute the distance matrix M_{ij} of all the $3D$ positions and compute the number p_i of poses that are closer than a threshold d. Then, the score of a frame is set as $w_i = \frac{1}{p_i}$, which is then normalized to get the weight $\bar{w}_i = w_i \frac{N}{\sum w_i}$ used to adjust the cost of all samples from frame i. Note that when points are sufficiently spaced we have $w_i = 1$ as $p_{ii} = 0 < d$ and this modification has no effect.

There was no noticeable improvement in results for the specific situations tested, but it's possible that this approach or similar considerations may turn out to be beneficial in other situations. Regardless, this a potential source of bias that should be addressed in the literature.

Chapter 5

Conclusions and future work

We conclude this work by summarizing results and related insights. We also comment on directions that may prove advantageous to other studies and to the advancement of the field.

5.1 Summary

In this work we proposed a depth completion architecture that achieved state-of-the-art results in a driving situation with sparse LiDAR measurements and camera images. This solution was then applied to a similar problem, where the sparse measurements actually come from a visual odometry pipeline. The proposed approach was studied under different situations in order to determine the main contributors to performance and evaluated in a common test benchmark, where a small improvement was verified. These analyses allows us to conclude that even greater benefits would be observed in the presence of inertial sensors or denser SLAM pipelines.

5.1.1 Dataset considerations

For this work, the KITTI dataset was used mainly due to its use in the literature, while Synthia had the advantage of containing dense labels. However, for the problem of fusing visual odometry and depth perception they both fall short in some aspects

which will be addressed in this section in order to guide the generation of other datasets.

As both datasets represent or try to recreate driving situations, it's expected that vehicle will be stationary for several frames. In the perspective of of single view applications, this is not a significant issue, but for some odometry solutions that perform optimization over a sliding window additional care must be taken in order keep information from the past. Moving objects can similarly become a more relevant issue for temporal data. It could be argued that this a natural challenge for any SLAM solution, but when the main interest resides in combining information it's practical to isolate and decouple different aspects of the problem. Beyond these issues, these repeated frames might create biases for very specific situations by oversampling very similar data.

Another practical consideration is the fact that having several small sequences (as happens in the KITTI dataset) make the problem harder for odometric approaches as there is not enough baseline in the first frames to triangulate far points and it becomes harder to verify adequate performance on all sequences. The KITTI odometry challenge, for example, uses around 10 sequences for training and 10 for test, while for this work roughly 70 were used. The Synthia dataset does not have this problem as it's comprised of 6 long sequences, but each weather variation contains approximately the same trajectory, so it's not as diverse.

It must also be noted that the initial ablation study and development of the neural network was given for a situation that is considerably different then the one it was ultimately used one: points distribution, range, scale, sparsity and consistency were all properties that changes from the initial development to the final application.

Finally, the lack of inertial sensors makes the problem more challenging without any practical advantage as inertial measurement units are substantially cheaper than the other sensors used. The scale problem is addressed in subsection 5.1.2, but for common SLAM pipelines inertial data provides useful priors for camera poses and greatly simplifies the optimization procedure.

5.1.2 The lack of scale problem

We proposed simple sparse data normalization scheme and demonstrated that useful information can be extracted from scale-less sparse depth measurements. This problem is not addressed in the literature and can be practically circumvented with cheap sensors, however other approaches could still be explored. For instance, the correct scale could be determined in an online fashion given a temporally consistent normalization and single frame depth estimation initializations could be provided to the odometry pipeline at the cost of a tighter coupling between systems and potential necessity to recompute the triangulations. This would complicate training as in general gradient descent is more efficient when samples are not correlated.

As potential approaches, one could also, for example, explore iterative solutions in order to refine the scale estimate to a stable value before proceeding to the next frames. This could be done with a separate branch that utilizes a common encoder part to output a single scalar and a potential loss based on how sensitive the output given the initial guess. Additionally, if the correct scale is known or can be estimated, one can sample slightly incorrect scales as input instead of having to recompute them.

5.2 Future directions

5.2.1 Multi-problem solution

Some recent works indicate that combining optical flow, semantic segmentation, depth estimation and related tasks together might produce better results than each problem individually [62][21]. This might also reduce dataset generation efforts as, for example, classifying objects may assist in identifying if they are moving or not. Additionally, a dense semantic label can be done by human annotators, but the same cannot be done with depth measurements.

5.2.2 Beyond the traditional convolution

Even though several tasks related to depth were improved with the introduction of neural networks, some results indicate that working with three-dimensional data requires a slightly different approach. This is made explicit by [16] and [58], two works that point out the importance of discontinuity in depth images and how the traditional spatial convolution may be limited as can commonly act as a smoother. They both achieve substantial improvement over other methods by, among other things, searching for a more adequate representation.

Appendix A

Supplementary information

A.1 Error Metrics

Given the n predicted values $\hat{y}$ and labels y, some of the error metrics used in this work and in the literature are mathematically defined below

$$\text{Root Mean Square Error (RMSE)} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{A.1}$$

$$\text{RMSE the log (RMSE-log)} = \sqrt{\frac{\sum_{i=1}^{n}\left(log(y_i) - \hat{l}og(y_i)\right)^2}{n}} \tag{A.2}$$

$$\text{Mean Average Error (MAE)} = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{A.3}$$

$$\text{Absolute Relative Error (Abs Rel)} = \sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\frac{1}{n} \tag{A.4}$$

$$\text{Squared Relative Error (Sq Rel)} = \sum_{i=1}^{n}\frac{(y_i - \hat{y}_i)^2}{y_i}\frac{1}{n} \tag{A.5}$$

$$\text{RMSE of the inverse (iRMSE)} = \sqrt{\frac{\sum_{i=1}^{n}\left(\frac{1}{y_i} - \frac{1}{\hat{y}_i}\right)^2}{n}} \tag{A.6}$$

$$\text{Mean Average Error of the inverse iMAE} = \frac{\sum_{i=1}^{n}\left|\frac{1}{y_i} - \frac{1}{\hat{y}_i}\right|}{n} \tag{A.7}$$

$$\text{Scale Invariant Logarithmic error (SILog)} = \frac{1}{n}\sum_i d_i^2 - \frac{1}{n^2}\left(\sum_i d_i\right)^2 \tag{A.8}$$

with $d_i = log(y_i) - log(\hat{y}_i)$. Note that some texts define the Squared relative error differently:

$$\text{Alternate Squared Relative Error} = \sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2 \frac{1}{n} \tag{A.9}$$

A.2 KITTI dataset visualiztion

We present additional information to chapter 2. Namely, visualization of the scanline distribution is given in Figure A-1 and additional comparisons for the KITTI depth completion challenge is given in Figure A-2

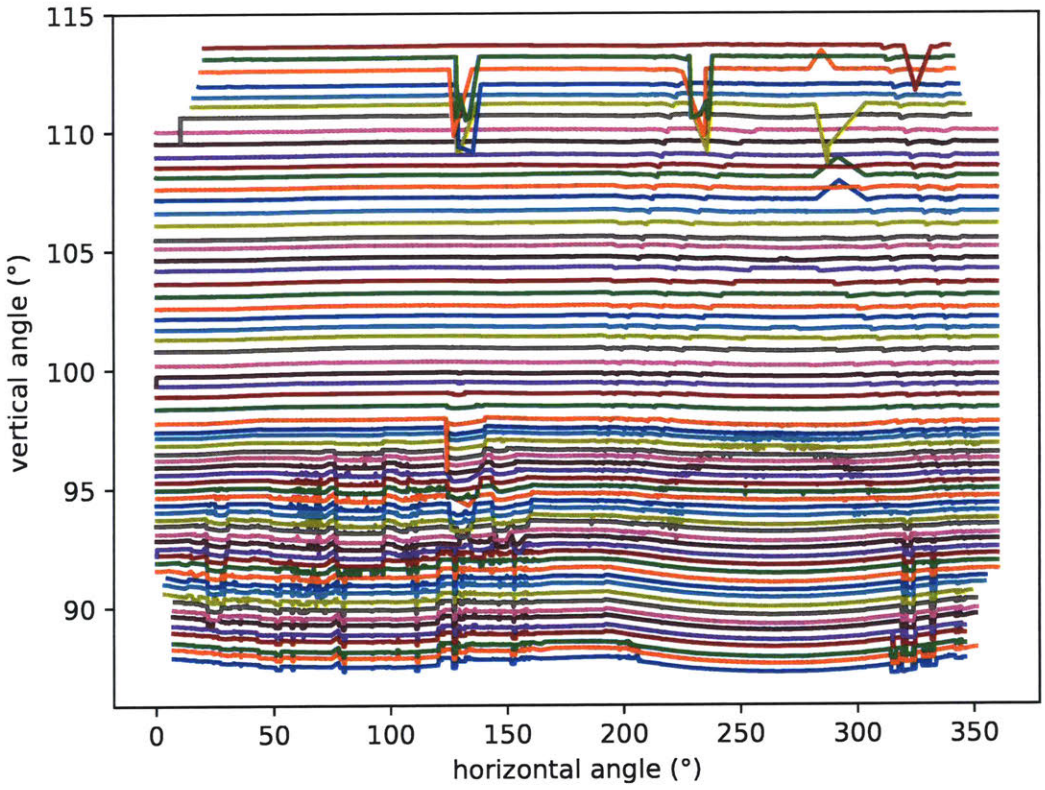
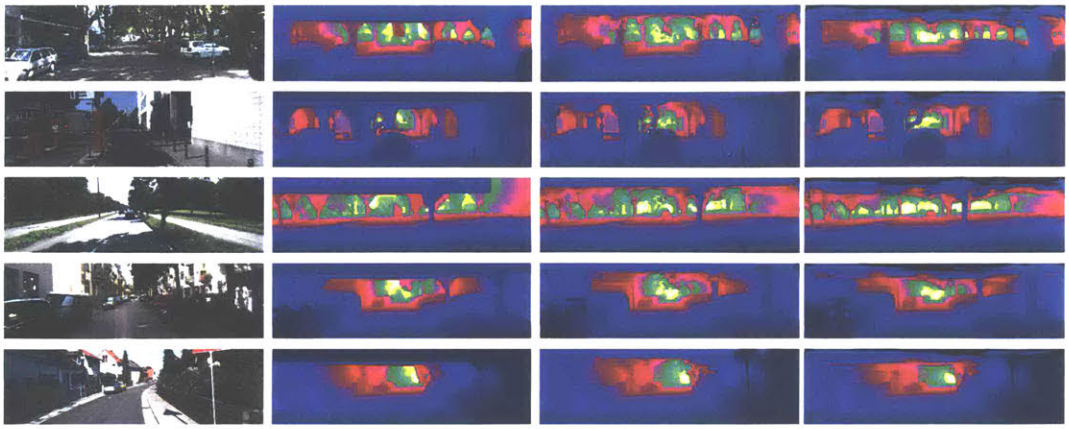


Figure A-1: Visualization the scanline distribution



(a) RGB (b) NConv-CNN [13] (c) NN+CNN2 [51] (d) Ours-d

Figure A-2: Comparision against other methods (best viewed in color).

A.3 NYU dataset

In order to illustrate the flexibility of the depth completion method, we present some visualizations on the NYU-Depth-v2 dataset[46] in Figure A-3 and comparison in Table A.1. This set of indoor environments display larger camera rotations and different depth ground truth (from an RGBD camera). Following the official data split the sparse measurements are generated by uniformly sampling 500 points from the true values. In comparison with other methods, the approach achieves similar state-of-the-art performance.

Table A.1: Comparison against state-of-the-art algorithms on the NYU dataset.

Method	RMSE	REL
Ma et al. [33].	0.230	0.044
Cheng et al. [5]	**0.117**	**0.016**
Ours (supervised)	0.133	0.027

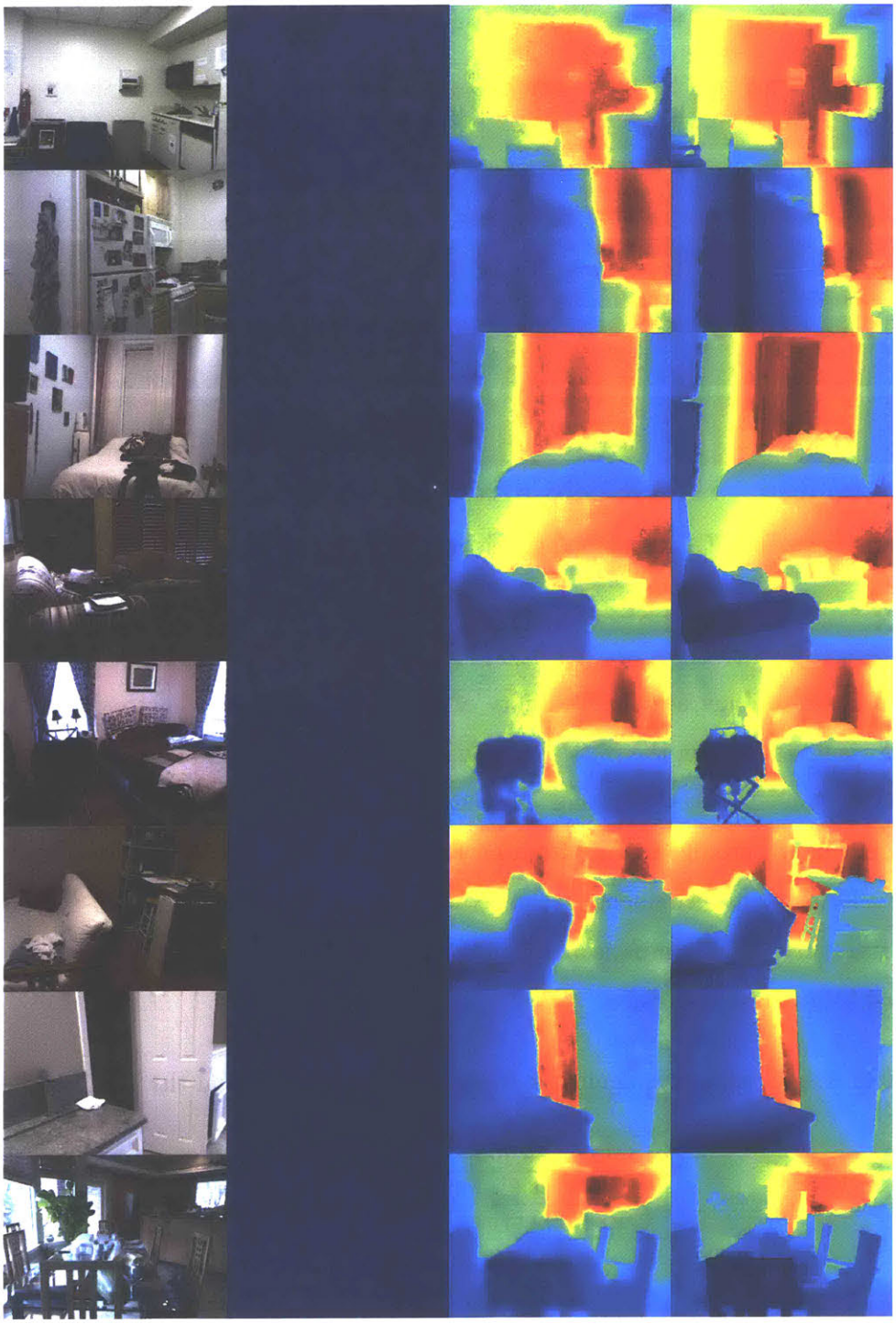


Figure A-3: Illustration of the results in the NYU Depth dataset.

Bibliography

[1] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[2] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. CodeSLAM-Learning a Compact, Optimisable Representation for Dense Visual SLAM. *arXiv preprint arXiv:1804.00874*, 2018.

[3] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[4] Hui Cheng, Zhuoqi Zheng, Jinhao He, Chongyu Chen, Keze Wang, and Liang Lin. Embedding Temporally Consistent Depth Recovery for Real-time Dense Mapping in Visual-inertial Odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 693–698. IEEE, 2018.

[5] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *European Conference on Computer Vision*, pages 108–125. Springer, Cham, 2018.

[6] Nathaniel Chodosh, Chaoyang Wang, and Simon Lucey. Deep Convolutional Compressed Sensing for LiDAR Depth Completion. *arXiv preprint arXiv:1803.08949*, 2018.

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[9] Martin Dimitrievski, Peter Veelaert, and Wilfried Philips. Learning Morphological Operators for Depth Completion. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 450–461. Springer, 2018.

[10] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[11] David Eigen, Christian Puhrsch, and Rob 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.

[12] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence Propagation through CNNs for Guided Sparse Depth Regression. *arXiv preprint arXiv:1811.01791*, 2018.

[13] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Propagating Confidences through CNNs for Sparse Data Regression. *arXiv preprint arXiv:1805.11913*, 2018.

[14] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2018.

[15] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014.

[16] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep Ordinal Regression Network for Monocular Depth Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[21] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE, 2018.

[22] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[25] Jason Ku, Ali Harakeh, and Steven L Waslander. In Defense of Classical Image Processing: Fast Depth Completion on the CPU. *arXiv preprint arXiv:1802.00036*, 2018.

[26] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2215–2223. IEEE, 2017.

[27] Yi Lin, Fei Gao, Tong Qin, Wenliang Gao, Tianbo Liu, William Wu, Zhenfei Yang, and Shaojie Shen. Autonomous aerial navigation using monocular visual-inertial fusion. *Journal of Field Robotics*, 35(1):23–51, 2018.

[28] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1253–1260. IEEE, 2010.

[29] Hans-Andrea Loeliger, Justin Dauwels, Junli Hu, Sascha Korl, Li Ping, and Frank R Kschischang. The factor graph approach to model-based signal processing. *Proceedings of the IEEE*, 95(6):1295–1322, 2007.

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[31] Hendrik Johannes Luinge. *Inertial sensing of human movement.* Twente University Press Enschede, 2002.

[32] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised Sparse-to-Dense: Self-supervised Depth Completion from LiDAR and Monocular Camera. *arXiv preprint arXiv:1807.00275*, 2018.

[33] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[34] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[35] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[36] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[37] Olaf Ronneberger, Philipp Fischer, and Thomas 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.

[38] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio 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.

[39] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[40] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization?(no, it is not about internal covariate shift). *arXiv preprint arXiv:1805.11604*, 2018.

[41] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006.

[42] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2009.

[43] Thomas Sayre-McCord, Winter Guerra, Amado Antonini, Jasper Arneberg, Austin Brown, Guilherme Cavalheiro, Yajun Fang, Alex Gorodetsky, Dave McCoy, Sebastian Quilter, Fabian Riether, Ezra Tal, Yunus Terzioglu, Luca Carlone, and Sertac Karaman. Visual-inertial navigation algorithm development using photorealistic camera simulation in the loop. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[44] Nick Schneider, Lukas Schneider, Peter Pinggera, Uwe Franke, Marc Pollefeys, and Christoph Stiller. Semantically guided depth upsampling. In *German Conference on Pattern Recognition*, pages 37–48. Springer, 2016.

[45] Jianbo Shi and Carlo Tomasi. Good features to track. Technical report, Cornell University, 1993.

[46] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[47] Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical report, 2015.

[48] Jan Stühmer, Stefan Gumhold, and Daniel Cremers. Real-time dense geometry from a handheld camera. In *Joint Pattern Recognition Symposium*, pages 11–20. Springer, 2010.

[49] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Deep neuroevolution: genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567*, 2017.

[50] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[51] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. *arXiv preprint arXiv:1708.06500*, 2017.

[52] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance Normalization: The Missing Ingredient for Fast Stylization. *CoRR*, abs/1607.08022, 2016.

[53] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *IEEE Conference on computer vision and pattern recognition (CVPR)*, volume 5, page 6, 2017.

[54] Chaoyang Wang and José Miguel Buenaposada. Learning Depth from Monocular Videos using Direct Methods.

[55] Kaixuan Wang and Shaojie Shen. MVDepthNet: Real-time Multiview Depth Estimation Neural Network. In *2018 International Conference on 3D Vision (3DV)*, pages 248–257. IEEE, 2018.

[56] Rui Wang, Jan-Michael Frahm, and Stephen M Pizer. Recurrent Neural Network for Learning DenseDepth and Ego-Motion from Video. *arXiv preprint arXiv:1805.06558*, 2018.

[57] Sen Wang, Ronald Clark, Hongkai Wen, and Niki 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.

[58] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. *arXiv preprint arXiv:1812.07179*, 2018.

[59] Woodside Capital Partners and Yole Développement. The Automotive LiDAR Market. `http://www.woodsidecap.com/wp-content/uploads/2018/04/Yole_WCP-LiDAR-Report_April-2018-FINAL-2.pdf`, April 2018. [Online; accessed 9-January-2019].

[60] Nan Yang, Rui Wang, Jörg Stückler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *European Conference on Computer Vision*, pages 835–852. Springer, 2018.

[61] Cheng Zhao, Li Sun, Pulak Purkait, Tom Duckett, and Rustam Stolkin. Learning monocular visual odometry with dense 3D mapping from dense 3D flow. *arXiv preprint arXiv:1803.02286*, 2018.

[62] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *European Conference on Computer Vision*, pages 851–868. Springer, 2018.

[63] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017.