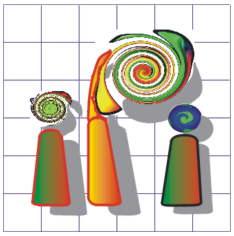


Gottfried Wilhelm Leibniz Universität Hannover
Institute of Photogrammetry and GeoInformation

Learning Aleatoric Uncertainty Estimation for Dense Stereo Matching End-to-End

In the course Geodesy and Geoinformation

Masterthesis

of

Dingxin Jin

Examiner:
Prof. Dr.-Ing. habil. Christian Heipke
Superviser:
Dr.-Ing. Max Mehltretter

Hannover, March 2022

Statement

I declare that this thesis has been composed solely by myself under the guidance of my supervisor. It has not been submitted, in whole or in part, to any examination authority. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own.

_____

Signature

_____

Place, Date

Abstract

In the research area of dense stereo matching, aleatoric uncertainty estimation is of increasingly large importance to reduce the influence of ambiguous results caused by the ill-posed nature of the problem. Deep learning-based models have been deployed to generate uncertainty measure from multiple data modalities like RGB images, predicted disparity map, cost volume etc. To deal with the bottleneck when concatenating learned features from different models, a new framework is proposed in this thesis enabling end-to-end training of a complex network combining multiple well-established CNN architectures for a fused uncertainty prediction. As the complementary task, a lightweight Cost-Volume-Analysis network with encoder-decoder structure to reduce its complexity is designed and evaluated based on different variants of uncertainty modeling. The effectiveness of end-to-end training strategy as well as the influence of features learned from cost volume are both quantitatively investigated via a series of experiments.

Keywords Dense Stereo Matching, Uncertainty Estimation, Deep Learning

Contents

1 Introduction

Extracting depth information from image sequences or videos is usually a basic step for solving high-level and complex computer vision tasks. Apart from classic 3D reconstruction applications, this technique is also widely investigated in the field of image recognition, object tracking, localization for mobile robots and self-driving cars.

Dense stereo matching is a classic task in photogrammetry and arguably the minimal case of structure-from-motion problem. Given epipolar rectified image pairs, the extracted geometric information of the scene is represented by a dense disparity map which explicitly gives a measure of pixel-wise depth. However, ambiguous solutions are usually unavoidable due to the ill-posed nature of projecting 2D points from the image plane back to the real-world 3D framework, resulting less reliable disparity estimates under challenging conditions such as occlusions, depth discontinuities, texture-less regions and so on. This difficult situation raises interest for the research of uncertainty quantification. The term uncertainty is also referred to as confidence in some literature, depending on the mathematical form used to interpret the uncertainty.

Considering the classic stereo matching pipeline (Scharstein and Szeliski, 2002), uncertainty cues are usually obtained from the matching cost volume generated in the cost computation step. Some properties closely related to uncertainty modeling are also extracted from the initial predicted disparity map and even the raw RGB (Hu and Mordohai, 2012). As most stereo matching solutions also include a disparity refinement step, the calculated uncertainty map could be further used to locate potentially mismatched pixels. In this way, challenging regions are refined based on reliable pixels so that the overall prediction accuracy is improved. Though a group of deep learning-based stereo matching approaches regress the disparity in an end-to-end manner, the uncertainty is still of great significance.

Starting from analyzing hand-crafted features or carefully selected feature combinations, early uncertainty estimation methodologies treated the design of metrics and classifiers as separate tasks. In recent times, the development of deep learning enables the joint training of feature extractors and classifiers. State-of-the-art architectures are able to regress confidence values from disparity patches (Poggi and Mattoccia, 2016), entire disparity map (Tosi et al., 2018) or cost volumes (Mehltretter

and Heipke, 2019). Attempts are also being made to fuse some of them to achieve better accuracy (Tosi et al., 2018). However, the effectiveness of an end-to-end fusion strategy integrating networks dealing with full-size cost volumes has never been investigated.

Thus, the objective of this thesis is on the development of an end-to-end CNN-based functional model estimating aleatoric uncertainty. The main contributions of this thesis are:

- A new framework estimating aleatoric uncertainty in an end-to-end manner utilising multiple modalities (estimated disparity, RGB and cost volume) is proposed. The advantages of end-to-end training is quantitatively evaluated.

- A new encoder-decoder-based CNN architecture estimating uncertainty from cost volume is built up. The performances of the new architecture as well as the influences of a larger receptive field are investigated.

- Experiments are conducted and the results are demonstrated regarding the aforementioned points.

In the following parts of the thesis, the theoretical foundations are first introduced in Chapter 2, including certain aspects like dense stereo matching, convolutional neural network and uncertainty modeling. Chapter 3 provides an thorough overview of the current status of the researches related to the scope of this thesis. A detailed description of the proposed general framework as well as the new CNN architectures are given in Chapter 4. The experimental settings are covered in Chapter 5, followed by the detailed illustrations and discussions of the obtained results in Chapter 6. The conclusions and an outlook for further investigations are made in Chapter 7.

2 Theoretical Background

This chapter provides fundamental concepts utilised further in the following parts of the thesis. First of all, a concise explanation of Dense Stereo Matching problem and its classic solutions are given (Section 2.1). This is followed by the description of basic structures in a modern CNN architecture and related concepts (Section 2.2). In the last section, different stochastic models applied for uncertainty quantification are discussed (Section 2.3).

2.1 Dense Stereo Matching

Dense Stereo Matching defines by its name the specific group of tasks in the domain of photogrammetric computer vision. Given two or more images capturing the same scene, the pixel-wise matching between images are established and used for the depth construction of the scene. Considering the minimal case of two epipolar rectified images, the relation between two corresponding pixels is quantitatively represented by a value d called disparity, which denotes the difference in the horizontal coordinate of two pixels. The search for a corresponding pixel and the optimal disparity is always constrained in the same row of image. A demonstration of dense stereo matching is shown in Figure 2.1.

The depth information could be directly computed from disparity. Given two epiploar rectified images, the mathematical relation between disparity d and depth z is described by the following equation:

$$z = \frac{fB}{d} \tag{2.1}$$

where f stands for the focal length of the cameras and B the distance between two camera centers.

Despite the simplification of problem description, the calculation of disparity remains a challenging task due to the ill-posed nature when projecting a 2D point to 3D space. Typical issues include occlusions, depth discontinuity, repetitive patterns, texture-less regions etc.

(a) left image (b) right image (c) disparity

Figure 2.1: An example of disparity estimation and depth reconstruction on stereo images. (Scharstein et al., 2014)

2.1.1 Classic Pipeline

The classic pipeline for a complete solution of dense stereo matching is defined by (Scharstein and Szeliski, 2002) as a four-step procedure (or a subset of them): *(1) Matching cost computation; (2) Cost aggregation; (3) Disparity computation/optimization; (4) disparity refinement.*

For the first step, the value of matching cost mathematically gives a measure of similarity between one pixel in the left image and another one in the paired right image. The computation of matching cost utilises the context information contained in the neighborhood that defined as a window centered at the same pixel. As the typical case for a stereo matching problem, the search of the corresponding pixel is constrained in the same horizontal line for the reason that the image pair has been calibrated to the epipolar line beforehand. Commonly used methods for this step include the absolute value of squared difference in intensity or census transformation. The latter will be illustrated in the next subsection. Modern CNNs are also applied to calculated the matching cost.

Under the assumption that neighboring pixels tend to have similar disparity values, the computed matching costs have to be aggregated to avoid noise and ambiguities so that a solid result could be generated. Therefore, the second step within the pipeline is as essential as the first step and could not be skipped in any cases.

For the third step, a winner-takes-all strategy simply chooses the global minimum disparity along the cost curve, while global approaches minimising the energy function are also popular like the semi-global matching (Hirschmuller, 2007). The final step includes further refinement methods that help increase the matching accuracy (Scharstein and Szeliski, 2002).

2.1.2 Census Transform

The most widely used non-learning-based approach to compute the matching cost is arguably the *Census Transform*, which is introduced by (Zabih and Woodfill, 1994) as a mapping of the local neighborhood surrounding a pixel P to a bit string representing the set of neighboring pixels whose intensity is less than that of P, as showed in Table 2.1.

131	85	43	1	1	0
131	71	27	1	P	0
1164	123	95	1	1	1

Table 2.1: An example of census transform with 3×3 window. The output string for pixel P is 11010111.

Given two pixels from left and right images along the same epipolar line and the corresponding census-transformed bit strings, the similarity is measured via Hamming distance denoting the number of bits that differ in the two bit strings.

One of the biggest advantages of census transform is the robustness to the illumination change, since the calculated matching cost only depends on the relative values instead of absolute ones. In the experimental parts of the thesis, the cost volume as well as its corresponding disparity used for evaluating the proposed methodology are all based on census transform.

2.1.3 Cost Volume

Cost Volume is the concept used to describe the special 3D volumetric data produced during matching cost calculation. As demonstrated in Figure 2.1, it shares the same height h and width w with the image pair and also has a depth along the d-axis according to the given disparity range. The vector in d-direction available for each pixel is also referred to as *Cost Curve*, containing the matching costs for the given pixel to the matching candidates in the paired image. In the remainder parts of the thesis, *Cost Volume* always refers to the one derived from the left image, though a cost volume for the right image is also available.

A cost volume not only serves as an intermediate output essential for the decision of optimal disparity values (Scharstein and Szeliski, 2002) but also provides important cues that are beneficial to the estimation of aleatoric uncertainty (Kendall et al., 2017; Mehltretter and Heipke, 2019).

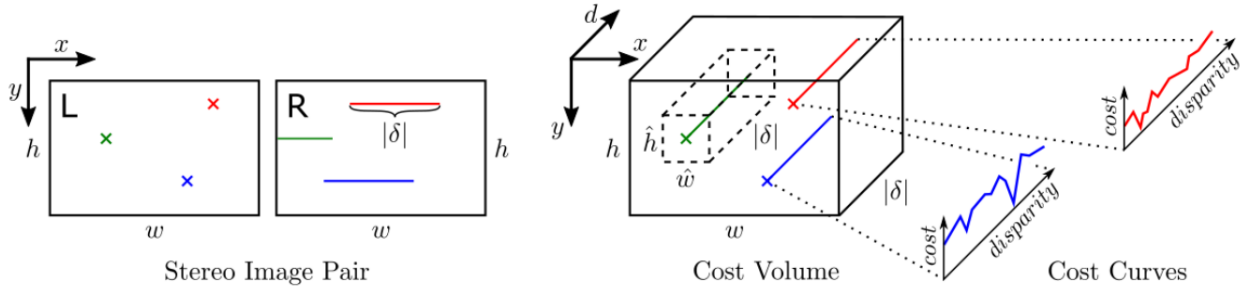


Figure 2.2: Relations among stereo image pair, cost volume and cost curves. Given an image pair both of size $h \times w$ and a pre-determined disparity range $|\delta|$, a cost volume refers to a volumetric data object of size $h \times w \times |\delta|$. For each pixel in the left image, a cost curve along the $d-$direction could be located in the cost volume according to the same x and y coordinates as the left image. A cost curve contains all the matching costs for the given pixel in left image and its candidates in the right image within the disparity range.

2.2 Convolutional Neural Network

Similar to other machine learning applications, a deep learning-based model mathematically defines a function:

$$\mathbf{y} = f(\mathbf{x}) \tag{2.2}$$

where the input $\mathbf{x}$ and output $\mathbf{y}$ are both vector elements. The interior structure of the model is composed of fundamental units that are described in the following part.

2.2.1 Basic Structure

Convolutional layer is the most fundamental component of a CNN architecture, which links a pixel of the output to a few number of pixels of the input using a pre-defined window called *filter kernel*. The parameters of a filter kernel keep unchanged when sliding along the x and y axis. In most cases, a convolutional layer consists of several kernels in order to extract sufficient contextual information for the generation of feature representations. Even though, the total number of parameters to be learned are significantly reduced compared with a fully-connected layer. Therefore, the development of deep networks with a large number of layers is possible. Deep networks are proved to have strong capabilities of extracting high-level semantic information from image and achieving higher accuracy.

Non-linearity refers to the activation functions which are also used in the general artificial neural

networks. By adding non-linearity, the output of a layer is not just a linear combination of the previous layer, contributing to the capability of the neural networks to approximate any functions. The mostly used non-linearity is arguably the *Rectified Linear Unit* (Nair and Hinton, 2010), which has the following mathematical form:

$$y = max(0, x) \tag{2.3}$$

In recent years the ReLU gains in popularity because it was found to greatly accelerate the convergence due to its non-saturating form. Compared to other activation function(i.g. sigmoid) where an exponential term is involved, it conduct cheaper calculations.

Batch normalization is the layer designed to help the model converge faster and avoid being stuck in the saturated regime. In practice, it's always combined with the following ReLU function to enhance the effectiveness.

Pooling Layer denotes the layer used for down-sampling the feature map. It also provides invariance to slightly different input image changes. A mostly used strategy downsampling is max pooling, where the maximal entry within a pooling window of a given size (i.g. 2x2 pixels) is extracted. Besides, convolutions with a stride larger than 1 also serves as a implicit pooling strategy.

Fully connected layers are often added to the end of a network architecture dealing with the extracted features representing the whole image, like in most image recognition applications. However, this is not the case for the dense prediction. On the contrary, the encoder-decoder structure is used, which is described in details in Section 2.2.3.

2.2.2 Training

A CNN model could be trained either supervised or unsupervised. The main distinction of supervised training from unsupervised one is the availability of ground truth. For a supervised training, the ground truth $\hat{y}$ is known for each training sample x and supposed to have the same format as the output y from the trained network given x. Typically for a dense prediction, the ground truth could be either sparse or dense, influencing the calculation of loss function. Relevant concepts regarding the training process include:

Loss function defines the objective function utilising the ground truth $\hat{y}$ and the predicted value y. The parameters of the network are optimized by minimizing the loss function. In terms of aleatoric uncertainty estimation, the determination of loss function depends on the uncertainty is modeled. This topic is detailed discussed in Section 2.3.

Back propagation refers to the techniques used for updating the parameters of CNN. For each iteration, the update begins at the very last layer according to the partial derivatives (gradients) calculated from the loss function and flows through the entire network to the top layer. An detailed explanation of back-propagation algorithm is not covered since it's out of the scope of the thesis.

Batch size describes the amount of training samples used for one update of parameters. Extreme cases include *Gradient Descent*, which utilises the entire dataset but is not always possible in practice due to the limitation of memory, and *Stochastic Gradient Descent*, which iterates on every single data sample and doesn't guarantee a stable convergence. As a result, the proper setting of batch size helps accelerate the convergence without causing memory problems.

Learning rate determines how much the parameters should be updated according to the calculated gradient. With a higher learning rate the optimization process is less likely to be stuck in a local minimum but also hard to converge. Therefore, a learning rate decay strategy is often used to learn a fine-tuned model.

2.2.3 Encoder-decoder Architecture

The concept of encoder-decoder networks was initially introduced for machine translation and sequence learning. It was first applied to the field of image processing as U-Net (Ronneberger et al., 2015) to handle the dense segmentation task. Encoders are responsible for producing a low dimensional dense representation for a given signal (image), while the decoders work recovering the signal lost by downsampling operations in the encoder. Encoder-decoder architectures form a subgroup of *Fully Convolutional Networks*, which means they predict dense outputs from arbitrary-sized inputs.

Figure 2.2 demonstrates the basic structure for a typical two-dimensional encoder-decoder networks, with the encoder in orange and decoder in yellow. The dotted black lines stands for the skip connections linking the layers of the same dimension from the encoder to the decoder. The skip connections are necessary to recover fine-grained information from downsampling layers.

2.2.4 Auxiliary Loss

There are basically two cases when an auxiliary loss is necessary. The first case is illustrated by Figure 2.3, especially for a very deep network. The auxiliary losses are hedged at certain layers to avoid vanishing gradients. In this case, the auxiliary losses can be replaced by other techniques like residual connections.

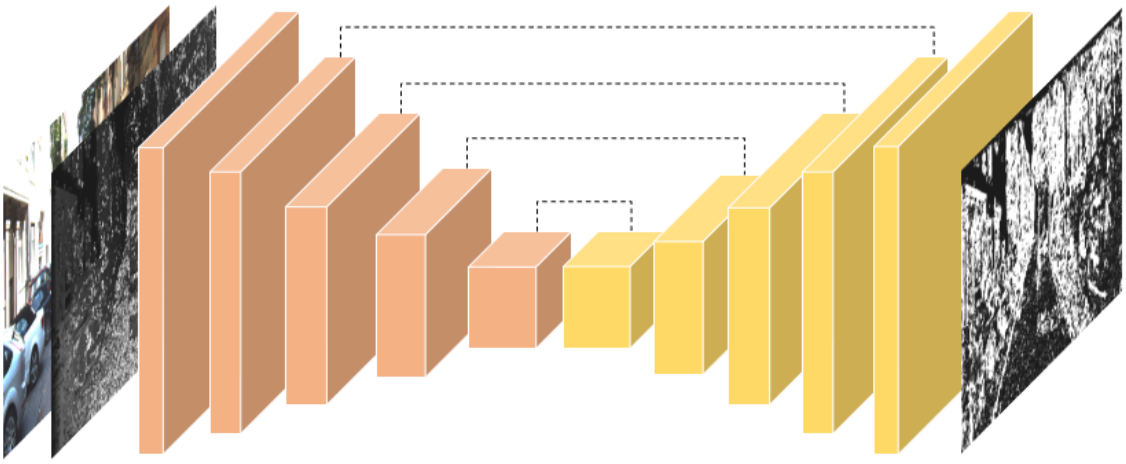


Figure 2.3: Illustration of a two dimensional encoder-decoder architecture, with encoder units in orange and decoder units in yellow. Dashed lines represent skip connections between blocks of the same resolution.

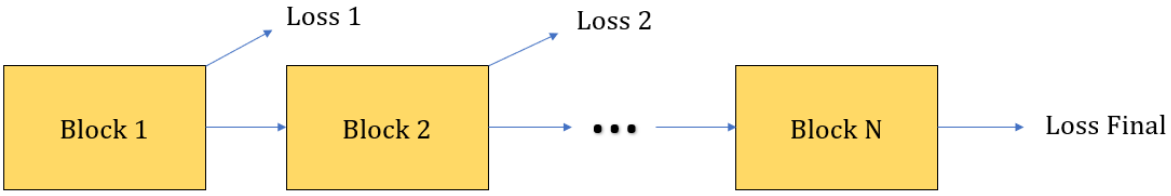


Figure 2.4: Auxiliary losses against vanishing gradients.

A more common application for the utilising of auxiliary loss is the multi-task learning. As shown in Figure 2.4, two different tasks could be combined in a single network architecture by sharing the top layers extracting features that are relevant for both tasks.

2.3 Uncertainty

Uncertainty is often classified into two categories: aleatoric and epistemic. Aleatoric uncertainty arises because of the unpredictable, random nature of the physical system under study, while epistemic uncertainty is due to the lack of knowledge of the system in respect to quantities and processes within the system. An uncertainty model answers two questions: (1) How the ground

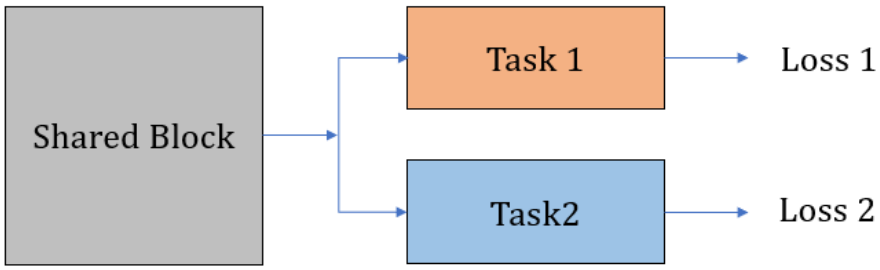


Figure 2.5: Auxiliary losses for multi-tasks learning.

truth for uncertainty is defined in a supervised deep learning process? (2) what type of loss functions should be applied given a functional model to be optimized? Therefore, two different models used in this thesis are given and explained in the following parts.

2.3.1 Confidence Model

In some literature, the uncertainty is often directly referred to as confidence. This is due to the popularity of using confidence model in the context of aleatric uncertainty estimation for dense stereo matching. A confidence measure is an assigned value representing the trust to a disparity estimate to be correct. A practical way to realize the idea is via a binary classification. Specially, the sigmoid non-linearity should be added to the final layer to output a confidence score γ.

The ground truth for the binary class labels can be derived from the error metric proposed in (Menze and Geiger, 2015): A disparity estimate d is assumed to be correct if either $\left|d - \hat{d}\right| < 3$ pixels or $\left|d - \hat{d}\right| < (\hat{d} \times 0.05)$, where $\hat{d}$ is the corresponding ground truth disparity. Using the predicted confidence score γ and the ground truth class labels $\hat{\gamma}$, the network is trained by minimizing the standard binary cross-entropy loss:

$$\mathcal{L}_{BC} = \frac{1}{N} \sum_{1}^{N} -\hat{\gamma} \cdot log(\gamma) - (1 - \hat{\gamma}) \cdot log(1 - \gamma) \tag{2.4}$$

where N stands for the total number of pixels with valid ground truth disparity.

2.3.2 Probabilistic Model

Probabilistic models are a sub-group of regression models that quantify the uncertainty as the magnitude of the error instead of measuring the probability of correctness. Another representative

example for the regression model is the residual model, which directly takes the difference between a pixel's estimated and its ground truth disparity as the ground truth of uncertainty to train the model.

As for a probabilistic model, the uncertainty is defined as variance or standard deviation of an presumed probabilistic distribution, while the absolute difference between ground truth and estimated disparity is used as an observation. Thus, the task of learning to predict aleatoric uncertainty could be interpreted in a Bayesian way, with the likelihood being maximized during the training process. Following this idea, the aleatoric uncertainty is learned in an implicit way, thus avoiding the need for a reference for the uncertainty.

Considering the widely used L1-norm in the context of training a CNN for the task of disparity regression (Kendall et al., 2017), the Laplace distribution is specified to describe the aleatoric uncertainty. Specially, the negative log likelihood of the distribution is utilised as the objective to enable the use of common optimizers:

$$-logp(\hat{d}_i|d_i) \propto \frac{\sqrt{2}}{\sigma_i}\left|d_i - \hat{d}_i\right| + log(\sigma_i) \tag{2.5}$$

where d and $\hat{d}$ represent respectively the estimated and ground truth disparity and σ the corresponding standard deviation of the presumed Laplace distribution for each pixel. In fact, the σ_i should be the uncertainty prediction made by the model. With the substitution $s = log(\sigma)$ the loss function is modified as:

$$\mathcal{L}_{Prob} = \frac{1}{N}\sum_{1}^{N}(\frac{\sqrt{2}}{exp(s_i)}\left|d - \hat{d}\right| + s_i) \tag{2.6}$$

where N denotes the total number of pixels with valid ground truth disparity. This formulation has the advantage of avoiding zero-valued denominators and being numerically more stable in the training process.

3 Related Work

This chapter outlines the development of techniques utilised for aleatoric uncertainty estimation in the context of dense stereo matching. In Section 3.1, approaches designed to obtain uncertainty features are discussed, with the main focus on the modern CNN-based methods that are closely related to the general idea of this thesis. In Section 3.2, mathematical models applied to interpret and quantify the uncertainty are concisely reviewed.

3.1 Uncertainty Feature Extraction

Different taxonomies of approaches estimating aleatoric uncertainty for DSM algorithms are applicable considering different aspects. According to (Mehltretter and Heipke, 2019), those solutions are divided into three groups: (1)hand-crafted features; (2)combined features; (3)deep learning. While the main criteria for such a division is the level of automation when extracting uncertainty features, another significant distinction comes from the data used for feature extraction, especially for deep learning-based methods. Different data inputs, which also called modalities in some literature, include estimated disparity map, initial RGB of the left image (sometimes also the right image) and the matching cost volume. Consequently, the CNN models are further divided into two groups: Disparity CNNs and Cost Volume CNNs. A brief overview of the traditional methods using individual or combined hand-crafted features is given in Section 3.1.1, while the following Section 3.1.2 and Section 3.1.3 set focus on deep learning-based strategies: Disparity CNNs using single or multiple modalities except cost volumes and Cost Volume CNNs dealing with cost volumes solely or in addition to other data inputs.

3.1.1 Classic Methods

Conventionally, the confidence for a disparity estimation is quantified upon a carefully selected set of hand-crafted features obtained via certain metrics. According to (Hu and Mordohai, 2012), those metrics are categorized into six main groups considering different properties: matching cost,

local properties of the cost curve, local minima of the cost curve, the entire cost curve, consistency between the left and right disparity maps, and distinctiveness-based measures. Usually, those metrics are only suited for the detection of specific problematic situations. For example, Matching Score Measure (MSM) is the best choice for occlusion detection, but it shows poor performance for discontinuities. As a result, the combination of features is essential for a precise uncertainty estimation.

Machine learning-based models have been applied at an early stage of confidence estimation to regress uncertainty values from combined features. For example, a random decision forest framework (Haeusler et al., 2013) has been built up dealing with feature vectors containing 23 handcrafted features, whose importance have been evaluated in terms of permutation importance. Further investigations (Park and Yoon, 2015) have taken more features into account.

3.1.2 Disparity CNNs

With the rapid growth of computer vision industries and growing popularity of deep learning-based applications, advanced solutions emerge in recent years, which are capable of learning the feature representation directly from the raw data.

Early attempts to apply CNN for a confidence measure of dense stereo matching are performed by CCNN (Poggi and Mattoccia, 2016), which aims to identify ambiguous areas from certain patterns repetitively appearing in the disparity map. Another solution called PBCP (Seki and Pollefeys, 2016) also adopted the similar strategy with an special hybrid design to accelerate the computation. Due to the limitation of information the disparity domain could provides, new ideas come up integrating other datatypes to the model to achieve better performance. EFN and LFN (Fu and Fard, 2018) are two variants unitizing RGB channels of the left image to enhance the performance of the disparity-only CNN. The main distinction is the fusion strategy adopted. While the Early Fusion Network (EFN) is directly fed with 4-channel RGB-D patches, a Late Fusion Network (LFN) extracts features independently from RGB and D and concatenate them in a single feature vector afterwards. A conclusion has been made that LFN achieves higher accuracy and has better generalization ability compared with EFN (Fu and Fard, 2018).

However, a patch-based network is not able to extract global context due to its small receptive field. This is an potential shortcoming typically when a colored image is available, which is beneficial for the location of problematic areas. Inspired by U-Net (Ronneberger et al., 2015), the ConfNet (Tosi et al., 2018) was introduced adopting an encoder-decoder architecture. Due to the smoothing effects caused by a large receptive field, it proves to be less accurate than the state-of-the-art local

method LFN.

3.1.3 Cost Volume CNNs

Though accurate results have been obtained via aforementioned approaches, the limitations of 2-dimensional CNN models should not be ignored. On the contrary, cost volumes contain well-structured geometric information calculated from paired images and provides more cues for uncertainty estimation than estimated disparity or RGB. In general, the approaches handling volumetric 3D data could be devided into two categories: projection-based and voxel-based.

In the field of dense stereo matching, end-to-end models for the task of disparity regression already exist dealing with the cost volume given or generated by the intermediate layers. For example, GC-Net (Kendall et al., 2017) adopt cost volume produced within the network.

Adopting CNN architectures to learn uncertainty features from cost volumes has been first tried by Kim et al. (2017). They have designed a top-K matching probability layer to project the initial cost volume to a probability cost volume with a fixed length in $d-$axis to solve the varying-size issue. In this way, the cost volume was reduced in size via sampling and normalization along the cost curve and fed to a 2D convolutional layer afterwards as a multi-channel feature map. The confidence measure was jointly made by the cost volume extractor and other sub-branches via a fusion network. Further models based on the basic idea of cost volume projection include UCF (Kim et al., 2018) and LAF (Kim et al., 2019), with the main distinctions on the fusion strategy and data inputs.

Considering the fact that cost curves in full resolution provide more cues of uncertainty, limitations exist for the projection-based methods. Thus, the Cost-Volume-Analysis Net (Mehltretter and Heipke, 2019) has been proposed to directly regress uncertainty values from the matching cost volume using 3D convolutions without any projection or reducing as pre-processing steps. Unlike approaches that only consider the single cost curve for each pixel (Gul et al., 2019), CVA-Net also takes into account neighboring pixels.

3.2 Uncertainty Modeling

Compared with functional models focusing on the extraction of features, less attentions have been paid to the interpretation and modeling of uncertainty. For deep learning-based approaches, the widely-used confidence model (Fu and Fard, 2018; Tosi et al., 2018) interprets the uncertainty as the

trust to a disparity to be correctly predicted, represented by a score between zero and one called confidence. A distinctive characteristic of learning-based uncertainty prediction is the implicit modeling of loss function, since the ground truth is not directly given. As for the confidence model, binary labels of correctness or incorrectness are generated by the error metric (Menze and Geiger, 2015) so that the pixel-wise binary classification is possible.

Another strategy for uncertainty modeling is to treat the uncertainty as a standard deviation or variance. The absolute difference between estimated and ground truth disparity has been used either directly as the ground truth for uncertainty (Mehltretter and Heipke, 2021) or as the observation of a probabilistic distribution (Kendall et al., 2017; Mehltretter and Heipke, 2021). For the latter, the uncertainty to be estimated is interpreted as one of the parameters defining a certain distribution, i.e. the Laplace distribution (Kendall et al., 2017).

4 Methodology

This chapter gives thorough explanations of the proposed methodology based on which the experiments are performed. An overview of the problem to be investigated is given in Section 4.1, followed by the introduction of general framework in Section 4.2. Detailed discussions on the specific branches contributing to the whole model are covered afterwards, including the local and global branches defined in LGC-Net (Section 4.3) as well as the modified CVA-Net (Section 4.4), which serves as the substitution of the original CVA-Net due to its time and storage efficiency. The loss function used for training is given in Section 4.5.

4.1 Problem Statement

The main focus of the thesis is set on the investigation of whether a CNN-based functional model benefit from an end-to-end architecture in terms of prediction accuracy while utilising multiple modalities. Typically for the task of aleatoric uncertainty estimation in the context of dense stereo matching, a tri-model input consisting of RGB, estimated disparity and cost volume is fed to the functional model to obtain an uncertainty map as the output. The cost volume and estimated disparity are not provided directly by most stereo datasets and should be calculated as a pre-processing step from the epipolar rectified image pair, which are simultaneously captured from the same scene. Thus, the specific dense matching algorithms should be specified in addition to the images used.

4.2 General Framework

The idea of developing an end-to-end model estimating confidence map from multiple modalities is derived from the LGC-Net proposed by Tosi et al. (2018). As illustrated in Figure 4.1, the LGC-Net combines the disparity map with two confidence maps predicted by the local and global sub-networks to form intermediate feature maps to be handled by the fusion network. Though solid

results have been proved, the LGC-Net still have some bottlenecks, which need further investigation and can be possibly improved in multiple ways.

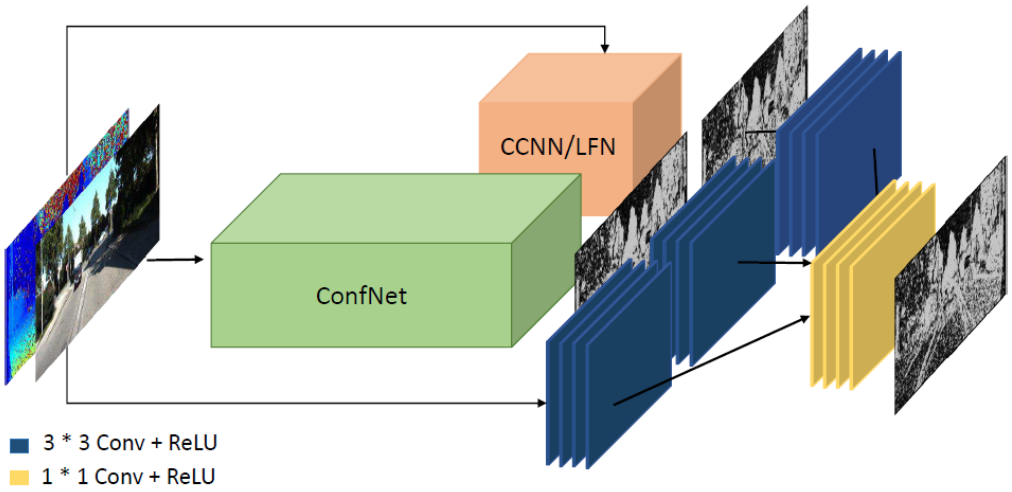


Figure 4.1: Basic architecture of LGC-Net. Given the reference RGB image and the estimated disparity, they are fed to both local and global sub-networks, whose outputs and the input disparity are processed by 3 independent towers and concatenated to predict the final confidence map. (Tosi et al., 2018)

The biggest issue within the architecture of LGC is arguably the potential loss of information when combining the respective features learned by local and global network. Since it adopts an strategy that only utilises the final uncertainty prediction as the feature representation, the fusion network is only allowed to perceive a two-dimensional feature vector for each pixel. As a result, the mutual influences of local and global networks may be not well quantified.

Another limitation of LGC is the input data it uses for training. A single disparity value represents only the positional index of the global optimum along the cost curve. Such a winner-take-all strategy abandoned mathematical descriptions of the other candidates which is critical for uncertainty quantification. This shortcoming could not be compensated simply by adding RGB-channels of the left image. Though the LGC-Net is originally designed to be a lightweight model, it's still worth figuring out whether a cost-volume would help the model achieve a better performance in terms of the forecast accuracy. Moreover, the two-stage training strategy applied in LGC-Net causes some unnecessary operational complexities in practice, which could be optimized in an end-to-end manner. So a new functional model is established, aiming to fix the aforementioned issues. Figure 4.2 defines the architecture of the general framework.

This framework explicitly gives a solution to all the problems proposed above. First of all, the

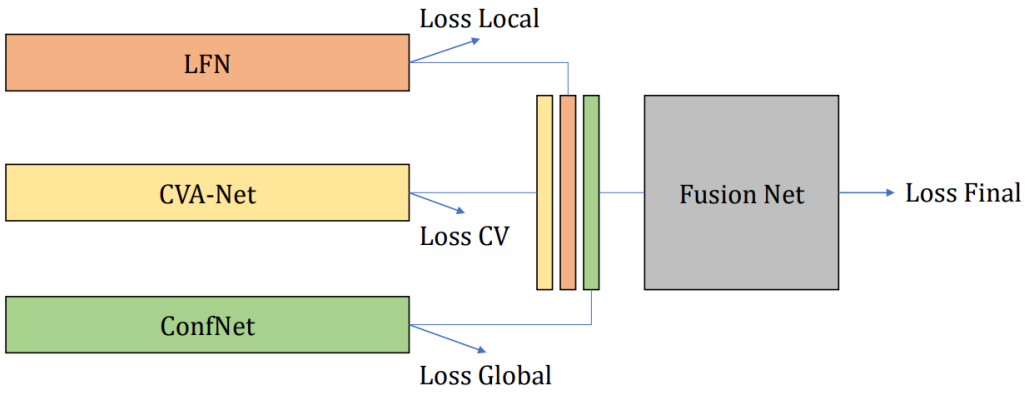


Figure 4.2: Basic structure of the new framework, where multi-channel feature maps generated by different sub-branches are concatenated to be processed by the fusion module. Auxiliary losses are linked to the corresponding blocks.

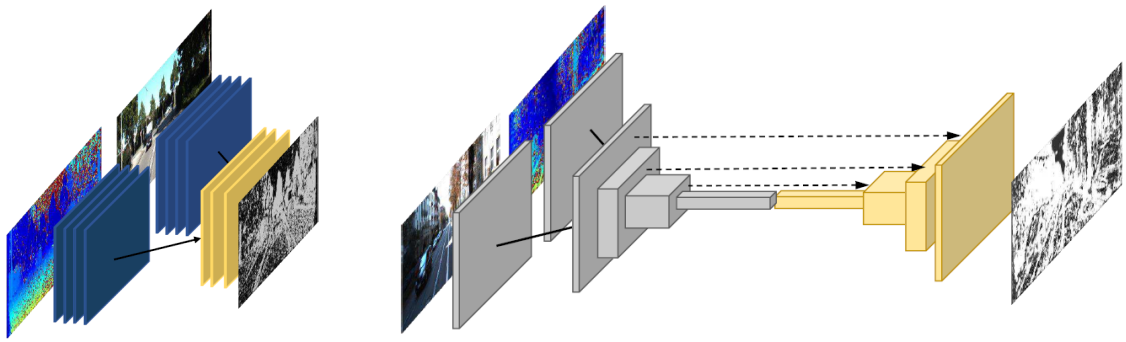
features extracted from raw data are concatenated at an earlier stage where the dimensions of vector haven't been reduced to one. Secondly, the CVA module also contributes features that are supposed to be invisible to disparity-based CNNs to the fusion module and thereby help enhance the performance of the whole model. Finally, it's an end-to-end design, which means there are no intermediate outputs available anymore.

In order to avoid potential overfitting issues and improve the generalization ability, an auxiliary loss is defined for each branch. All these losses along with the final loss are utilised during the training process. Besides, the auxiliary losses also provide an effective way to monitor the influences of each branch.

It should also be noted that different formats of training samples and other hyper-parameters defined respectively for each sub-network should be unified when trained jointly in an end-to-end manner, i.g. the input size, batch size, learning rate etc. New branches could also be added, making the model flexible and extensible.

4.3 Local and Global Branches

The architectures of local and global branches have already been defined by (Fu and Fard, 2018) and (Tosi et al., 2018). Handling the same data input, the main distinction between local and global approaches is the receptive field of the network. To train the model, image patches of size



(a) LFN (b) ConfNet

Figure 4.3: Architecture of the local (LFN) and global (ConfNet) branches. (Tosi et al., 2018)

9×9 are prepared as data samples for the local network, while random crops of size 256×512 are utilised for the global network. Specially, the late fusion strategy is adopted for the local branch to achieve better accuracy, which means the features are independently extracted from RGB image and disparity map before concatenation, as shown in Figure 4.3. And the encoder-decoder structure is essential for the ConfNet to integrating global information to the confidence estimation.

When training end-to-end, modifications have to be made to the LFN so as to keep its output feature maps of the same size as that of the ConfNet. Training LFN on random crops doesn't change the structure of LFN and its corresponding receptive field but only the batch size. The fusion module is a shallow network consisting of four 2D convolutional layers.

4.4 Modification of CVA-Net

As demonstrated in Figure 4.4, the initial Cost-Volume-Analysis Net (Mehltretter and Heipke, 2019) consists of three consecutive modules. For neighboring fusion, the dimension in x and y axis is gradually reduced to 1 to integrate local contextual information, resulting the *de facto* 1D convolutional operations in the depth processing stage learning the uncertainty measure along the entire cost curve. Unfortunately, the CVA-net could not be trained at a big batch size because of the large memory overhead caused by the size of cost volume. This problem also influences the testing process, which is performed in a slow pixel-wise manner instead of feeding the complete cost volume to the network.

Similar to *Late Fusion Network*, the initial CVA has to be transformed into a crop-based one so as to fit to the *ConfNet*. But it's not operational possible because of the aforementioned reasons. In

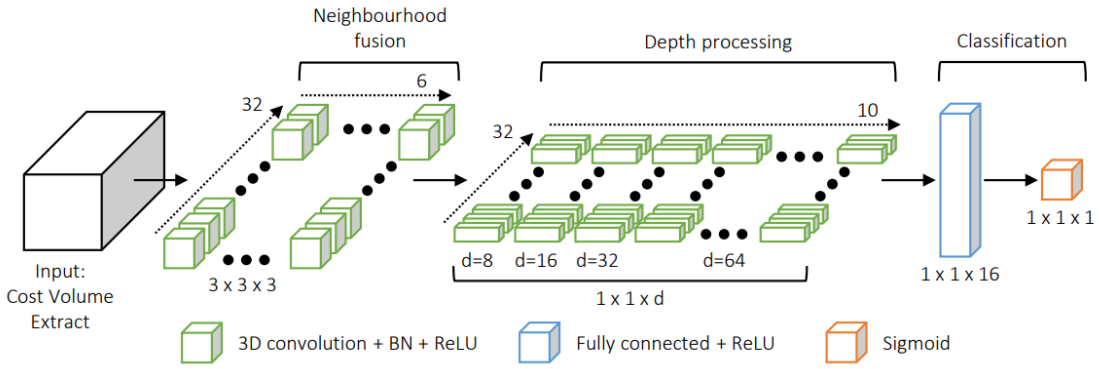


Figure 4.4: Architecture of the initial CVA-Net. (Mehltretter and Heipke, 2019)

fact, it only composes the first step of the modification and new structures (e.g. encoder-decoder structure) has to be integrated. Since a 3D encoder-decoder structure has already been applied to the GC-Net (Kendall et al., 2017) to conduct a end-to-end regression of dense matching, it's not a completely new idea.

Table 4.1 describes the detailed configuration of the modified CVA. To design such an architecture, specific considerations in the original CVA are followed to some extent. For example, the resolution-keeping convolutions (i.e. layer 2, 3, 5, 6) utilise one-dimensional kernels in z direction in order to learn features along the cost curve. With the kernel size of $3 \times 3 \times 3$ handling the neighborhood fusion, stride convolutions (layer 4, 7, 10, 13) down-samples the cost volumes in x and y directions but not always (only for layer 4 and 7) in z axis in order to keep as much information on the cost curve as possible.

The 3D encoder-decoder architecture enables the dense features to be represented in five resolutions with the smallest tensor being only 1/32 the height and with of the initial input. Skip connections are performed as an *add* operation of the two tensors with the same dimensions. It's worth noting that in the decoder the depth of tensor is kept unchanged as 32, so the layers in encoder has to be modified correspondingly.

A comparison between the modified and original CVA is conducted as the first experiment in Chapter 6.

4.5 Loss Function

The loss function used for updating the general model is described by the follwoing equation:

$$\mathcal{L}_{Total} = \alpha \cdot \mathcal{L}_{Local} + \beta \cdot \mathcal{L}_{Global} + \gamma \cdot \mathcal{L}_{CV} + \mathcal{L}_{Final} \tag{4.1}$$

where $\mathcal{L}_{Local}$, $\mathcal{L}_{Global}$, $\mathcal{L}_{CV}$ represent the loss of local branch, global branch and cost volume branch respectively. Along with the loss of fusion module $\mathcal{L}_{Final}$, the total loss $\mathcal{L}_{Total}$ is built up. Specially, α, β, γ defines the influence of each loss and are simply set to 1 since all branches produce similar accuracy.

Layer	Description	Output Tensor Dim.
Input	Cost Volume	256×512×256
1	3D conv., 3×3×3, 8 filters, stride 2×2×2	128×256×128
2	3D conv., 1×1×16, 8 filters, stride 1×1×2	128×256×64
3	3D conv., 1×1×16, 8 filters, stride 1×1×2	128×256×32
4	From 1: 3D conv., 3×3×3, 16 filters, stride 2×2×2	64×128×64
5	3D conv., 1×1×16, 16 filters	64×128×64
6	3D conv., 1×1×16, 16 filters, stride 1×1×2	64×128×32
7	From 4: 3D conv., 3×3×3, 32 filters, stride 2×2×1	32×64×64
8	3D conv., 1×1×32, 32 filters	32×64×64
9	3D conv., 1×1×32, 32 filters, stride 1×1×2	32×64×32
10	From 7: 3D conv., 3×3×3, 32 filters, stride 2×2×1	16×32×64
11	3D conv., 1×1×64, 32 filters	16×32×64
12	3D conv., 1×1×64, 32 filters, stride 1×1×2	16×32×32
13	From 10: 3D conv., 3×3×3, 64 filters, stride 2×2×1	8×16×64
14	3D conv., 1×1×64, 64 filters	8×16×64
15	3D conv., 1×1×64, 64 filters	8×16×32
16	3D transposed conv., 3×3×3, 32 filters, stride 2×2×1 Add layer 16 and 12	16×32×32
17	3D transposed conv., 3×3×3, 32 filters, stride 2×2×1 Add layer 17 and 9	32×64×32
18	3D transposed conv., 3×3×3, 16 filters, stride 2×2×1 Add layer 18 and 6	64×128×32
19	3D transposed conv., 3×3×3, 8 filters, stride 2×2×1 Add layer 19 and 3	128×256×32
20	3D transposed conv., 3×3×3, 1 filter, stride 2×2×1 (no ReLu or BN)	256×512×32
	3D conv., 1×1×32, sigmoid	256×512

Table 4.1: Details on the architecture of modified CVA-Net adopting a 3D encoder-decoder structure.

5 Experiments

In this chapter, the designs of the experiments are explained in detail. The objectives of the experiments are firstly given by Section 5.1, with a couple of essential questions to be discussed, according to which the whole experimental series are built up. This is followed by a thorough description of the datasets selected for training and testing in Section 5.2, as well as the relevant settings regarding the network in Section 5.3. At last, methods applied for the evaluation of the experimental results are introduced in Section 5.4.

5.1 Objectives

The main task of this thesis is the development of a new functional model being able to give a measure of the aleatoric uncertainty for a given disparity estimation in an end-to-end manner, and in the meanwhile taking advantages of multiple widely-used state-of-the-art architectures. The methodology has already been detailed described and comprehensively discussed in Chapter 4. In order to evaluate its effectiveness, specific aspects have to be considered. They are summarized as the following questions, according to which the experiments in Chapter 6 are performed and organized:

(1) Is the modified CVA-Net a good substitution of the original one?

Compared with the original CVA-Net, the changed version differs in several aspects. Beside the larger receptive field in the image plane, the depth processing along d-axis might also be insufficient due to the lost details when sub-sampling the cost curve. Therefore, its performance has to be evaluated as the first step for further applications.

(2) Does the end-to-end model have advantages compared with the old solution?

The assumption that more fused feature channels benefit the overall uncertainty regression needs further investigation. As the main focus of this thesis, the convergences of auxiliary losses are also analyzed jointly to figure out the influences of different branches to the final prediction.

(3) How does CVA-branch influence the overall performances?

A cost volume contains more clues relevant to the uncertainty modeling than a disparity estimate or a RGB image. In the scenario of end-to-end training, it's also possible to utilise the cost volume as a new modality. A comparison between networks with and without CVA-Net gives an answer to this question.

5.2 Datasets

Two different real-world datasets are referred to and utilised to train and test the proposed networks. Both of the datasets provide rectified stereo image pairs taken by a calibrated stereo camera. The KITTI dataset is used for training and testing, and Middlebury-v3 for cross-validation.

The KITTI dataset (Menze and Geiger, 2015) is recorded from a moving platform while driving in an urban environment. Therefore, the scenes and objects presented in the images are mainly vehicles, traffic signs, different buildings as well as trees and bushes. A drawback of KITTI dataset is the sparsity of ground truth due to the limitation of LiDAR measurement, from which the ground truth values are obtained. KITTI-12 contains 194 stereo image pairs and KITTI-15 provides 200 more.

The Middlebury-v3 dataset (Scharstein et al., 2014) contains 15 images taken from indoor scenes. The scenes are carefully selected showing specific challenges faced by stereo matching algorithms. Contrary to KITTI dataset, the ground truth is densely given with sub-pixel accuracy, making itself an ideal option for cross-validation.

5.3 Training and Test Settings

Most of the functional models to be compared in the following experiment series, including modified CVA-Net, LGC, LGC-E (end-to-end version of LGC without CVA-branch) and LGCV (the proposed general model integrating CVA-branch), are trained on a mixture of 20 images from KITTI-12 and 20 images from KITTI-15. For technical reasons, the original CVA-Net is trained only on 10 images from KITTI-12 and 10 images from KITTI-15 but the results are good enough to be compared. The validation set is composed of two images from KITTI-15. The corresponding cost volume and disparity estimate for each image are computed via census transform.

The training and validation samples of initial CVA-Net are cost volume extracts of size $13 \times 13 \times 256$

centered at pixels where the ground truth is available. As the pre-processing step, all the samples are loaded to the memory before training, which means the number of images used for training should not be too large. Similar strategies are also followed by the local and fusion sub-networks in LGC utilizing image patches of size 9×9. As for the modified CVA-Net, cost volumes are loaded dynamically for each iteration and thus reduce the memory requirements, making it possible to utilise more images for training. Similar to the training of ConfNet, a cost volume extract of size $256 \times 512 \times 256$ is randomly cropped from the entire cost volume and fed to the network as a data sample. Image patches are also replaced by random crops of size 256×512 for the LFN sub-branch in LGC-E and LGCV developed in the thesis.

All the aforementioned networks are implemented on TensorFlow 2.1. Relevant hyperparameters are summarised in Table 5.1.

Optimiser	Adam
Kernel initialiser	Glorot uniform
Learning rate	10^{-4}
Batch size	128 (LGC & initial CVA) or
	1 (LGC-E, LGCV, modified CVA)

Table 5.1: Common hyperparameters for original and modified CVAs, sub-networks of LGC, LGC-E and LGCV.

Testing results are obtained from 50 images of KITTI-15 and 15 images of Middlebury and evaluated respectively. During testing, full-size images and cost volumes are fed to the trained model since the whole network is fully convolutional and able to deal with inputs of any size larger than the size of training crops.

5.4 Evaluation Strategy

5.4.1 AUC

The first metric used for the comparison between different functional models is Area Under the Curve (AUC) value, which could be calculated from the corresponding Receiver Operating Characteristic (ROC) curve. This is a well-established approach to evaluate the performance of a binary classifier, based on which most of the confidence models are built up.(Hu and Mordohai, 2012). To obtain ROC, a percentage list containing numbers between 0 and 1 in an ascending order should be pre-defined. Based on each value within the list, a certain percentage (i.e. 10%) of pixels with valid

ground truth and the highest confidence (lowest uncertainty) are extracted from the disparity map and the corresponding error rates are calculated as the error labels are known given the ground truth. The AUC refers to the area under the obtained ROC curve.

For a given disparity estimate, the optimal AUC could be determined by the following equation:

$$AUC_{opt} = \int_{1-\epsilon}^{1} \frac{p-(1-\epsilon)}{p} dp = \epsilon + (1-\epsilon)\ln(1-\epsilon) \tag{5.1}$$

where p denotes the percentage of pixels sampled from the disparity map and ϵ represents the overall error. A confidence map estimated by deep CNN models usually contains a large number of pixels with the same highest confidence, resulting high discretisation errors when generating the ROC curve. Consequently, the calculated AUC of a given confidence map could be lower the the optimal value in some cases. To deal with this issue, the interval-based ROC is calculated according the new sampling strategy (Mehltretter and Heipke, 2021).

5.4.2 Correlation Coefficient

For the evaluation of probabilistic model, AUC metric has two drawbacks. Firstly, the difference between ground truth and estimated disparity is replaced by a binary label. Secondly, the sampling strategy based on the percentage list only considered the relative order of the estimated uncertainty values instead of the absolute ones. This is not suitable for a probabilistic model, which interprets the uncertainty as the parameter of a given distribution. Thus, an alternative metric called *Pearson Correlation Coefficient* is utilised to overcome this issue, which measures the linear dependence between two stochastic variables X and Y. The mathematical form is given as:

$$\rho(X,Y) = \frac{\sum_{i=1}^{N}(X_i-\mu_X)(Y_i-\mu_Y)}{\sqrt{\sum_{i=1}^{N}(X_i-\mu_X)^2 \sum_{i=1}^{N}(Y_i-\mu_Y)^2}} \tag{5.2}$$

where μ stands for the mean of the corresponding variable and N for the number of observations. In terms of a probabilistic model, X and Y represent respectively the collection of estimated uncertainty values and the numerical differences between ground truth and estimated disparity. The higher the correlation, the better the predicted uncertainty.

5.4.3 Region Mask

In order to compare the performances of models on different challenging areas, evaluation results are demonstrated in four groups: (1)Regions without occlusions; (2)Regions with occlusions; (3)Regions of discontinuous disparity; (4)Texture-less regions. Therefore, binary region masks are introduced

to filter the ground truth, which is calculated beforehand from either initial RGB or ground truth disparity. For example, the texture-less areas refer to the regions where intensity in RGB-channels changes extremely slowly among neighbouring pixels. Regions of discontinuous disparity, however, are generated from the ground truth disparity, filtering out the areas where the gaps between neighbouring pixels in terms of the disparity value are larger than the pre-defined threshold. For KITTI benchamark, where the ground truth labels are not given densely, binary masks for depth discontinuous regions are not available.

6 Results

This chapter analyses the experimental results and answers the questions proposed in Chapter 5. The performances of the modified CVA-Net are firstly evaluated in Section 6.1, while the effects and characteristics of the end-to-end LGC (LGC-E) are covered in Section 6.2. The influences of cost volumes on the overall performances of the network are included in Section 6.3.

6.1 Performance of Modified CVA-Net

To evaluate the performance of modified CVA-Net, a comparative experiment between the initial and modified CVA-Net is performed. For this experiment, both confidence model and probabilistic model are utilised.

6.1.1 Confidence Model

The quantitative comparison between initial and modified CVA is illustrated in Table 6.1 with respect to the AUC. The results of cross-validation on Middlebury dataset are also given. A gap between the performances of initial and modified CVA can be observed, since a trade-off exists between accuracy and memory consumption to design the new architecture. In some case, the modified CVA performs as well as the initial CVA, while the initial CVA is better at dealing with challenging regions such as texture-less areas. The cross-validation results also suggest the initial CVA has stronger generalization abilities. In general, the difference of AUC is around 1×10^{-2}, which means the modified CVA is a good substitution for the initial one to be integrated in the following experiments.

KITTI-15			
avg. AUC = $10^{-2}\times$	Opt.	Init.	**Mod.**
Without Occ.	7.83	9.75	9.85
With Occ.	8.17	10.07	10.20
Discontinuous	-	-	-
Textureless	18.79	21.24	22.42

Middlebury-v3			
avg. AUC = $10^{-2}\times$	Opt.	Init.	**Mod.**
Without Occ.	4.53	6.21	7.35
With Occ.	6.97	9.04	10.28
Discontinuous	8.80	16.99	17.49
Textureless	10.92	12.94	13.94

Table 6.1: Comparison of initial and modified CVA-Net in terms of average AUC. Models are tested on both KITTI-15 and Middlebury datasets. Performances are also evaluated on specific challenging areas.

6.1.2 Probabilistic Model

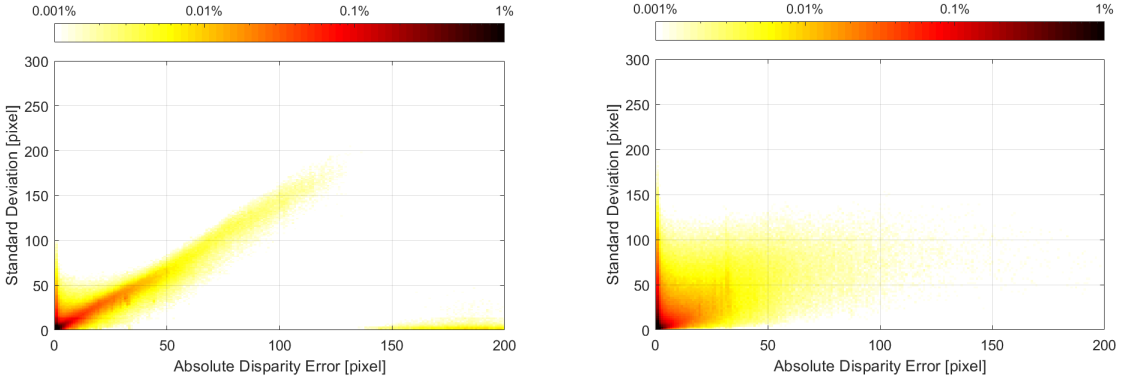
According to the experimental results, as shown in Figure 6.1 and Figure 6.2, the original CVA-Net achieves better results than the modified version on both KITTI-15 and Middlebury datasets. However, the gap is not as large as what the plots have suggested in terms of the correlation coefficient (0.67 vs 0.66 for KIITI-15 and 0.7 vs 0.63 for Middlebury). A possible account for this phenomenon is that relative small uncertainty values have been predicted by the original CVA for the pixels with high absolute disparity error, which are reflected by the unexpected high density appearing in the bottom right of the plot. Those pixels negatively effect the overall linear dependence between absolute error and uncertainty.

6.1.3 Discussion

Although the experimental results look not too bad on both confidence and probabilistic models, the encoder-decoder-based modified CVA-Net still has significant drawbacks compared to the original one, which are mainly caused by the down-samplings along the depth of cost volume. However, such an operation is unavoidable if the size of intermediate tensors has to been reduced. Therefore, further improvements can be made to keep more details along the depth by modulating the structure of the network.

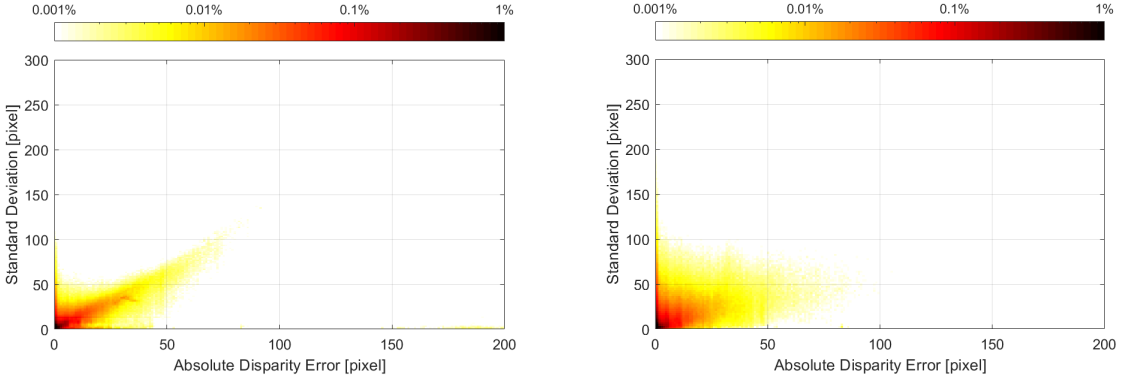
6.2 Performance of End-to-end Architecture

The effectiveness of end-to-end training strategy is verified by the comparison between LGC and LGC-E. Both networks fuse the features learned from LFN and ConfNet, but in different ways.



(a) original CVA ($\rho = 0.67$) (b) modified CVA ($\rho = 0.66$)

Figure 6.1: Absolute error uncertainty relation evaluated on 50 images of the KITTI-15 dataset. The logarithmic colour scale represents the percentage of pixels showing the respective error and estimated uncertainty. The variable ρ stands for the correlation coefficient achieved for the respective network.



(a) original CVA ($\rho = 0.7$) (b) modified CVA ($\rho = 0.63$)

Figure 6.2: Absolute error uncertainty relation evaluated on 15 images of the Middlebury dataset. The logarithmic colour scale represents the percentage of pixels showing the respective error and estimated uncertainty. The variable ρ stands for the correlation coefficient achieved for the respective network.

The convergences of multiple losses hedged to LGC-E are analysed as the first step. Since the The evaluation of LGC-E is performed on the confidence model only.

6.2.1 Convergence

Figure 6.3 shows the changes of validation losses when training LGC-E. Based on the same training configuration, the distinctions between local and global approaches are obvious: while a trend of overfitting could be observed after around 200 epochs for both global and final losses, the local loss converges slower and doesn't change too much after reaching the optimum. For a smaller size of training set, the global and fusion sub-network are more likely to be overfitted and an early stop is necessary.

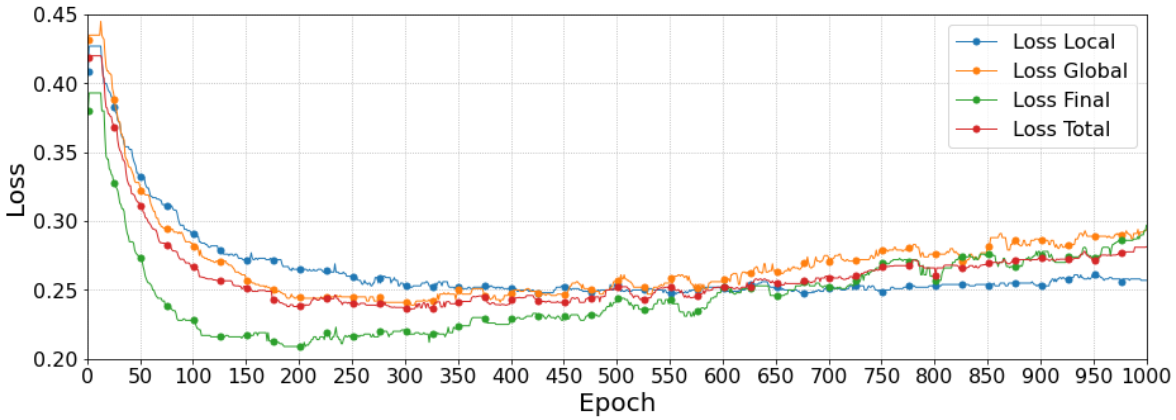


Figure 6.3: Convergence of validation losses for LGC-E. The initial Loss Total has been divided by three to share the same scale with other losses.

6.2.2 Comparison to LGC

According to the testing results on KITTI-15, LGC-E doesn't show a higher accuracy in any case compared with the initial LGC. A possible reason for the outcome is the different characteristic between local and global sub-networks. As shown in Figure 6.3, the global and fusion module converge quicker than the local one, which means the features extracted from local sub-network should not be the optimal when the early stop has been made.

However, the cross-validation results on Middlebury prove the potential of LGC-E for a better performance. According to the results obtained in section 6.3, the global branch performs better for KITTI-15 dataset but worse for Middlebury images compared with the local branch. The

KITTI-15			
avg. AUC = 10^{-2}	Opt.	LGC	**LGC-E**
Without Occ.	7.83	8.36	8.97
With Occ.	8.17	8.76	9.41
Discontinuous	-	-	-
Textureless	18.79	19.66	21.85

Middlebury-v3			
avg. AUC = 10^{-2}	Opt.	LGC	**LGC-E**
Without Occ.	4.53	7.29	6.57
With Occ.	6.97	10.61	9.51
Discontinuous	8.80	18.55	15.81
Textureless	10.92	13.56	13.16

Table 6.2: Comparison of original and end-to-end LGC in terms of AUC. Models are tested both on KITTI-15 and Middlebury datasets. Performances are also evaluated on specific challenging areas.

different sensibilities of local and global branches to the domain gap are learned to some extent by LGC-E to help improve the overall accuracy.

6.3 Influence of Cost Volume

The influence of cost volume is evaluated by comparing LGC-E with the general framework LGCV discussed in Chapter 4.2, which is established by integrating the modified CVA to LGC-E. A new auxiliary loss is added correspondingly to the CVA branch. The convergence is illustrated in Figure 6.2.

6.3.1 Convergence

Combining the feature representations learned from CV, the final loss achieves a value lower than 0.2, suggesting a better overall accuracy than LGC-E. Even though the loss CV itself is always the highest during the whole training process, meaning less precision than local and global sub-network, it still provides clues essential for a correct uncertainty estimation.

6.3.2 Comparison

Apart from the final confidence measure, each sub-network also contributes its prediction based on the auxiliary loss. An example of outputs is given in Figure 6.5. The evaluation results for all those sub-networks are summarized in terms of AUC in Table 6.3. Specially, the results obtained from LGC-E are also listed to be compared.

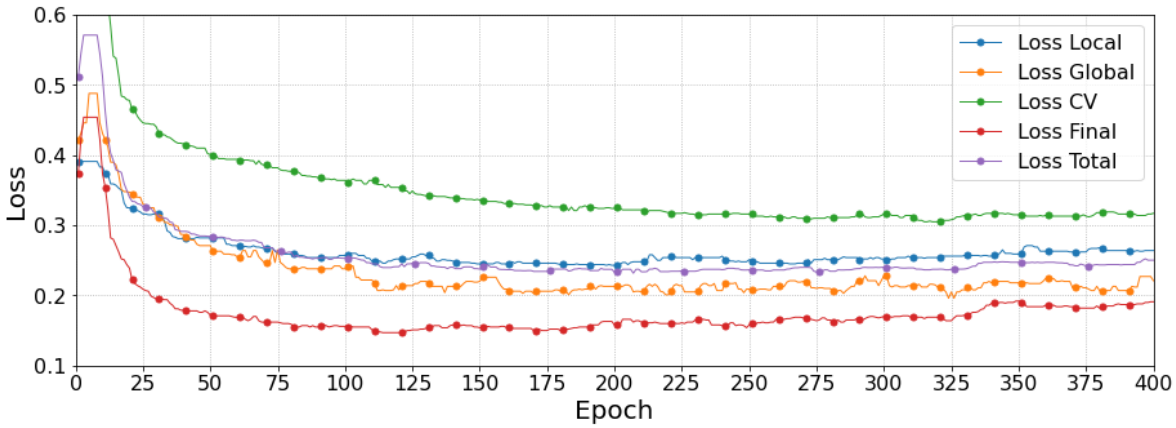


Figure 6.4: Convergence of validation losses for the general framework (LGCV). The initial Loss Total has been divided by four to share the same scale with other losses.

The final prediction prove itself to be the best compared with other outputs in most cases. The only exception occurs when dealing with disparity discontinuity in Middlebury-v3 dataset. A conclusion could be made that the overall accuracy has been significantly proved with the modified CVA-Net integrated. Compared between two datasets, it's also worth noting the global sub-network shows better results in KITTI benchmark than in Middlebury-v3 considering the relative value compared to the local branch. This is probability because of the domain gap between two datasets and the global branch is more sensitive to it since it has larger receptive field.

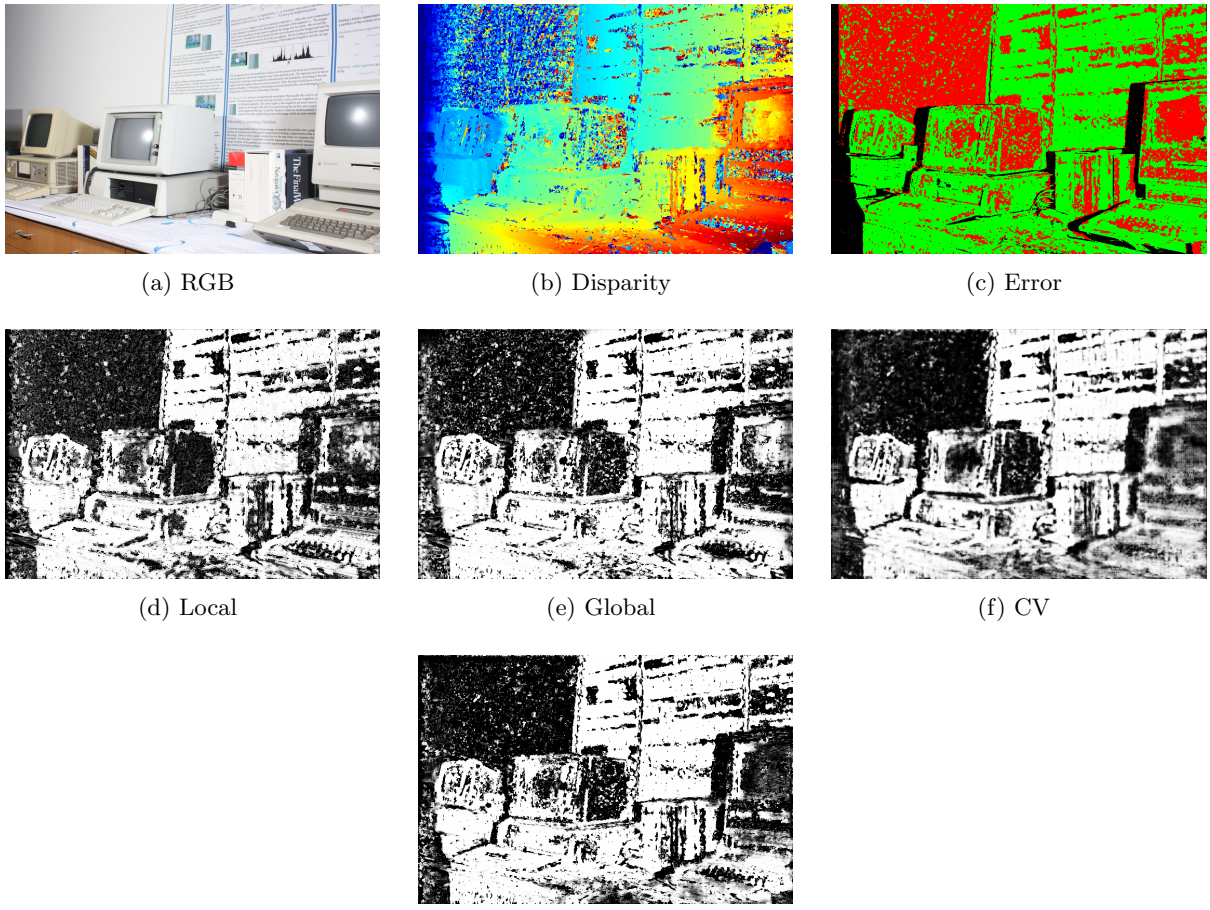
KITTI-15

avg. AUC = 10^{-2}	Opt.	Local	Global	LGC-E	CV	**Final**
Without Occ.	7.83	8.79	8.22	8.97	10.39	**7.92**
With Occ.	8.17	9.21	8.62	9.41	10.76	**8.28**
Discontinuous	-	-	-	-	-	-
Textureless	18.79	21.45	19.81	21.85	23.59	**18.90**

Middlebury-v3

avg. AUC = 10^{-2}	Opt.	Local	Global	LGC-E	CV	**Final**
Without Occ.	4.53	6.39	7.15	6.57	7.14	**6.10**
With Occ.	6.97	9.35	10.44	9.51	9.96	**9.03**
Discontinuous	8.80	**15.42**	17.80	15.81	16.31	16.60
Textureless	10.92	12.90	13.75	13.16	14.01	**12.79**

Table 6.3: Comparison between different outputs from the general framework, including local, global, cv, and final branches. The results for LGC-E are also listed.



(a) RGB (b) Disparity (c) Error

(d) Local (e) Global (f) CV

(g) Final

Figure 6.5: An example from Middlebury dataset, containing RGB image, estimated disparity, error map and different confidence measures obtained from sub-networks. Jet colour-map is used for the visualisation of disparity map with blue-end denoting the largest disparity. Green and red labels stand for the correct and incorrect disparity estimates in the error map and black labels for the pixels without reference of ground truth. In the confidence maps, brighter pixels denotes larger confidence values.

7 Conclusion and Outlook

This thesis proposed a new framework for the estimation of aleatoric uncertainty in terms of dense stereo matching, enabling the end-to-end training of a complex network combining multiple well-established CNN architectures dealing with different modalities. To train the entire network, original training configurations for different sub-networks were unified to fit to each other. As a result, the new version of Cost-Volume-Analysis network architecture was designed to be integrated to the proposed framework in a technically possible way. Utilising 3D encoder-decoder structure, the lightweight CVA-Net differed from the local approach and extracted global contextual information with a larger receptive field.

To evaluate the effectiveness of new CVA-Net as well as the general framework, comparative experiments have been designed and performed between different architectures or training configurations. Quantitative evaluation strategies includes AUC values on confidence model and correlation coefficients on probabilistic model. Despite distinctions in characteristics, the modified CVA-Net achieved almost as good performances as the original one under certain circumstances based on statistics. Though the end-to-end training strategy itself didn't show outstanding improvements, the contribution of cost volume sub-branch has been proved to be not trivial to the high accuracy of the final uncertainty measure.

Future studies could be conducted on the further optimization of the encoder-decoder-based CVA-Net with respect to the accuracy and memory overhead. New sub-networks could also be integrated to the general framework utilising other modalities and adding new uncertainty cues. Moreover, an investigation on the different coefficients combining multiple auxiliary losses are also reasonable.

Bibliography

Fu, Z. and Fard, M. A., 2018. Learning confidence measures by multi-modal convolutional neural networks. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, pp. 1321–1330.

Gul, M. S. K., Bätz, M. and Keinert, J., 2019. Pixel-wise confidences for stereo disparities using recurrent neural networks. In: *BMVC*, p. 23.

Haeusler, R., Nair, R. and Kondermann, D., 2013. Ensemble learning for confidence measures in stereo vision. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 305–312.

Hirschmuller, H., 2007. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence* 30(2), pp. 328–341.

Hu, X. and Mordohai, P., 2012. A quantitative evaluation of confidence measures for stereo vision. *IEEE transactions on pattern analysis and machine intelligence* 34(11), pp. 2121–2133.

Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A. and Bry, A., 2017. End-to-end learning of geometry and context for deep stereo regression. In: *Proceedings of the IEEE international conference on computer vision*, pp. 66–75.

Kim, S., Kim, S., Min, D. and Sohn, K., 2019. Laf-net: Locally adaptive fusion networks for stereo confidence estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 205–214.

Kim, S., Min, D., Ham, B., Kim, S. and Sohn, K., 2017. Deep stereo confidence prediction for depth estimation. In: *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, pp. 992–996.

Kim, S., Min, D., Kim, S. and Sohn, K., 2018. Unified confidence estimation networks for robust stereo matching. *IEEE Transactions on Image Processing* 28(3), pp. 1299–1313.

Mehltretter, M. and Heipke, C., 2019. Cnn-based cost volume analysis as confidence measure for dense matching. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*.

Mehltretter, M. and Heipke, C., 2021. Aleatoric uncertainty estimation for dense stereo matching via cnn-based cost volume analysis. *ISPRS Journal of Photogrammetry and Remote Sensing* 171, pp. 63–75.

Menze, M. and Geiger, A., 2015. Object scene flow for autonomous vehicles. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3061–3070.

Nair, V. and Hinton, G. E., 2010. Rectified linear units improve restricted boltzmann machines. In: *Icml*.

Park, M.-G. and Yoon, K.-J., 2015. Leveraging stereo matching with learning-based confidence measures. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 101–109.

Poggi, M. and Mattoccia, S., 2016. Learning from scratch a confidence measure. In: *Bmvc*, Vol. 2, p. 4.

Ronneberger, O., Fischer, P. and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical image computing and computer-assisted intervention*, Springer, pp. 234–241.

Scharstein, D. and Szeliski, R., 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision* 47(1), pp. 7–42.

Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X. and Westling, P., 2014. High-resolution stereo datasets with subpixel-accurate ground truth. In: *German conference on pattern recognition*, Springer, pp. 31–42.

Seki, A. and Pollefeys, M., 2016. Patch based confidence prediction for dense disparity map. In: *BMVC*, Vol. 2, p. 4.

Tosi, F., Poggi, M., Benincasa, A. and Mattoccia, S., 2018. Beyond local reasoning for stereo confidence estimation with deep learning. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 319–334.

Zabih, R. and Woodfill, J., 1994. Non-parametric local transforms for computing visual correspon-

dence. In: *European conference on computer vision*, Springer, pp. 151–158.