

Improved Stereo Matching with Constant Highway Networks and Reflective Confidence Learning

Amit Shaked¹ and Lior Wolf^{1,2}

¹The Blavatnik School of Computer Science, Tel Aviv University, Israel

²Facebook AI Research

Abstract

We present an improved three-step pipeline for the stereo matching problem and introduce multiple novelties at each stage. We propose a new highway network architecture for computing the matching cost at each possible disparity, based on multilevel weighted residual shortcuts, trained with a hybrid loss that supports multilevel comparison of image patches. A novel post-processing step is then introduced, which employs a second deep convolutional neural network for pooling global information from multiple disparities. This network outputs both the image disparity map, which replaces the conventional “winner takes all” strategy, and a confidence in the prediction. The confidence score is achieved by training the network with a new technique that we call the reflective loss. Lastly, the learned confidence is employed in order to better detect outliers in the refinement step. The proposed pipeline achieves state of the art accuracy on the largest and most competitive stereo benchmarks, and the learned confidence is shown to outperform all existing alternatives.

1. Introduction

The modern pipeline for stereo matching, which achieves state of the art results on the most challenging benchmarks, contains a deep neural network for computing the matching score, and a few heuristic post-processing steps. The main goal of these processing steps is to incorporate spatial information in order to verify the plausibility of the proposed matching and to selectively smooth and refine the obtained results.

We present methods that improve the deep matching network by incorporating, among other improvements, a variant of highway networks [32] with a multilevel skip-connection structure and a gating signal that is fixed at a constant. Analysis of this architecture is provided, and ex-

perimental results are shown to support its advantages over many existing Residual Network alternatives [16, 6, 18, 32].

To compute the disparity image, a second network is introduced to replace the “winner takes all” (WTA) rule, and is currently applied to obtain both the predicted disparity at each position and the confidence in this result as a separate output.

A confidence measure for the disparity prediction has been the subject of considerable study. More generally, assessing the correctness of a CNN output is also a subject that is under magnifying glass. We propose a new training signal, which we call the *reflective loss*. The labels used for this loss are changing dynamically, based on the current success of the CNN on each training sample. In our case, when the disparity predicted by the network for a given example during training is correct, the target confidence label of the sample is 1, otherwise 0.

The obtained confidence is a crucial part of the subsequent refinement step of the pipeline, in which we detect incorrect disparity predictions and replace them by interpolating neighboring pixels.

The contributions of this paper are: (i) In Sec. 3 we present a new highway network architecture for patch matching including multilevel constant highway gating and scaling layers that control the receptive field of the network. (ii) This network is trained with a new hybrid loss (Sec. 3.3) for better use of the description-decision network architecture. (iii) Computing the disparity image by using a CNN instead of the previously suggested WTA strategy, as depicted in Sec. 4. (iv) In Sec. 4.2 we introduce a novel way to measure the correctness of the output of a CNN via reflective learning that outperforms other techniques in the literature for assessing confidence in stereo matching. (v) In Sec. 5 we show how to use this confidence score for better outlier detection and correction in the refinement process. (vi) Achieving the best results on the KITTI 2012 and KITTI 2015 stereo data sets with an error rates of 2.27 and 3.42, respectively, improving the 2.43 and 3.89 of the MC-

CNN[36] baseline. (vii) Improving the fast architecture to achieve the best results on the KITTI 2012 and KITTI 2015 for methods that run under 5 seconds, with an error rate of 2.63 and 3.78 on these benchmarks, in comparison to the 2.82 and 4.62 baseline. (viii) An open source code¹ for easily using and modifying the pipeline.

2. Related work

Computing the matching cost via convolutional neural networks was firstly introduced by Zbontar and LeCun [35, 36]. This pipeline was subsequently modified: [21] reduced the computation time with only minor reduction to the accuracy; object knowledge and semantic segmentation were used to create object-category specific disparity proposals [10], and [20] applied adaptive smoothness constraints using texture and edge information for a dense stereo estimation.

Residual Networks [16] (ResNets) are neural networks with skip connections. These networks, which are a specific case of Highway Networks [32], present state of the art results in the most competitive computer vision tasks. However, this is not true with stereo matching. The success of residual networks was attributed to the ability to train very deep networks when employing skip connections [12]. A complementary view is presented by [33], who attribute it to the power of ensembles and present an unraveled view of ResNets that depicts ResNets as an ensemble of networks that share weights.

Very recently, a concurrent tech report proposed a different residual network architecture for the task of image classification, which, like us, employs multilevel skip connection [18]. The two main differences from our architecture are: First, by introducing the learned λ coefficient as a constant highway gate, we allow the network to adjust the contribution of the added connections. Second, we add scaling layers to control the receptive field of the network. Another very recent report entangles the network with more residual connections to create densely connected residual networks [6]. We evaluated these architectures and found them (Sec. 6) to be inferior to the proposed solution, which is much simpler. We also found that adding our constant skip connections contributes significantly to the above architectures.

Estimating the confidence of stereo matches in order to interpolate correspondences is one of the most popular research topics in stereo vision [8, 27, 9, 26]. A very recent work [31] was the first to leverage a CNN for stereo confidence measure. They incorporated conventional confidence features to the input of the CNN and trained the network especially for this purpose. Our global disparity network is different in four major ways: (i) We apply a single network

to obtain both the confidence score and a much more accurate disparity map. (ii) Our confidence indication is trained with reflective loss that depends not only on the ground truth but also on predicted labels that change dynamically during training. (iii) While [31] uses the confidence to improve the performance of the Semi-Global Matching step [13], we incorporate it in the outlier detection step during the disparity image refinement process. (iv) Our solution is not bounded to stereo matching and our reflective loss is a novel and general technique for evaluating confidence.

3. Computing the matching cost

The first step in a modern stereo matching pipeline is based on computing a matching cost at each position for every disparity under consideration. Starting from two images, left and right, for every position \mathbf{p} in the left image and disparity d , we compute the matching cost between a patch centered around $\mathbf{p} = (x, y)$ in the left image and a patch centered around $\mathbf{pd} = (x - d, y)$ in the right. The cost is expected to be low for patches centered around the same 3D location and high otherwise. Similar to previous work, we employ a convolutional neural network, trained on pairs of small image patches where the true disparity is known.

The novel architecture of our network is presented in Fig. 1. It consists of a composition of the following components: A constant highway residual block denoted as inner- λ -residual, which consists of two convolutional layers with 3×3 filters, 1×1 padding in order to preserve spatial size, and ReLU non-linearities, followed by a constant highway skip-connection (see Sec. 3.2). Two such blocks, followed by another constant highway connection, are combined into a second level of residual blocks denoted as outer- λ -residual blocks. Between outer- λ -residual blocks another convolutional layer, denoted as scaling layer, with 3×3 filters and no padding is added, followed by ReLU non-linearity, in order to increase the receptive field of the network. Note that since both inner- and outer-residual blocks maintain the spatial size, the scaling layers are the only factor on the size of the receptive field. For example, a network with 5 scaling layers of 3×3 filters has a 11×11 receptive field, no matter how many inner and outer blocks are being used. In this work, we use a description network that is composed of 5 outer blocks, separated by scaling layers.

In order to compare two image patches, two identical (tied weights) description networks are employed and two descriptors vectors are extracted – u_l and u_r . During training, two pathways are then used to compare the patches and produce a matching cost. The first pathway, denoted as the decision sub-network, is a fully connected network, which concatenates the two representations into a single vector $[u_l^T, u_r^T]^T$ and is trained via the cross-entropy loss. The second pathway directly employs the hinge loss criterion to

¹The code is available at <https://github.com/amitshaked/resmatch>

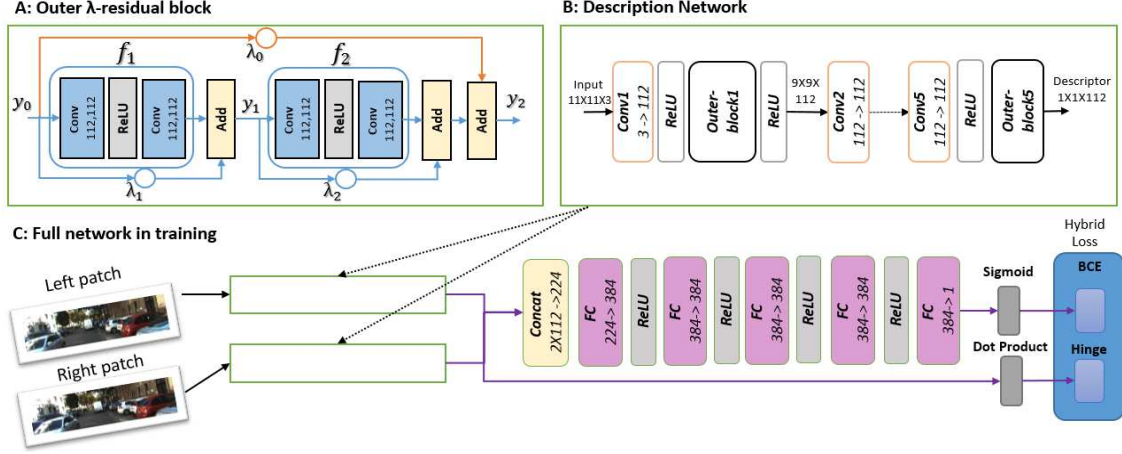


Figure 1: Our λ -ResMatch architecture of the matching cost network. (a) Outer λ -residual block that consists of two inner λ -residual blocks. (b) The core description network that consists of five outer λ -residual blocks, separated by scaling layers with ReLU activations. (c) Two-tower structure with tied parameters [11]. The description network outputs the two feature maps, which are the input for two pathways: the first concatenates and passes them to the fully-connected decision network which is trained via the cross-entropy loss, and the second directly employs a Hinge loss criterion to the dot product of the representations.

the dot product of the two representations $u_l^\top u_r$.

When computing the matching cost of two full size images, the description tensors of the two images, U^L and U^R , can be computed in a single forward pass in the description sub-network. Then, for every disparity d under consideration, the matching cost $C(\mathbf{p}, d)$ is computed by propagating $U^L(\mathbf{p})$ and $U^R(\mathbf{p}d)$ in the decision sub-network. This requires a total of *disparity_max* forward passes, which makes it the main factor on the methods runtime.

In order to have the flexibility to choose between accuracy and speed, one applies the full decision network, or uses only the dot-product based similarity score. In both cases, the added term, which is not used in run time, improves the performance of the matching network. This effect and the trade-off is further studied in Sec. 6.

In the following subsections, we elaborate on each of the structural novelties we introduce to matching networks.

3.1. Inner-Outer Residual blocks

Using a deeper network does not always mean a better prediction. For example, Zbontar and LeCun [36] report that five layer and six layer architectures for the description network are outperformed by the proposed four-layer architecture. Inspired by [16], we want to deepen the network by adding skip connections and employ residual blocks. However, residual networks are ineffective for matching.

In our experience, stacking residual blocks (with or without constant highway skip connections as described below), leads to great difficulties in making the network converge to a meaningful solution and does not improve the quality of

the prediction. We, therefore, suggest further deepening our network by introducing a second level of skip connections and adopting another connection every two inner residual blocks. In addition, spatial pooling layers followed by batch normalization are discarded. They are potentially harmful for matching networks since they reduce the resolution and sensitivity. They are, therefore, incompatible with the stereo matching task.

We believe that the added capacity of the proposed architecture improves the results both quantitatively and qualitatively since they allow us to employ new sources of information that are inaccessible in the conventional architectures. For example, our network benefits from the use of color, while the literature reports no added benefit from it (see Supplementary).

3.2. Constant highway skip connection

In order to further improve the effectiveness of the residual shortcuts, we introduce a constant highway skip connection, in which the identity shortcut of the residual building block is weighted by a learned factor λ , and formally defined as:

$$y_{i+1} = f_{i+1}(y_i) + \lambda_{i+1} \cdot y_i \quad (1)$$

In the highway network, the two terms $f_{i+1}(y_i)$ and y_i are weighted by t_{i+1} and $1 - t_{i+1}$ respectively, where t_{i+1} is a function of y_i . In our case, the weighting collapses to a learned parameter λ_{i+1} .

To further understand the effect of the multilevel constant highway connections, we unravel it. Consider an outer block that consists of two inner blocks as shown in Fig. 1(a).

The formulation of the output y_2 is recursive by its nature and can be unrolled as:

$$\begin{aligned} y_2 &= \lambda_0 y_0 + \lambda_2 \cdot y_1 + f_2(y_1) \\ &= \lambda_0 y_0 + \lambda_2 (\lambda_1 y_0 + f_1(y_0)) + f_2(\lambda_1 y_0 + f_1(y_0)) \\ &= (\lambda_0 + \lambda_2 \lambda_1) y_0 + \lambda_2 f_1(y_0) + f_2(\lambda_1 y_0 + f_1(y_0)) \end{aligned} \quad (2)$$

One can see that the added parameter λ_2 controls the flow of f_1 , λ_1 balances the input to f_2 and the outer parameter λ_0 controls y_0 . That way, when the residual network is interpenetrated as an ensemble of the possible paths [33], the learned parameters determine the contribution of each path to the ensemble. For example, we observed that they adopt much smaller values in the upper layers of the network to reduce the effect of the shortest paths and bias the network toward deeper representations (see Supplementary).

All λ -parameters are initialized with the value of one in order to emulate vanilla residual connections, and are then adjusted by back-propagation, as usual. No regularization is applied to this term.

3.3. Hybrid loss

While further processing the output of the two descriptor networks improves the ability to discriminate between matching and non-matching patches, it comes at the cost of making the descriptors less explicit. We, therefore, suggest combining two losses together: a hinge loss over the dot product $s = u_l^\top u_r$ and the cross-entropy over the decision network's output v . Similar to [36] we consider pairs of examples, matching and non-matching, centered around the same image position, and the compound loss is given by:

$$loss = \alpha \cdot XEnt(v_+, v_-) + (1 - \alpha) \cdot Hinge(s_+, s_-) \quad (3)$$

where $Hinge(s_+, s_-) = \max(0, m + s_- - s_+)$ and $XEnt(v_+, v_-) = -(\log(v_-) + \log(1 - v_+))$

Note that the dot product produces similarity score. Therefore, when choosing the fast pathway the output is multiplied by -1 to represent the matching cost. A margin of $m = 0.2$ for the hinge loss and $\alpha = 0.8$ are used throughout the experiments.

4. Computing the disparity image

The computation of the matching cost results in a map $C(\mathbf{p}, d)$ of size $H \times W \times \text{disparity_max}$ in which the matching cost of every position is computed for every possible disparity. The goal of the next stage is to output the disparity image $D(\mathbf{p})$ of size $H \times W$ with the predicted disparity in every position.

Modern stereo matching pipelines use few post-processing steps, and then apply the ‘‘winner takes all’’ strategy: $D(\mathbf{p}) = \text{argmin}_d C(\mathbf{p}, d)$. The post processing is

required, since even with improved matching networks, in order to be competitive, the method needs to incorporate information from the neighboring pixels that is beyond a simple maximization. Following Mei et al [23], we begin by applying cross-based cost aggregation [17] (CBCA) to combine information from neighboring pixels by averaging the cost with respect to depth discontinuities, continue with semi global matching [13] (SGM) to enforce smoothness constraints on the disparity image, and then apply few more iterations of the cost aggregation, as described in [36].

While CBCA and SGM contribute greatly to the success of modern stereo matching pipelines, they are limited and error especially in challenging situations where machine learning can help. These situations include occluded or distorted areas, highly reflective or sparse texture regions and illumination changes. Fig. 2(b) presents one example in which these schemes fail to correct the matching in reflective regions such as car glass.

To overcome this challenge, while one can follow the footsteps of Guney and Geiger[10] and use object-category specific disparity proposals, this requires explicit object knowledge and semantic segmentation, which we choose to avoid for two reasons: the introduced computational complexity, and the loss of generality associated with specific objects models. Instead, we propose to apply a learned criterion and replace the WTA approach. We construct a global disparity convolutional neural network and propagate the entire matching cost map to output the disparity at each position. An example of how this method can help in challenging situations is presented in Fig. 2(c). This network is trained with a novel reflective loss to simultaneously produce a confidence measure in the network's disparity prediction, to be used later in the refinement process, as described in Sec 5.

4.1. Global Disparity Network

The data set used to train the disparity network is composed of processed images from the matching cost network training data. For each pair of left and right images, we compute the full image size matching cost at each possible disparity as described in Sec. 3, and then apply CBCA and SGM. Note that the matching network returns independent probability estimations, and that post CBCA and SGM, the values can become negative and not bounded to a specific range. We, therefore, apply $Tanh$ in order to bring the values to the fixed range $[-1, 1]$. The target (ground truth) value for each matching costs patch is the disparity of its central pixel. Sampling 9×9 patches this way, we collect 25 (17) million training examples for training the KITTI 2012 (2015) disparity network.

The patches are fed to the global disparity network (GDN) as described in Fig. 3. Two layers are considered as target layers: FC3, that outputs the vector \mathbf{y} , which is the

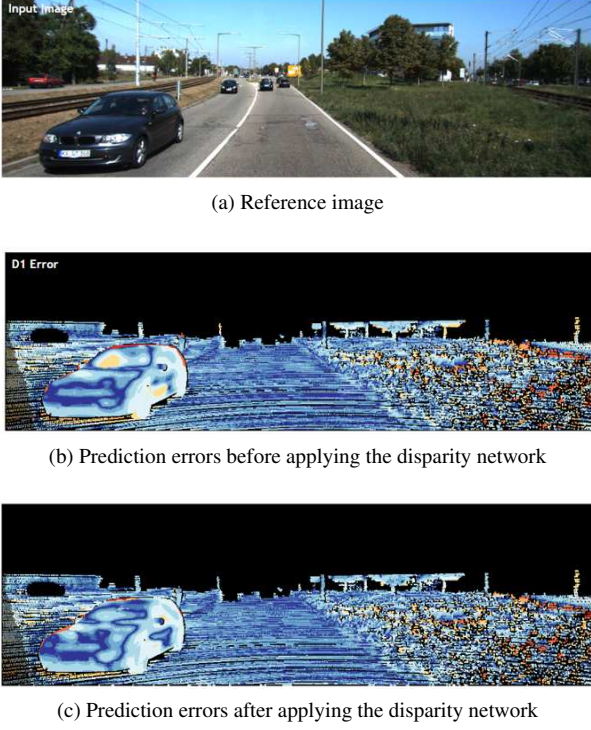


Figure 2: An example taken from KITTI 2015 data set showing the effect of the disparity network. Observe the errors in the predicted car glass disparities before and after applying the disparity network.

score y_i for every disparity d_i , and FC5, which depicts the confidence c in the prediction. The loss function on FC3, inspired by [21], is a weighted cross-entropy loss that centers the mass around the ground truth y^{GT} , with respect to the error metric of the data set:

$$loss(\mathbf{y}, y^{GT}) = - \sum_{y_i} p(y_i, y^{GT}) \cdot \log \frac{e^{-y_i}}{\sum_j e^{y_j}} \quad (4)$$

where $p(y_i, y^{GT})$ is a smooth target distribution, centered around the ground-truth y^{GT} . For the KITTI data set, we are interested in 3-pixel error metric and use:

$$p(y_i, y^{GT}) = \begin{cases} \lambda_1 & \text{if } |y_i - y^{GT}| \leq 1 \\ \lambda_2 & \text{if } 1 < |y_i - y^{GT}| \leq 2 \\ \lambda_3 & \text{if } 2 < |y_i - y^{GT}| \leq 3 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

The main difference from [21] is that we allow real-valued y^{GT} and the loss is modified accordingly. The values used in our work are $\lambda_1 = 0.65$, $\lambda_2 = 0.25$, $\lambda_3 = 0.1$. A model that outputs the disparity instead of the scores vector was also tested with different regression losses and found to be inferior to our model.

4.2. Reflective confidence

In order to obtain a confidence measure from the disparity network, we simultaneously train a binary classifier consists of two fully-connected layers via the binary cross-entropy loss. The training labels for this loss reflect the correctness of the score vector \mathbf{y} , which is also the input to the classifier, as shown in Fig 3. After each forward pass, $\text{argmax}_i y_i$ is compared to the ground truth disparity y^{GT} . If the prediction is correct, i.e differs from the ground truth by less than one pixel, the sample is considered positive, otherwise negative. Note that although the KITTI data set requires an error less than three pixels, we notice that training the confidence allowing a three pixel error (and not just one) causes too many positive samples and is not effective.

This loss is unconventional in the sense that the target value depends not only on the ground truth, but also on the activations of the network. To our knowledge, this is the first loss in the literature that is based on labels that change dynamically during training in this way.

This reflective loss is combined with the weighted cross-entropy loss of FC3 using 15:85 weights, respectively. We employ mini-batches of size 128 and a momentum of 0.9. The network was trained for 15 epochs, starting with a learning rate of 0.003 and decimating it on the 12th epoch.

5. Disparity refinement

While the global disparity network greatly contributes to the quality of the predicted disparity image, it can still suffer from known issues such as depth discontinuities and outlier pixel predictions. The goal of the third and last stage is to refine the disparity image and output the final prediction. Similar to [36, 23], we employ a three-step refinement process: (i) left-right consistency check for outlier pixel detection and interpolation, in which we incorporate our confidence score; (ii) sub-pixel enhancement in order to increase the image resolution, and (iii) median and bilateral filtering for smoothing the disparity image without blurring the edges. The second and third steps are performed exactly as in [36]. The first step is described below.

Denote $C^L(\mathbf{p})$ as the confidence score at position \mathbf{p} of the prediction $d = D^L(\mathbf{p})$ obtained by using the left image as a reference, and $C^R(\mathbf{pd})$ the confidence score at the correspondent position $\mathbf{p} - \mathbf{d}$ of the prediction $D^R(\mathbf{pd})$, obtained by using the right image as a reference. We label each position \mathbf{p} applying these rules in turn:

$$\begin{aligned} \text{correct} & \quad \text{if } |d - D^R(\mathbf{pd})| \leq \tau_1 \quad \text{or} \\ & \quad (C^L(\mathbf{p}) \geq \tau_2 \text{ and } C^L(\mathbf{p}) - C^R(\mathbf{pd}) \geq \tau_3) \\ \text{mismatch} & \quad \text{if there exist } \hat{d} \neq d \text{ s.t. } |\hat{d} - D^R(\mathbf{p}\hat{\mathbf{d}})| \leq \tau_4 \\ \text{occlusion} & \quad \text{otherwise} \end{aligned}$$

That means a pixel is labeled as *correct* if the two predic-

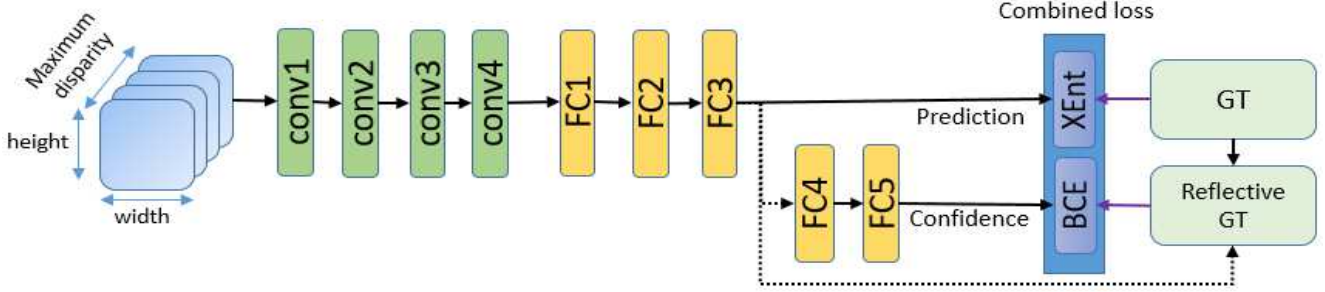


Figure 3: The global disparity network model for representing disparity patches. ReLU units are used as activation functions following every convolution and fully connected layer. Two layers are considered as target layers: FC3 on which LogSoftMax is applied to determine the predicted disparities, and FC5 which depicts the confidence measure. Cross-entropy loss for the prediction and binary cross-entropy loss for the confidence measure are combined together using 85:15 weights, respectively.

tions $D^L(\mathbf{p})$ and $D^R(\mathbf{p}\hat{\mathbf{d}})$ match, or they don't match but the reference prediction $D(\mathbf{p}) = D^L(\mathbf{p})$ is much more reliable. When neither holds, a pixel is considered *mismatch* if there exist another disparity \hat{d} such that if it were the prediction, it would have matched $D^R(\mathbf{p}\hat{\mathbf{d}})$. If none exist, the pixel is considered as *occlusion*. Throughout our experiments, we use $\tau_1 = 1$ the maximum left-right prediction disagreement, $\tau_2 = 0.7$ the minimum confidence score in the prediction, $\tau_3 = 0.1$ the minimum left-right confidence gap, and $\tau_4 = 1$ the maximum left-right prediction disagreement for other possible disparities.

For pixels labeled as *mismatch*, we want the disparity value to come from the reliable neighboring pixels and so take the median of the nearest neighbors labeled as *correct* from 16 different directions. The value of outliers \mathbf{p} marked as *occlusion* most likely come from the background. Therefore, the interpolation is done by moving left until the first *correct* pixel and use its value.

6. Experimental results

We evaluated our pipeline on the three largest and most competitive stereo data sets: KITTI 2012, KITTI 2015 and Middlebury. Comparisons with the state of the art and components analysis are provided.

6.1. Benchmark results

KITTI stereo data sets: The KITTI 2012 [7] data set contains 194 training and 195 testing images, and the KITTI 2015 [24] data set contains 200 training and 200 testing images. The error is measured as the percentage of pixels for which the true disparity and the predicted disparity differ by more than three pixels. The leader-boards of the two data sets are presented in Tab. 1 and Tab. 2. The reported error rates were obtained by submitting the generated disparity maps of the testing images to the online evaluation servers. Our accurate method is ranked first on both benchmarks and improved the error rate of the mc-cnn [36] base-

	Method	Set.	NOC	ALL	runtime
1	Ours		2.91	3.42	48s
2	Displets v2[10]	S	3.09	3.43	265s
3	PCBP[25]		3.17	3.61	68s
4	Ours-fast		3.29	3.78	2.8s
5	MC-CNN-acrt[36]		3.33	3.89	67s

Table 1: The highest ranking methods on KITTI 2015 due to November 2016, ordered by the error rate for all pixels. The S in the settings indicates the use of semantic segmentation. Very recently, two more anonymous submissions were submitted to the online evaluation server. CNNF+SGM achieves an error rate of 3.60 for all pixels and 3.04 for non-occluded pixels, and SN that achieves 3.66 and 3.09, respectively. We do not know whether or not they use segmentation.

line from 2.43 to 2.27 on KITTI 2012 and from 3.89 to 3.42 on KITTI 2015.

Our fast architecture was also submitted to the online servers, and a comparison between methods that run under five seconds is presented in Tab. 3 and Tab. 4.

The runtime was measured by testing our pipeline on the NVIDIA Titan X (pascal) graphics processor unit.

Middlebury stereo data set: The Middlebury stereo data set contains five separate works in the years 2001 [3], 2003 [30], 2005 [28], 2006 [14], and 2014 [4]. The image pairs are indoor scenes given in a full, half and quarter resolution, and rectified perfectly using precise 2D correspondences for perfect rectification, or imperfectly using standard calibration procedures. We trained our network on half resolution, due to the limited size of our GPU's memory card, on pairs rectified imperfectly, since only two out of fifteen image pairs in the test set are rectified perfectly. The error is measured for pixel disparity predictions that differ from the ground truth by more than two pixels, and is al-

Method	Set	NOC	ALL	runtime
1 Ours		2.27	3.40	48s
2 PCBP[25]		2.36	3.45	68s
3 Displets v2[10]	S	2.37	3.09	265s
4 MC-CNN-acrt[36]		2.43	3.63	67s
5 cfusion[25]	MV	2.46	2.69	70s

Table 2: The highest ranking methods on KITTI 2012 due to November 2016, ordered by the error rate for non occluded pixels. The S in the settings indicates the use of semantic segmentation and MV the use of more than two temporally adjacent images. The very recent anonymous submissions mentioned in Tab. 1 were also submitted here, where CNNF+SGM achieves 2.28 error rate for non occluded pixels and 3.48 for all pixels, and SN 2.29 and 3.50 respectively.

Rank	Method	NOC	ALL	runtime
1	Ours-fast	3.29	3.78	2.8s
2	DispNetC[22]	4.05	4.34	0.06s
3	Content-CNN[21]	4.00	4.54	1s
4	MC-CNN-fast[36]	?	4.62	0.8s
5	SGM+CNN(anon)	4.36	5.04	2s

Table 3: The highest ranking methods on KITTI 2015 for methods under 5 seconds due to November 2016.

Rank	Method	NOC	ALL	runtime
1	Ours-fast	2.63	3.68	2.8s
2	MC-CNN-fast[36]	2.82	?	0.7s
3	Content-CNN[21]	3.07	4.29	0.7s
4	Deep Embed[2]	3.10	4.24	3s
5	SPS-st[34]	3.39	4.41	2s

Table 4: The highest ranking methods on KITTI 2012 for methods under 5 seconds due to November 2016.

ways computed on full resolution. Hence, when training on half resolution, we are interested in less than one pixel error. The data set contains 60 image pairs for which the ground truth is available, but unlike KITTI, the maximal disparity is not fixed and varies between 30 and 800. Our global disparity network feature input plane is the size of the maximal disparity, and since there are only a few pairs for each possible value, there was not enough data to train the network at a fixed size. We have, therefore, tested our λ -ResMatch architecture using the post processing of [36]. As can be seen in Tab 5, the fast architecture introduces substantial improvement, lowering the validation error from 9.87 reported on [36] to 9.08. We were not able to reproduce the

Matching	Processing	KITTI 2012		KITTI 2015		MB
		fast	act	fast	act	fast
content [21]	-	6.61	-	7.23	-	-
content +HW [32]	-	6.70	-	7.01	-	-
content +RN [16]	-	6.96	-	7.05	-	-
content + λ -RM	-	5.60	-	6.94	-	-
mc-cnn [36]	SM (w.t.a) [36]	3.02	2.61	3.99	3.25	9.87
λ -RM	SM (w.t.a) [36]	2.73	2.45	3.69	3.15	9.08
$\bar{\lambda}$ -RM	SM+GDN	2.66	2.40	3.18	2.87	-
λ-RM	SM+GDN+CR	2.65	2.38	3.16	2.83	-

Table 5: Comparison between [21], [36] and λ -ResMatch pipelines, also when augmenting with our λ -ResMatch, or with alternative residual architectures: Highway [32] or ResNets [16]. The validation error is computed by splitting the KITTI and Middlebury stereo data sets into 80-20 train validation.

7.91 error rate reported for the accurate architecture. Training MC-CNN [36] with its published code obtained 8.18 validation error that we improved with λ -ResMatch to 8.01.

6.2. Components Analysis

In order to demonstrate the effectiveness of our novelties in each stage of the pipeline, we gradually tested them on the above data sets. Table 5 reports the contribution of adding each stage. One can see that on the KITTI 2015 data set, the greatest improvement comes from employing the global disparity network, while on KITTI 2012 it is the novel constant highway network. This is due to the fact that vehicles in motion are densely labeled in KITTI 2015 and car glass is included in the evaluation, and, therefore, reflective regions are more common.

λ -ResMatch architecture: We have tested our accurate architecture for the matching cost network thoroughly and compared it to five different architectures: (i) the baseline for our work MC-CNN [36], (ii) conventional highway network, (iii) ResNets [16], (iv) the concurrent work of Densely Connected Residual network [6], and (v) the concurrent work of Residual networks of residual networks [18] which also suggests to add another level of residual connection. In early experiments, we used the published code of these architectures as our matching cost network, but the results were far from competitive. We then removed the batch normalization and pooling layers and replaced them with our scaling layers, except the work of densely connected in which we used their original “transition layers”. These results are reported in Tab. 6, at the first row of every architecture. The second row contains further experiments of other architecture variants where we replaced the vanilla residual shortcuts with our constant highway skip connections. The results show that the multilevel constant connections contribute in almost all cases. To let the comparison be as direct as possible, we tested the accurate architecture

	Inner shortcut	Outer shortcut	KITTI 2012	KITTI 2015	MB
mc-cnn[36]	-	-	2.84	3.53	9.73
Highway[32]	-	-	2.81	3.51	9.77
ResNet[16]	A	-	2.82	3.71	10.03
λ variant	λ	-	2.81	3.55	10.01
DC[6]	A	-	3.86	5.02	11.13
λ variant	λ	-	3.42	4.43	11.07
RoR[18]	A	C	2.86	3.52	9.68
λ variant	λ	$\lambda \cdot C$	2.84	3.53	9.95
Variants of our method without the hybrid loss	A	A	2.78	3.49	9.63
	λ	A	2.75	3.42	9.83
	A	λ	2.78	3.46	10.3
	λ	λ	2.73	3.42	9.60
λ-ResMatch	λ	λ	2.71	3.35	9.53

Table 6: The validation errors of different architectures and their λ -variants, when trained on 20% of the data. “A” shortcut is the identity connection, “C” is 1X1-convolution and “ λ ” is our constant highway skip-connection.

with and without hybrid loss training. One can see that the λ -ResMatch architecture achieves the best accuracy in the task of stereo matching on all data sets, and that the added hybrid loss further improves the results.

Reflective confidence: To evaluate the performance of our new method for confidence indication, we compared it to the six most widely used techniques, using the AUC measure. These techniques belong to different categories according to which aspects of stereo cost estimation they take into account [15]. The notations we use to describe the different methods are: $d_1(\mathbf{p}) = D(\mathbf{p}) = D^L(\mathbf{p})$ the predicted disparity at position \mathbf{p} when using the left image as a reference, $c_1(\mathbf{p}) = C_{SGM}(\mathbf{p}, d_1)$ the matching cost of the prediction disparity (before applying the global disparity network), c_2 the second smallest local minimum, and $prob(\mathbf{p}) = C_{GDN}(\mathbf{p}, d_1) = \max_d C_{GDN}(\mathbf{p}, d)$ the probability of the predicted disparity.

The evaluated methods are (i) Matching Score Measure (MSM)[5] which assigns higher confidence to lower costs: $C_{MSM} = -c_1$. (ii) The probability (PROB) of the prediction after applying the disparity network $C_{prob} = prob$, (iii) The Curvature (CUR) of the matching cost $C_{CUR} = -2 \cdot c(d_1) + c(d_1 - 1) + c(d_1 + 1)$ that is widely used in the literature, (iv) the Peak Ratio (PKRN)[15] is the ratio between the second smallest cost and the smallest $C_{PKRN} = \frac{c_2}{c_1}$, (v) the Negative Entropy Measure (NEM)[29] $p(d) = \frac{e^{-c_1}}{\sum_d e^{-c(d)}}$, $C_{NEM} = -\sum_d p(d) \log p(d)$ that takes into consideration the entire cost curvature, and (vi) the Left Right Difference (LRD)[15] which utilizes both left-right consistency check and the margin between the two smallest minima of the cost: $C_{LRD} = \frac{c_2 - c_1}{|c_1 - \min c_R(x - d_1, y, d_R)|}$.

We tested the above measures on 40 random validation

	Ref	MSM	Prob	CUR	PKRN	NEM	LRD
KITTI2012	0.943	0.928	0.648	0.772	0.930	0.919	0.833
KITTI2015	0.894	0.850	0.758	0.832	0.853	0.864	0.812

Table 7: The average AUC over the entire validation set for different confidence measures.

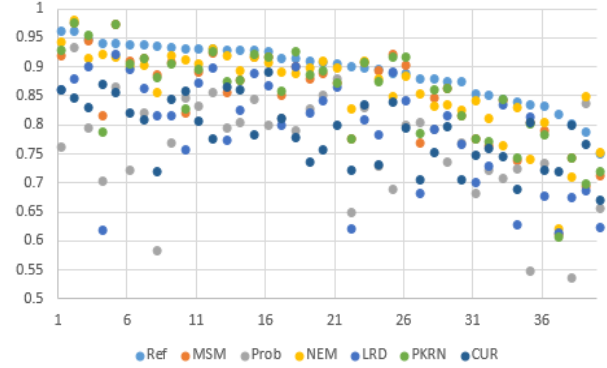


Figure 4: AUC of confidence measures on 40 random validation images from the KITTI 2015 stereo data set.

images from the KITTI 2012 and the KITTI 2015 data sets. The results for KITTI 2015 presented in Fig 4, and the very similar results for KITTI 2012 that can be found in the supplementary, show that our reflective confidence measure performs better on almost every image. The average score over the entire data set in Tab. 7 shows that it is also the overall most accurate in both data sets.

7. Discussion

It is interesting to note that unlike the most recent state of the art results, we make no use of semantic segmentation. Semantic segmentation employs additional training data that is not used by our method and requires an additional runtime. Nevertheless, it would be interesting to see whether the benefits of semantic segmentation are complementary to the increased performance we obtain, in which case, our results could be further improved with little effort.

The need for specific residual architectures for matching probably goes hand in hand with the favorable performance of moderate depth networks, the detrimental effect of batch normalization, and other unique practices in this domain. More study is required in order to understand what sets this problem apart from categorization problems.

We believe that the reflective loss can be extended to other problems and applications such as the gradual learning schemes self-paced [19] and curriculum [1] learning.

Acknowledgments

This research is supported by the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI).

References

- [1] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.
- [2] Z. Chen, X. Sun, Y. Yu, L. Wang, and C. Huang. A deep visual correspondence embedding model for stereo matching costs. *ICCV*, 2015.
- [3] R. S. Daniel Scharstein. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, pages 7–42, Vol 47, 2002.
- [4] Y. K. G. K. N. N. X. W. Daniel Scharstein, Heiko Hirschmuller and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition (GCPR)*, 2014.
- [5] M. M. G. Egnal and R. Wildes. A stereo confidence metric using single view imagery with comparison to five alternative approaches. In *Image and Vision Computing*, 2004.
- [6] K. Q. W. Gao Huang, Zhuang Liu. Densely connected convolutional networks. In *arXiv:1608.06993*, 2016.
- [7] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [8] R. Gherardi. Confidence-based cost modulation for stereo matching. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2008.
- [9] L. v. d. M. Gorkem Saygili and E. A. Hendriks. Stereo similarity metric fusion using stereo confidence confidence measures. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2014.
- [10] F. Guney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [11] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [12] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. *arXiv preprint arXiv:1603.05027*, 2016.
- [13] D. S. Heiko Hirschmuller. Evaluation of cost functions for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [14] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [15] X. Hu and P. Mordohai. A quantitative evaluation of confidence measures for stereo vision. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.
- [16] S. R. J. S. Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.
- [17] J. L. Ke Zhang and G. Lafruit. Cross-based local stereo matching using orthogonal integral images. In *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1073–1079, 2009.
- [18] T. X. H. X. Y. L. G. T. L. Ke Zhang, Miao Sun. Residual networks of residual networks: Multilevel residual networks. In *arXiv:1608.02908*, 2016.
- [19] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010.
- [20] C.-S. K. Kyung-Rae Kim. Adaptive smoothness constraints for efficient stereo matching using texture and edge information. 2016.
- [21] W. Luo, A. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [22] N. Mayer, E. Ilg, P. Husser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, June 2016.
- [23] S. X.-Z. M. W. H. Z. X. e. a. Mei, X. On building an accurate stereo matching system on graphics hardware. In *Computer Vision Workshops (ICCV Workshops)*, 2011.
- [24] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [25] V. Ntouskos and F. Pirri. Confidence driven tgv fusion. *arXiv preprint arXiv:1603.09302*, 2016.
- [26] M.-G. Park and K.-J. Yoon. Leveraging stereo matching with learning-based confidence measures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 101–109, 2015.
- [27] S. G. D. Pfeiffer and N. Schneider. Exploiting the power of stereo confidences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 297–304, 2013.
- [28] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [29] D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. In *Stereo Matching with Nonlinear Diffusion*, 1998.
- [30] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.
- [31] A. Seki and M. Pollefeys. Patch based confidence prediction for dense disparity map. In *British Machine Vision Conference (BMVC)*, 2016.
- [32] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.
- [33] A. Veit, M. Wilber, and S. Belongie. Residual networks behave like ensembles of relatively shallow networks. In *NIPS*, 2016.
- [34] K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *ECCV*, 2014.

- [35] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1592–1599, June 2015.
- [36] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *CoRR*, abs/1510.05970, 2015.