Supplementary Material for Uncertainty-Aware CNNs for Depth Completion: Uncertainty from Beginning to End

Anonymous CVPR submission

Paper ID 7778

1. Implementation Details

In this section, we give more details on the implementation of our proposed method such as the loss function and the design of the confidence estimation and the noise variance estimation networks.

1.1. The Loss Function

We drove a loss function to train the proposed probabilistic normalized convolutional neural networks *(pNCNN)*, which reads:

$$C(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} \underbrace{\frac{\parallel y_i - \hat{r}_i^L \parallel^2}{s_i}}_{\text{Data term}} + \underbrace{\log(s_i)}_{\text{Regl. term}}, \tag{1}$$

where s_i is the proposed uncertainty measure and it is equal to $\sigma_i^2/\langle \mathbf{a}|\mathbf{c}\rangle$. For convenience and numerical stability, we modify the regularization term so that s_i becomes consistent with the data term. This leads to:

$$C(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} \frac{\parallel y_i - \hat{r}_i^L \parallel^2}{s_i} - \log(\frac{1}{s_i}) \ , \tag{2}$$

This can be expanded using the definition of s_i:

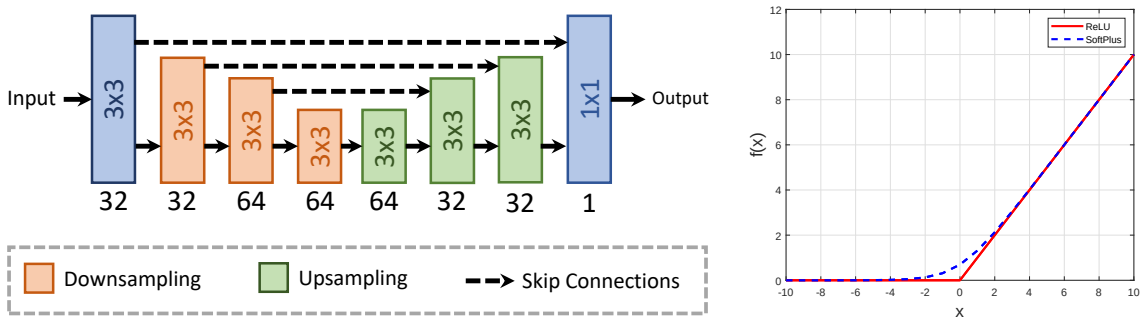
$$C(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} \frac{\langle \mathbf{a}^L|\mathbf{c}^L\rangle}{\sigma_i^2} \parallel y_i - \hat{r}_i^L \parallel^2 - \log(\frac{\langle \mathbf{a}^L|\mathbf{c}^L\rangle}{\sigma_i^2}) \tag{3}$$

where $\mathbf{a}^L, \mathbf{c}^L$ are the learned applicability and the output confidence of the last normalized convolution layer L respectively. This expansion makes it clear that our proposed uncertainty measure depends both on the output confidence from the normalized convolution layer and observations noise variance. A higher noise variance will reduce the output confidence from the NCNN and vice versa. This indicates that our proposed uncertainty measure encodes the single observation noise as well as the confidence with respect to the neighboring pixels.

1.2. The Architecture

We propose to learn the input confidence using a compact UNet [6] that is trained end-to-end with a normalized convolutional neural network (NCNN) [3]. We also learn observations noise variance using a similar UNet. The design of this UNet is shown in Figure 1a and it is identical for both networks. It is worth mentioning that this network has only 3 scales compared to original UNet which has 4 scale, since we found empirically that the 4th scale does not improve the estimation. The number of channels per convolution layer was significantly reduced for computational efficiency.

The choice of the activation for the last layer is crucial since it must produce valid range of values for confidences $[0, \infty[$. We choose the SoftPlus function (Shown in Figure 1b) due to its similarity to the ReLU activation. However, it does not suffer from the gradient discontinuity at zeros.



(a) (b)

Figure 1: (a) The proposed compact UNet used for the confidence estimation network and the noise variance estimation network. (b) The SoftPlus activation used at the final layer in comparison with the ReLU activation.

2. Ensemble methods

In the main paper, we evaluate different fusion schemes for an ensemble of our network *pNCNN*. We showed that all fusion schemes utilizing our proposed uncertainty measure outperform the commonly used fusion using the standard mean. Here, we give the definition for the evaluated fusion schemes.

2.1. The Standard Mean

The Mean fusion method refers to the average over the predictions y_i^k at pixel i:

$$\hat{y}_i = \frac{1}{N} \sum_{k=1}^{N} y_i^k \quad . \tag{4}$$

2.2. The Weighted Mean

Since the mean fusion does not take into account the uncertainties, we weight the predictions using their confidences c_i^k:

$$\hat{y}_i = \frac{1}{\sum_{k=1}^{N} c_i^k} \sum_{k=1}^{N} c_i^k y_i^k \quad . \tag{5}$$

2.3. Max Voting

Another commonly used voting scheme is to select the most confident prediction $k_i = \arg_m \max c_i^m$

$$\hat{y}_i = y_i^{k_i} \quad . \tag{6}$$

2.4. Maximum Likelihood Estimate

We can interpret our predictors as components of a Gaussian Mixture Model. If the prediction corresponds to the mean and the confidence corresponds to the unnormalized mixture weights, we can write the likelihood of a prediction $\hat{x}$ given predictions y^k from the networks as:

$$l(\hat{x}_i) = \frac{1}{\sum_{k=1}^{N} c_i^k} \sum_{k=1}^{N} \frac{c_i^k}{\sqrt{2\pi v^2}} \exp\left(\frac{\|\hat{x}_i - y_i^k\|^2}{2v^2} \right) \quad . \tag{7}$$

We can formulate an inference procedure based on the MLE for each pixel i as:

$$\hat{y}_i = \arg\max_{\hat{x}_i} \sum_{k=1}^{N} \frac{c_i^k}{v_i} \exp\left(\frac{\| \hat{x}_i - y_i^k \|^2}{2v_i^2} \right) \quad , \tag{8}$$

Optimization Procedure The likelihood function of a Gaussian Mixture Model is in general non-convex. However, for the 1D case, the number of modes is constrained to at most the number of components in the mixture [1]. Since it is guaranteed that the global maxima will be found if all local maximas are explored, we optimize the objective starting from each of the predictions. We use the ADAM optimizer with a maximum amount of steps set to 500. And we select the maximum of the local maximas which were found. Note that since we do not explicitly estimate the variances of the components we set v^2 to 0.1 for our experiments.

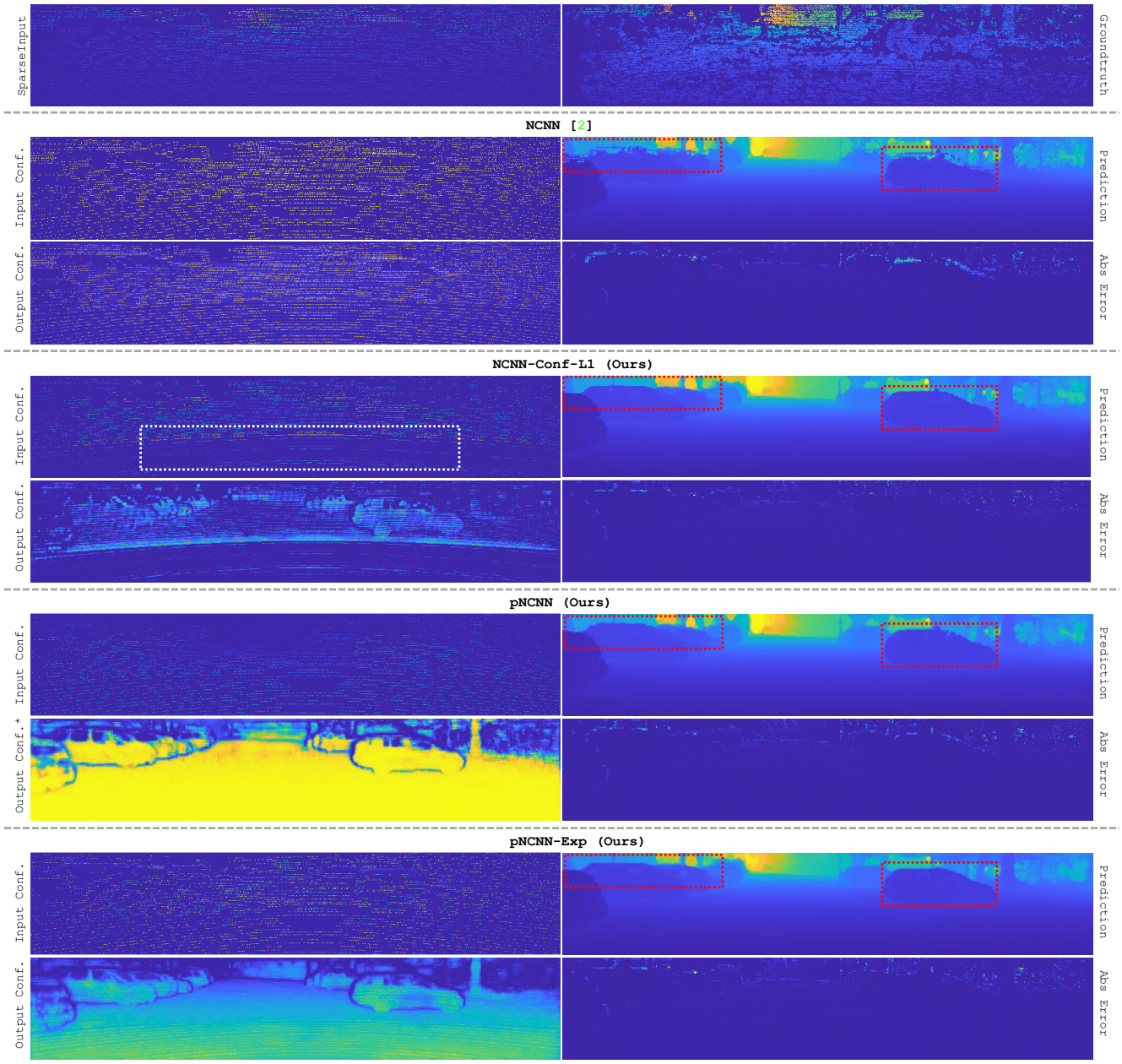
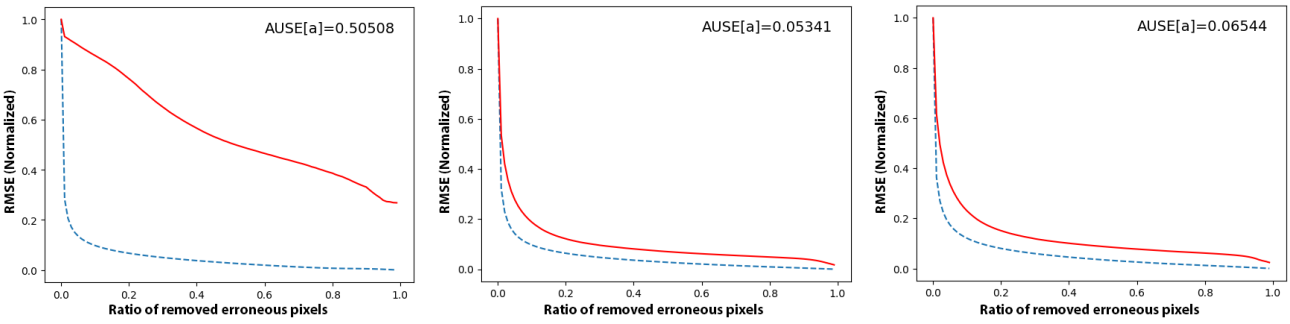


Figure 2: A qualitative example from the selected validation set of the KITTI-Depth dataset [7]. * denotes logarithmically scaled.



(a) (b) (c)

Figure 3: Sparsification plots for (a) *NCNN-Conf*, (b) *pNCNN*, and (c) *pNCNN-Exp*. The blue curve is the oracle and AUSE is the area between the two curves.

3. Additional Results

In this section, we show additional results for all the experiments in the paper. First, we show some qualitative examples on the KITTI-Depth dataset [7]. Then, we show the sparsificiation plots for our proposed uncertainty measure that were used to calculate the AUSE metric. Afterwards, we show some qualitative examples for multi-path interference correction and sparse optical flow rectification. Finally, we show illustrations on the NYU dataset [5] for the case of undisturbed input data.

3.1. Qualitative Results for The KITTI-Depth dataset

Figure 2 and 4 show qualitative examples for *NCNN* [2], our proposed *NCNN-Conf-L1*, *pNCNN*, and *pNCNN-Exp* from the selected validation set of the KITTI-Depth [7] dataset. *NCNN* assigns binary confidence to the input, which results in artifacts at regions with disturbed measurements especially edges (indicated with red squares). Our proposed *NCNN-Conf-L1* on the other hand, learns a proper input confidence which discards input measurements that causes the prediction error to increase. This causes the final prediction to be artifact-free and sharp along edges. It is worth mentioning that our input confidence estimation learned to discard some of the true measurements (indicated with the white squares) as well in order to produce smoother surfaces. Those discarded measurements are compensated for using other measurements on the end points of the same surface.

It is clear the output confidence from *NCNN-Conf-L1* is a densified version of the estimated input confidence. But it does not provide full uncertainty information for all observations in the prediction. Our proposed *pNCNN* addresses this problem and produces a reliable uncertainty measure for all observations. However, the prediction error at some disturbed measurements increase where the presumed Gaussian error model does not hold (indicated with the red squares if Figure 2). By applying the exponential function to s_i in the data term of the loss in *pNCNN-Exp*, the network focuses more on minimizing the prediction error for those disturbed measurements and produces a better prediction. Note that the range for the certainty measure changes with *pNCNN-Exp* due to the exponential scaling.

3.2. The Quality of the Proposed Uncertainty Measure

To examine the quality of our proposed uncertainty measure, we look at the commonly used sparsification plots [4]. Sparsification plots show how efficiently the uncertainty measure discards the erroneous measurements. The baseline in this case is the prediction error itself, which is denoted as *the oracle*. Sparsification plots for *NCNN-Conf-L1*, *pNCNN*, and *pNCNN-Exp* are shown in Figure 3. The uncertainty measure from *NCNN-Conf-L1* is not correlated with the oracle as the classical normalized convolution framework does not constitute any probabilistic properties. Our proposed probabilistic normalized convolution *pNCNN* on the other hand, produces an accurate uncertainty measure that is very similar to the error oracle. The modified version *pNCNN-Exp* also produces a high-quality uncertainty measure, but with a better handling of outliers.

3.3. Multi-Path Interference Correction

Figure 5 shows two qualitative results for the FLAT dataset. The first row, shows a scene with small areas of missing data. These areas are well handled by the *pNCNN* and the confidences clearly shows the uncertainty that exist in these areas and

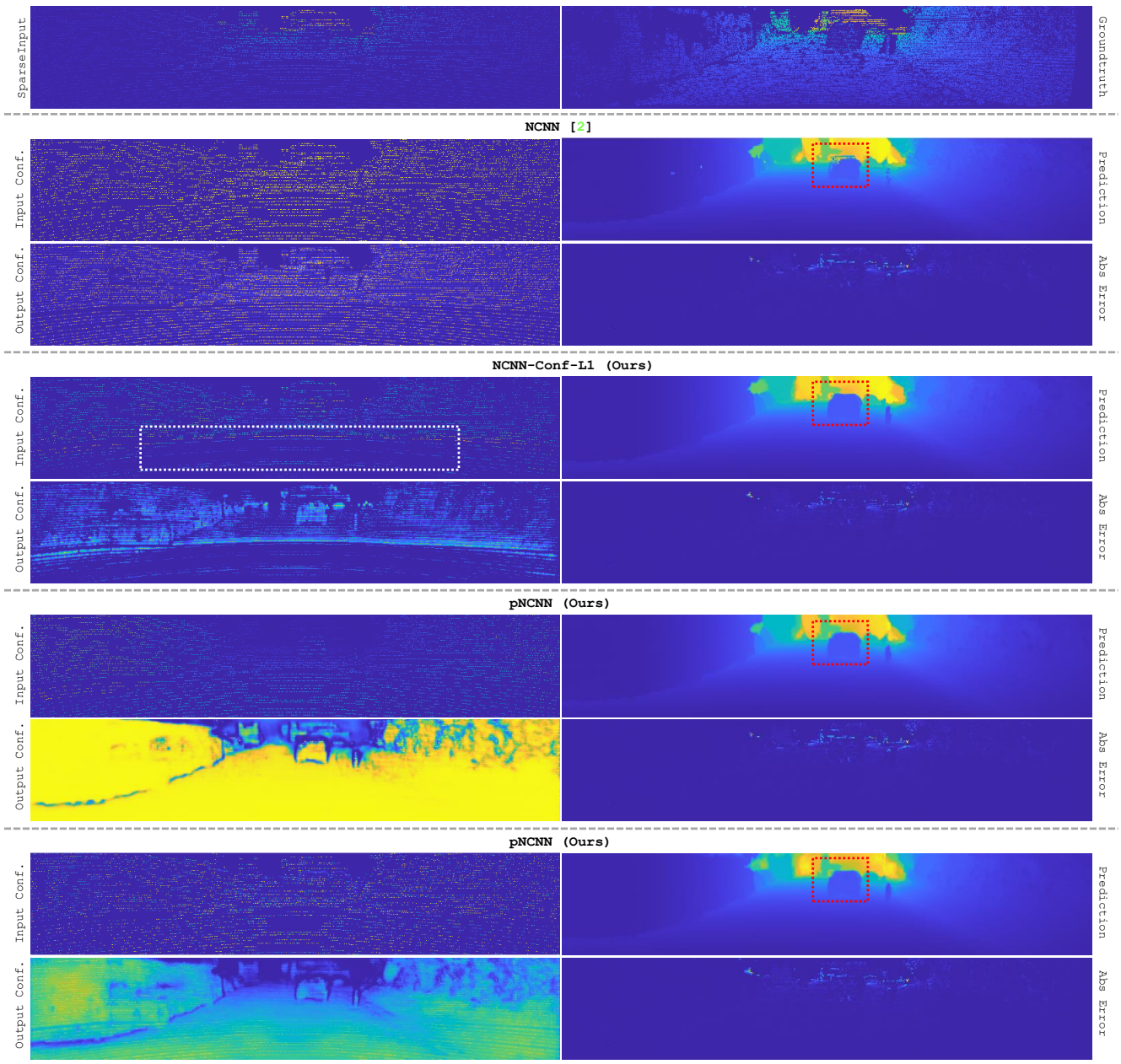


Figure 4: A second qualitative example from the selected validation set of the KITTI-Depth dataset [7]

on edges. The scene in the second row illustrates the effect of larger areas of missing data. These areas are missing too much data for the network to handle. As such, the output confidences is used to mask these parts of the signal. This illustrates the strength of our formulation in handling both smaller areas were the missing data can be extrapolated and larger areas where high uncertainty is assigned.

3.4. What happens when the input is undisturbed?

Figure 6 shows some qualitative examples on the NYU dataset [5] for our *NCNN-Conf-L1* compared to the standard NCNN [2]. In these examples, the sparse input is *undisturbed* and NCNN should perform well using the binary input confidences. However, NCNN struggles along edges due to equally trusting the background and the foreground. Our *NCNN-Conf-L1* on

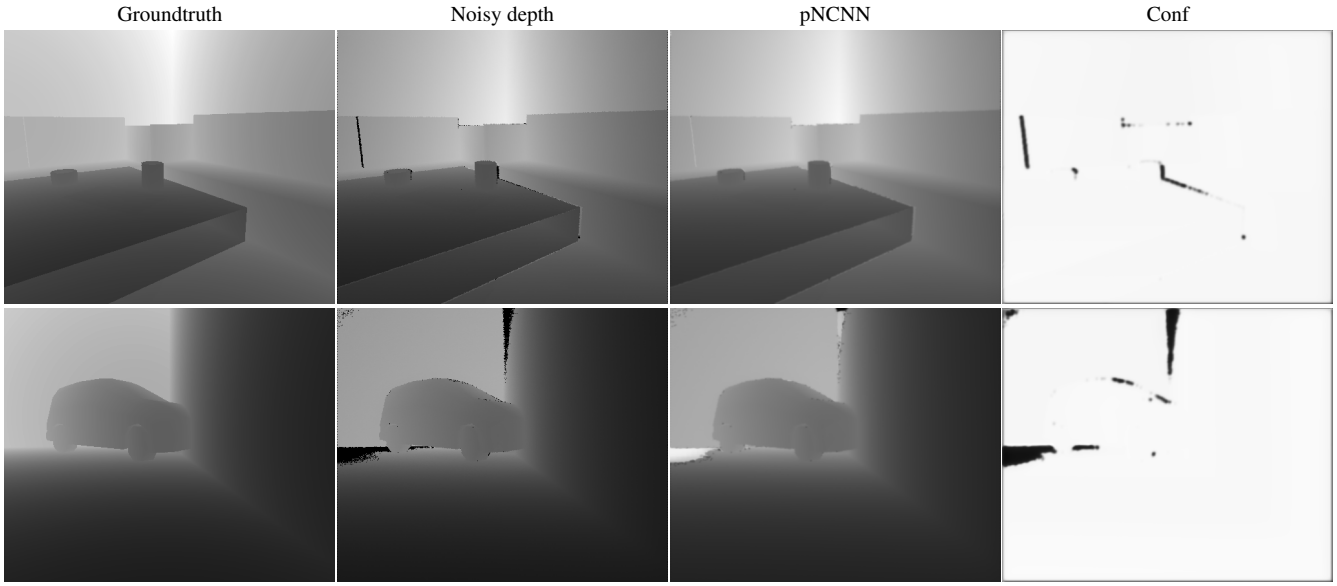


Figure 5: A qualitative example from the FLAT dataset showing the predicted output and the confidence of the proposed approach.

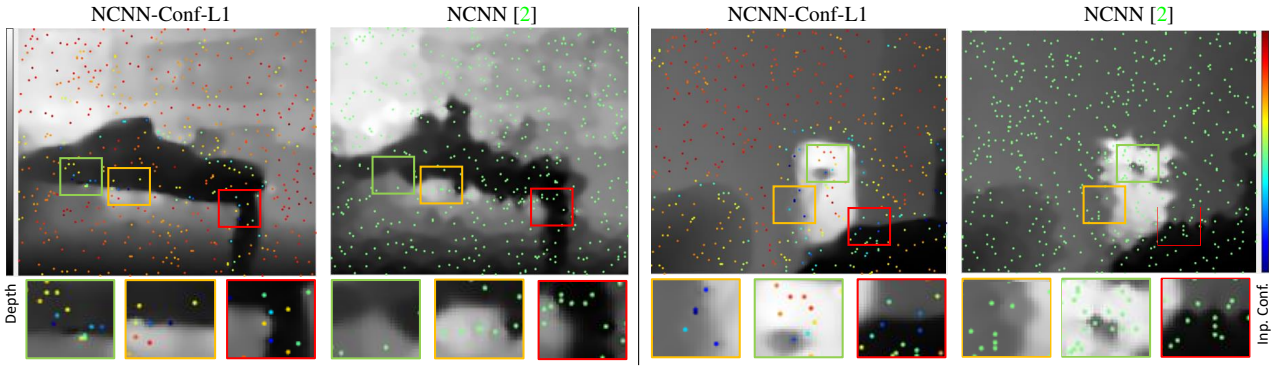


Figure 6: An examples from the NYU [5] dataset, where our confidence estimator (left) down-weights depth samples close to edges in order to obtain sharper output. On the other hand, the NCNN [2] struggles along edges due to equally trusting all input samples.

the other hand, learns proper input confidences that preserve edges similar to non-linear filtering.

3.5. Sparse Optical Flow Rectification

We include more results for the sparse optical flow rectification to demonstrate the generalization capabilities of our approach to other types of data. Qualitative examples are shown in Figure 7 and 8. Our method successfully removes noisy flow vectors despite the fact that they look completely random. This demonstrates the generalization capabilities of our approach in identifying the inherent noise in the data in a self-supervised manner.

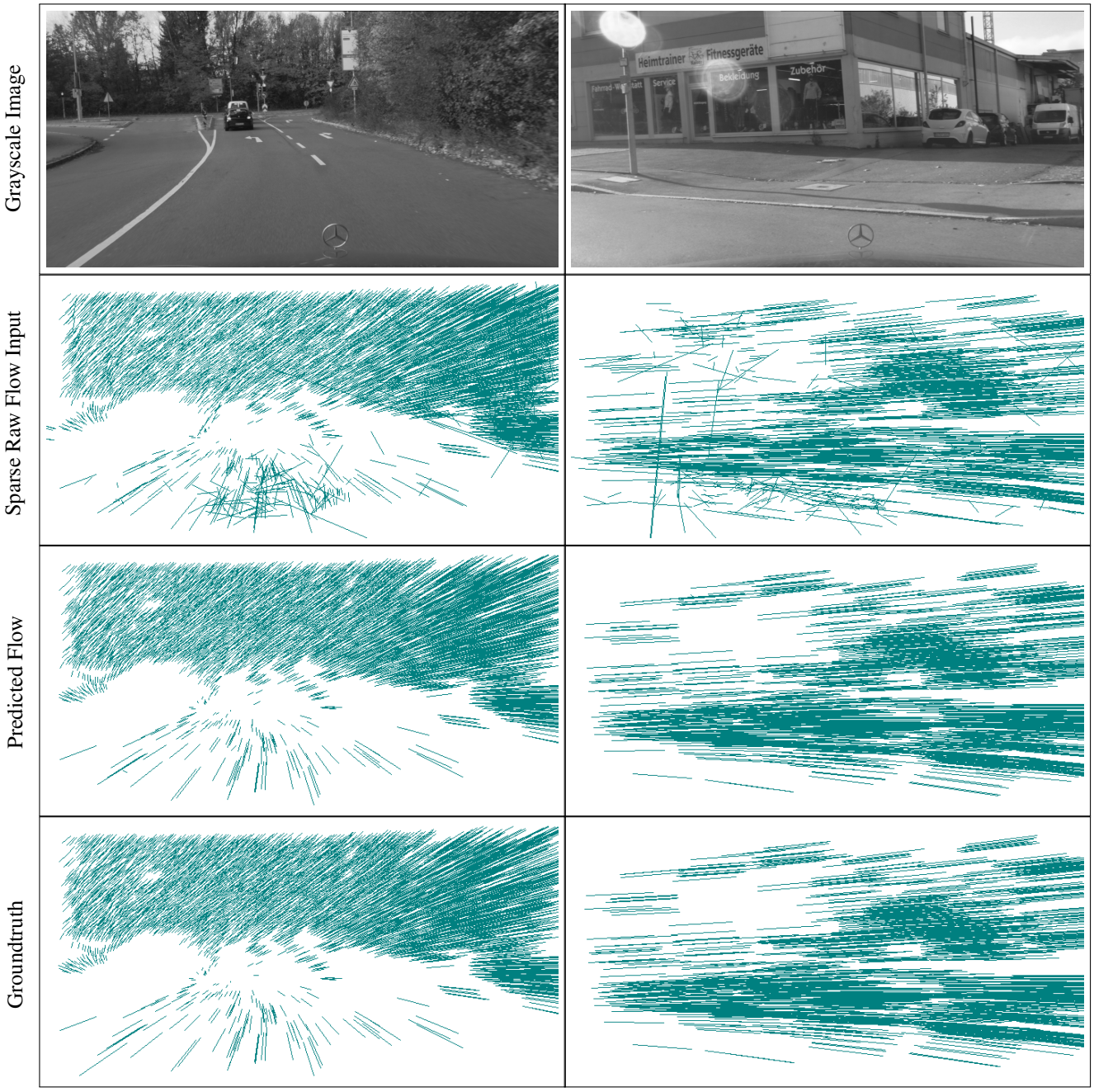


Figure 7: Two validation samples which highlight the networks noise reduction ability. To the left: tracking failures on the nearly homogeneous road. To the right tracking failures caused by glare. Note that the grayscale image is for visualization and not used.

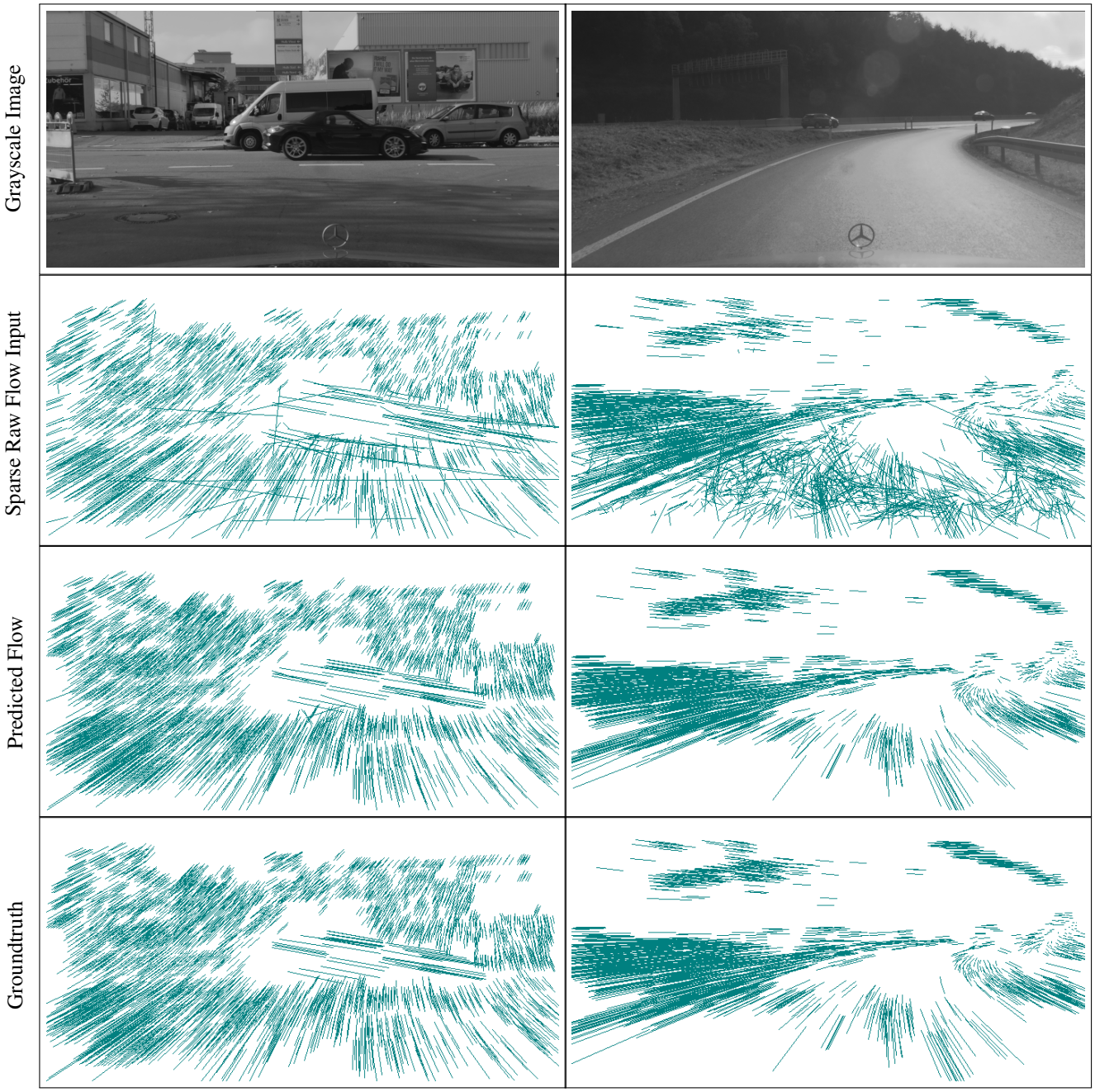


Figure 8: Left: validation sample with moving rigid objects, demonstrating that the system is not limited to a single epipolar geometry. Right: tracking failure cased by road reflection that is also rectified by our method. Note that the grayscale image is for visualization and not used.

References

[1] Miguel Á Carreira-Perpiñán and Christopher KI Williams. On the number of modes of a gaussian mixture. In *International Conference on Scale-Space Theories in Computer Vision*, pages 625–640. Springer, 2003. 3

[2] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Propagating confidences through cnns for sparse data regression. In *The British Machine Vision Conference (BMVC), Northumbria University, Newcastle upon Tyne, England, UK, 3-6 September, 2018*, 2018. 4, 5, 6

[3] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 1

[4] Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 652–667, 2018. 4

[5] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 4, 5, 6

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[7] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017. 3, 4, 5