International Journal of Computer Vision (2021) 129:2565–2582
https://doi.org/10.1007/s11263-021-01485-5

Learned Collaborative Stereo Refinement

Patrick Knöbelreiter[1] · Thomas Pock[1]

Received: 7 January 2020 / Accepted: 28 May 2021 / Published online: 20 June 2021

Abstract
In this work, we propose a learning-based method to denoise and refine disparity maps. The proposed variational network arises naturally from unrolling the iterates of a proximal gradient method applied to a variational energy defined in a joint disparity, color, and confidence image space. Our method allows to learn a robust collaborative regularizer leveraging the joint statistics of the color image, the confidence map and the disparity map. Due to the variational structure of our method, the individual steps can be easily visualized, thus enabling interpretability of the method. We can therefore provide interesting insights into how our method refines and denoises disparity maps. To this end, we can visualize and interpret the learned filters and activation functions and prove the increased reliability of the predicted pixel-wise confidence maps. Furthermore, the optimization based structure of our refinement module allows us to compute *eigen disparity maps*, which reveal structural properties of our refinement module. The efficiency of our method is demonstrated on the publicly available stereo benchmarks Middlebury 2014 and Kitti 2015.

Keywords Stereo · Refinement · Deep learning · Optimization · Interpretable AI

1 Introduction

Computing 3D information from a stereo image pair is one of the most important problems in computer vision. One reason for this is that depth information is a very strong cue to understanding visual scenes, and depth information is therefore an integral part of many vision based systems. For example, in autonomous driving, it is not sufficient to identify the objects visible in the scene semantically, but the distance to the objects is also very important. A lidar scanner can be used for distance estimates, but is often too expensive and provides only sparse depth estimates. Therefore, the primary approach is to compute depth information only from stereo images. However, due to reflections, occlusions, difficult illuminations etc.,, the calculation of depth information from images is still a very challenging task. To tackle these difficulties the computation of dense depth maps is usually split up into the four steps (i) matching cost computation, (ii) cost aggregation, (iii) disparity computation and (iv) disparity refinement (Scharstein and Szeliski 2002). In deep learning based approaches (i) and (ii) are usually implemented in a matching convolutional neural network (CNN), (iii) is done using graphical models or 3D regularization CNNs and (iv) is done with a refinement module (Tulyakov et al. 2018).

There are many approaches to tackle (i)-(iii). However, there are only a few learning-based works for disparity refinement (iv) (see Sect. 2). Existing work to refine the disparity maps is often based on black-box CNNs to learn a residual from an initial disparity map to a refined disparity map. In this work we want to overcome these black-box refinement networks with a simple, effective and most important easily interpretable refinement approach for disparity maps. We tackle the refinement problem with a learnable hierarchical variational network. This allows us to exploit both the power of deep learning and the interpretability of variational methods. In order to show the effectiveness of the proposed refinement module, we conduct experiments on directly refining/denoising winner-takes-all (WTA) solutions of feature matching and as a pure post-processing module on top of an existing stereo method.

Communicated by Simone Frintrop.

This work was partly supported from the ERC starting Grant HOMOVIS (No. 640156).

✉ Patrick Knöbelreiter
knoebelreiter@icg.tugraz.at

Thomas Pock
pock@icg.tugraz.at

[1] Graz University of Technology, Graz, Austria

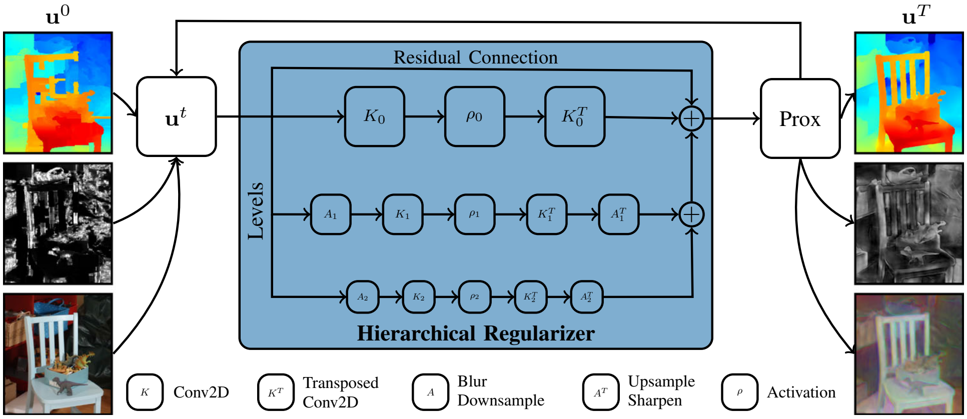


Fig. 1 Model overview. Our model takes three inputs, an initial disparity map, confidence map and the color image. The collaborative hierarchical regularizer iteratively computes a refined disparity map and yields refined confidences and a color image providing cues for depth discontinuities. The subscripts indicate the hierarchical level of the image pyramid

Figure 1 shows an overview of our method. The inputs to our method are an initial disparity map, a pixel-wise confidence map and the corresponding RGB color image. These three inputs span a collaborative space in which our hierarchical regularizer iteratively refines the initial inputs. Finally, the output of the hierarchical regularizer is the refined disparity map, a refined confidence map and a refined color image. Note that the refined (= output) color and confidence image are a byproduct of the refinement process.

Contributions We propose a learnable variational refinement network which takes advantage of the joint information of the color image, the disparity map and a confidence map to compute a refined disparity map. Our proposed method can be derived from the iterates of a proximal gradient method specifically designed for stereo refinement. Additionally, we evaluate a broad range of possible architectural choices in an ablation study. We demonstrate the interpretability of our model by visualizing the intermediate iterates and showing the learned filters as well as the learned activation functions. We show the effectiveness of our method by participating on the two complementary publicly available benchmarks Middlebury 2014 and Kitti 2015.

This paper extends the conference paper (Knöbelreiter and Pock 2019), where we additionally study (i) a model with shared parameters over the iterations, (ii) a comparison with the recent lightweight StereoNet refinement module (Khamis et al. 2018) and (iii) a new section, where we analyze the VN. To this end, we show how to compute eigen disparity maps that reveal structural properties of the learned regularizer and analyze the refined confidences in order to show the increased reliability of the confidences predicted by our model.

2 Related Work

We propose a learnable model using the modeling power of variational calculus to explicitly guide the refinement process for stereo. This combination of learning and classical optimization for stereo refinement allows us to group the related work into the three categories (i) variational methods, (ii) disparity refinement and (iii) learnable optimization schemes. We review the most related works of these categories in the following paragraphs.

Variational Methods Variational methods formulate the dense correspondence problem as minimization of an energy functional comprising a data fidelity term and a smoothness term. We use here the term *correspondence problem* to indicate that the following methods can in general be used for both optical flow and stereo, because stereo can be considered as optical flow in horizontal direction only. The data-term usually measures the raw intensity difference (Brox et al. 2004; Zach et al. 2007; Chambolle and Pock 2011) between the reference view and the warped other view. The regularizer imposes prior knowledge on the resulting disparity map. This is, the disparity map is assumed to be piecewise smooth. Prominent regularizers are the robust Total Variation (TV) (Zach et al. 2007) and the higher order generalization of TV as e.g. used by Ranftl et al. (2012, 2014) or by Kuschk and Cremers (2013). Variational approaches have two important advantages in the context of stereo. They naturally produce sub-pixel accurate disparities and they are easily interpretable. In order to capture large displacements as well, a coarse-to-fine warping scheme (Brox et al. 2004) is necessary. To overcome the warping scheme without losing fine details, variational methods can also be used to refine an initial disparity map. This has e.g. be done by Shekhovtsov et al. (2016) who refined the initial disparity estimates coming from a Conditional Random Field (CRF). Similarly, Revaud et al. (2015) and Maurer et al. (2017) used a variational method for refining optical flow.

Disparity Refinement Here we want to focus on the refinement of an initial disparity map. The initial disparity map can be e.g. the WTA solution of a matching volume or any other output of a stereo algorithm. One important approach of refinement algorithms is the fast bilateral solver (FBS) (Barron and Poole 2016). This algorithm refines the initial disparity estimate by solving an optimization problem containing an ℓ_2 smoothness- and an ℓ_2 data-fidelity term. The fast bilateral solver is the most related work to ours. However, in this work we replace the ℓ_2 norm with the robust ℓ_1 norm. More importantly, we additionally replace the hand-crafted smoothness term by a learnable multi-scale regularizer. Another refinement method was proposed by Gidaris and Komodakis (2017). They also start with an initial disparity map, detect erroneous regions and then replace and refine these regions to get a high-quality output. Pang et al. (2017) proposed to apply one and the same network twice. They compute the initial disparity map in a first pass, warp the second view with the initial disparity map and then compute only the residual to obtain a high quality disparity map. Liang et al. (2018) also improved the results by adding a refinement sub-network on top of the regularization network. We want to stress that the CNN based refinement networks (Liang et al. 2018; Pang et al. 2017) do not have a specialized architecture for refinement as opposed to the proposed model. Khamis et al. (2018) also focused on the refinement of coarse initial disparity maps in a hierarchical setting. They explicitly construct a light-weight network which is used to compute a residual between the initial disparity map and the refined map. Khamis et al. (2018) therefore uses only standard CNN building blocks with explicitly modeled residual connections. In difference, our method naturally provides the residual connections and we gain control and interpretability of the refinement process through our specialized, optimization based architecture. We show a direct comparison between both methods in the experiments and it will turn out that our approach is actually beneficial in interpretability and performance.

Learnable Optimization Schemes Learnable optimization schemes are based on unrolling the iterates of optimization algorithms. We divide the approaches into two categories. In the first category the optimization iterates are mainly used to utilize the structure during learning. For example in Riegler et al. (2016) 10 iterations of a TGV regularized variational method are unrolled and used for depth super-resolution. However, they learn only the step-sizes for the algorithm and keep the algorithm fixed. Similarly, in Vogel et al. (2018) unrolling 10k iterations of the FISTA (Beck and Teboulle 2009) algorithm is proposed. The second category includes methods where the optimization scheme is not only used to provide the structure, but it is also generalized by adding additional learnable parameters directly to the optimization iterates. For example Vogel and Pock (2017) proposed a primal-dual-network for low-level vision problems, where the authors learned the inference part of a Markov Random Field (MRF) model generalizing a primal-dual algorithm. Chen et al. (2015) generalized a reaction-diffusion model and successfully learned a model for image denoising. Based on Chen et al. (2015) a generalized incremental proximal gradient method was proposed in Kobler et al. (2017), where the authors showed connections to residual units (He et al. 2016). Wang et al. (2016) proposed proximal deep structured models where the authors perform inference with their recurrent network. Meinhardt et al. (2017) learned proximal operators using denoising networks for regularization. We built on the work of Chen et al., but specially designed the energy terms for the stereo task. Additionally, we allow to regularize on multiple spatial resolutions jointly and make use of the robust ℓ_1 function in our data-terms.

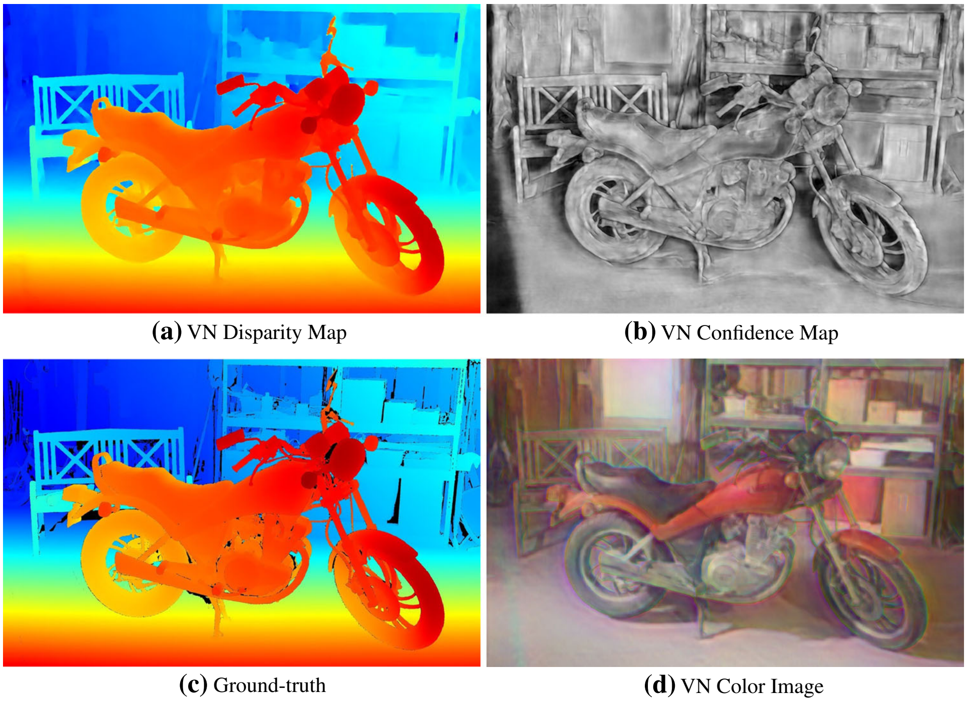
3 Method

We consider images to be functions $f : \Omega \to \mathbb{R}^C$, with $\Omega \subset \mathbb{N}_+^2$ and C is the number of channels which is 3 for RGB color images. Given two images f^0 and f^1 from a rectified stereo pair, we want to compute dense disparities d such that $f^0(x) = f^1(x - \tilde{d})$, i.e. we want to compute the horizontal shift $\tilde{d} = (d, 0)$ for each pixel $x = (x_1, x_2)$ between the reference image f^0 and the second image f^1. Here, we propose a novel variational refinement network for stereo which operates solely in 2D image space and is thus very efficient. The input to our method is an initial disparity map $\check{u} : \Omega \to [0, D]$, where D is the maximal disparity, a reference image f^0 and a pixel-wise confidence map $c : \Omega \to [0, 1]$. We explain the computation of the initial disparity- and confidence map in detail in Sect. 4. Right now, we just assume we have given the inputs.

The proposed variational network is a method to regularize, denoise and refine a noisy disparity map with learnable filters and learnable potential functions. Hence, the task we want to solve is the following: Given a noisy disparity map $\check{u}$, we want to recover the clean disparity with T learnable variational network steps. We do not make any assumptions on the quality of the initial disparity map, i.e. the initial disparity map may contain many strong outliers.

3.1 Collaborative Disparity Denoising

As the main contribution of this paper, we propose a method that performs a collaborative denoising in the joint color image, disparity and confidence space (see Fig. 2). Our model is based on the following three observations: (i) Depth discontinuities coincide with object boundaries, because we use the left image as the reference image (ii) discontinuities in the confidence image are expected to be close to left-sided object boundaries and (iii) the confidence image can be used as a pixel-wise weighting factor in the data fidelity term. Based on these three observations, we propose the following


(a) VN Disparity Map

(b) VN Confidence Map

(c) Ground-truth

(d) VN Color Image

Fig. 2 Collaborative disparity denoising. Our method produces three outputs: **a** the refined disparity map, **b** the refined confidence map and **d** the refined color image. **c** Shows the ground-truth image for comparison (black pixels = invalid). Note how our method is able to preserve fine details such as the spokes of the motorcycle

collaborative variational denoising model

$$\min_{\mathbf{u}} \mathcal{R}(\mathbf{u}) + \mathcal{D}(\mathbf{u}), \tag{1}$$

where $\mathbf{u} = (\mathbf{u}^{rgb}, u^d, u^c) : \Omega \to \mathbb{R}^5$, i.e. $\mathbf{u}$ contains for every pixel an RGB color value, a disparity value and a confidence value. $\mathcal{R}(\mathbf{u})$ denotes the collaborative regularizer and it is given by a multi-scale and multi-channel version of the Fields of Experts (FoE) model (Roth and Black 2009) with L scales and K channels.

$$\mathcal{R}(\mathbf{u}; \theta) = \sum_{l=1}^{L} \sum_{k=1}^{K} \sum_{x \in \Omega} \phi_k^l \left(\left(K_k^l A^l \mathbf{u} \right) (x) \right), \tag{2}$$

where $A^l : \mathbb{R}^5 \mapsto \mathbb{R}^5$ are combined blur and downsampling operators, $K_k^l : \mathbb{R}^5 \mapsto \mathbb{R}$ are linear convolution operators and $\phi_k^l : \mathbb{R} \mapsto \mathbb{R}$ are non-linear potential functions. The vector θ holds the parameters of the regularizer which will be detailed later. Note that multiple levels allow the model to operate on different spatial resolutions and therefore enables the denoising of large corrupted areas. Intuitively, the collaborative regularizer captures the statistics of the joint color, confidence and disparity space. Hence, it will be necessary to learn the linear operators and the non-linear potential functions from data. It will turn out that the combination of filtering in the joint color-disparity-confidence space at multiple hierarchical pyramid levels and specifically learned channel-wise potential functions make our model powerful.

$\mathcal{D}(\mathbf{u})$ denotes the collaborative data fidelity term and it is defined by

$$\mathcal{D}(\mathbf{u}; \theta) = \frac{\lambda}{2} \|\mathbf{u}^{rgb} - \mathbf{f}^0\|^2 + \mu \|u^c - c\|_1 + \nu \|u^d - \check{d}\|_{u^c,1}, \tag{3}$$

where θ is again a placeholder for the learnable parameters. The first term ensures that the smoothed color image $\mathbf{u}^{rgb}$ does not deviate too much from the original color image $\mathbf{f}^0$. We use here a quadratic ℓ_2 term, because we do not assume any strong outliers in the color image. The second term ensures that the smoothed confidence map stays close to the original confidence map. Here we use an ℓ_1 norm in order to deal with outliers in the initial confidence map. The last term is the data fidelity term of the disparity map. It is given by an ℓ_1 norm which is pixel-wise weighted by the confidence measure u^c, i.e.

$$\|r\|_{w,1} = \sum_{i=1}^{N} w_i |r_i|, \tag{4}$$

where $r, w \in \mathbb{R}^N$. Hence, data fidelity is enforced in high-confidence regions and suppressed in low-confidence regions. Note that the weighted ℓ_1 norm additionally ties the disparity map with the confidence map during the steps of the variational network.

Proximal Gradient Method (PGM) We consider a PGM (Parikh and Boyd 2014) whose iterates are given by

$$\mathbf{u}_{t+1} = \text{prox}_{\alpha_t \mathcal{D}}(\mathbf{u}_t - \alpha_t \nabla \mathcal{R}(\mathbf{u}_t)), \tag{5}$$

where α_t is the step-size, $\nabla\mathcal{R}(\mathbf{u}_t)$ is the gradient of the regularizer which is given by

$$\nabla\mathcal{R}(\mathbf{u}) = \sum_{l=1}^{L}\sum_{k=1}^{K}(K_k^l A^l)^T \rho_k^l \left(K_k^l A^l \mathbf{u}\right), \tag{6}$$

where $\rho_k^l = \text{diag}((\phi_k^l)')$. Hence, ρ_k^l is the derivative of the potential function and can be interpreted as the activation function in our regularizer. A visual comparison between potential and activation-functions is shown in Fig. 10. $\text{prox}_{\alpha_t\mathcal{D}}$ denotes the proximal operator with respect to the data fidelity term, which is defined by

$$\text{prox}_{\alpha_t\mathcal{D}}(\tilde{\mathbf{u}}) = \arg\min_{\mathbf{u}} \mathcal{D}(\mathbf{u}) + \frac{1}{2\alpha_t}\|\mathbf{u} - \tilde{\mathbf{u}}\|_2^2. \tag{7}$$

Note that the proximal map allows to handle the non-smooth data fidelity terms such as the ℓ_1 norm. Additionally, there is a strong link between proximal gradient methods and residual units which allows to incrementally reconstruct a solution (see Fig. 1).

Proximal Operators for the Data Terms The proximal operator in Equation 7 is an optimization problem itself. We need to compute the proximal operator for the ℓ_1 and the ℓ_2 function. Both can be computed in closed form. Therefore, let us consider the proximal operator of a function f:

$$\text{prox}_{\tau f}(\tilde{u}) = \arg\min_{u} f(u) + \frac{1}{2\tau}\|u - \tilde{u}\|^2. \tag{8}$$

First, we present the result of the proximal operator for the ℓ_2 function

$$f(u) = \frac{\lambda}{2}\|u - u_0\|^2. \tag{9}$$

Inserting Equation 9 into Equation 8 and setting the derivative w.r.t. u to zero, we can compute the optimal solution u^* with

$$u^* = \frac{\tilde{u} + \tau\lambda u_0}{1 + \tau\lambda}, \tag{10}$$

where for the color image data term, $u_0 = I_0$ and $\tilde{u} = u^{rgb}$.

Similarly, we compute the proximal operator of the weighted ℓ_1 function

$$f(u) = \gamma\|u - u_0\|_{w,1} = \gamma\sum_{x\in\Omega} w(x)|u(x) - u_0(x)|. \tag{11}$$

The absolute function is not differentiable at 0 and therefore the optimality condition requires the sub-differential to contain 0. The closed form solution of the proximal operator Equation 8 with f being the ℓ_1 function as defined in Equation 11 is given by

$$u^* = u_0 + \max(0, |\tilde{u} - u_0| - \tau\gamma w) \cdot \text{sign}(\tilde{u} - u_0). \tag{12}$$

Thus, for the disparity data term we set $w = c$ and $u_0 = \check{d}$. Since the confidence u^c is present in the confidence data term, and linearly dependent in the disparity data term, we make use of the identity

$$\text{prox}_{\tau f}(\tilde{u}) = \text{prox}_{\tau g}(\tilde{u} - a) \tag{13}$$

for functions $f(u) = g(u) + a^T u + b$. In our setting $g(u) = \mu\|u^c - c\|_1$ is the confidence data-term and $a = |u^d - \check{d}|$.

Variational Network Our collaborative denoising algorithm consists of performing a fixed number of T iterations of the proximal gradient method Equation 5. In order to increase the flexibility we allow the model parameters to change in each iteration.

$$\mathbf{u}_{t+1} = \text{prox}_{\alpha_t\mathcal{D}(\cdot,\theta_t)}(\mathbf{u}_t - \alpha_t\nabla\mathcal{R}(\mathbf{u}_t, \theta_t)),\ 0 \le t \le T - 1 \tag{14}$$

Following Chen et al. (2015), Kobler et al. (2017) we parametrize the derivatives of the potential functions ρ_k^l in (6) using Gaussian radial basis functions (RBF)

$$\rho_k^{l,t}(s) = \beta_k^{l,t}\sum_{b=1}^{B} w_{k,b}^{l,t}\exp\left(-\frac{(s - \gamma_b)^2}{2\sigma^2}\right) \tag{15}$$

to allow learning of appropriate activation functions from the data. We sample the means γ_b regularly on the interval $[-3, 3]$, σ is the standard deviation of the Gaussian kernel and $\beta_k^{l,t}$ is a scaling factor. The linear operators $K_k^{l,t}$ are implemented as multi-channel 2D convolutions with convolution kernels $\kappa_k^{l,t}$. In summary, the parameters in each step are given by $\theta_t = \{\kappa_k^{l,t}, \beta_k^{l,t}, w_{k,b}^{l,t}, \mu^t, \nu^t, \lambda^t, \alpha_t, \}$.

4 Computing Inputs

Our proposed refinement method can be applied to any stereo method coming along with a cost-volume, which is the case for the majority of existing stereo methods.

Probability Volume Assume we have given a cost-volume $v : \Omega \times \{0, \ldots, D - 1\} \to \mathbb{R}$, where smaller costs mean a higher likelihood of the respective disparity values. In order to map the values onto probabilities $p : \Omega \times \{0, \ldots, D - 1\}$, we make use of the "softmax" function, that is

$$p(x, d) = \frac{\exp(\frac{-v(x,d)}{\eta})}{\sum_{d'=0}^{D-1}\exp(\frac{-v(x,d')}{\eta})}, \tag{16}$$

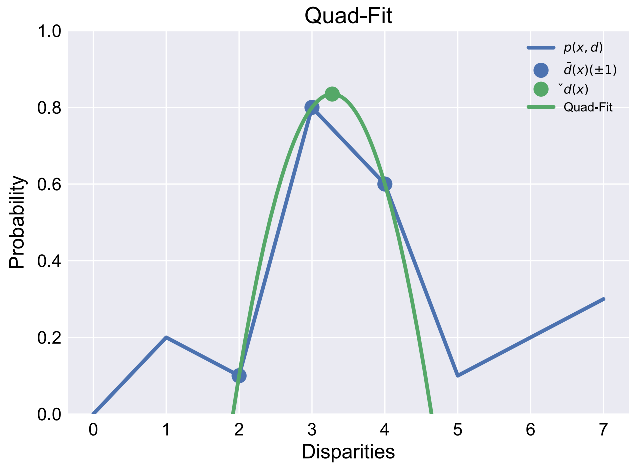


Fig. 3 Visualization of the quadratic fitting. We select the points next to the maximum value and fit a quadratic function. Computing the extremum of the quadratic functions yields the refined disparity and the refined probability

where η influences the smoothness of the probability distribution.

Initial Disparity Map

From Equation 16 we can compute the WTA solution by a pixel-wise arg max over the disparity dimension, i.e.,

$$\bar{d}(x) \in \arg\max_{d} \; p(x,d). \tag{17}$$

Moreover, we compute a sub-pixel accurate disparity map $\check{d}(x)$ by fitting a quadratic function to the probability volume. This is equivalent to performing one step of Newton's algorithm:

$$\check{d}(x) = \bar{d}(x) - \frac{\delta^{+}(p(x,\cdot))(\bar{d}(x))}{\delta^{-}(\delta^{+}(p(x,\cdot)))(\bar{d}(x))}, \tag{18}$$

where $\delta^{\{+,-\}}$ denote standard forward and backward differences in the disparity dimension. Furthermore, we compute the refined value of the probabilities, denoted as $\check{p}(x)$, via linear interpolation in the probability volume.

In the joint training of our feature network and the regularization network we need to backpropagate the gradient through the refined disparities. Therefore, we must compute the gradient of our sub-pixel accurate disparity map w.r.t. the probability volume. The gradient is non-zero only for the supporting points of the quadratic function (shown in blue in Fig. 3) and it is given by

$$\frac{\partial \check{d}(x)}{\partial p(x,d)} = \begin{cases} \frac{\delta^{c}(p(x,\cdot))(\bar{d}(x))}{(\delta^{-}(\delta^{+}(p(x,\cdot)))(\bar{d}(x)))^2} & \text{if } d = \bar{d}(x) \\ \frac{\delta^{+}(p(x,\cdot))(\bar{d}(x))}{(\delta^{-}(\delta^{+}(p(x,\cdot)))(\bar{d}(x)))^2} & \text{if } d = \bar{d}(x) - 1 \\ \frac{\delta^{-}(p(x,\cdot))(\bar{d}(x))}{(\delta^{-}(\delta^{+}(p(x,\cdot)))(\bar{d}(x)))^2} & \text{if } d = \bar{d}(x) + 1 \\ 0 & \text{else,} \end{cases} \tag{19}$$

where $\delta^{\{+,-,c\}}$ are standard forward-, backward- and central-differences in the disparity dimension. Note, that we overcome the problem of the non-differentiable arg min function with the fitting of the quadratic function. Figure 3 shows a visualization of the quadratic fitting procedure.

Initial Confidence Measure The computation of a confidence measure of the stereo results is important for many applications and a research topic on its own (Hu and Mordohai 2012). Here we take advantage of the probabilistic nature of our matching costs $\check{p}(x)$. Moreover, we make use of geometric constraints by using a left-right (LR) consistency check, where the left and right images are interchanged. This allows us to identify occluded regions. We compute the probability of a pixel being not occluded as

$$p_o(x) = \frac{\max(\varepsilon - \text{dist}_{lr}(x), 0)}{\varepsilon} \in [0,1], \tag{20}$$

where

$$\text{dist}_{lr}(x) = |\check{d}_l(x) + \check{d}_r(x + \check{d}_l(x))| \tag{21}$$

is the disparity difference between the left prediction $\check{d}_l$ and the right prediction $\check{d}_r$ and the parameter ε acts as a threshold and is set to $\varepsilon = 3$ in all experiments. The final confidence measure is given by

$$c(x) = \check{p}(x) p_o(x) \in [0,1]. \tag{22}$$

Thus, we define our total confidence as the product of the matching confidence and the LR confidence. Most of the pixels not surviving the LR check are pixels in occluded regions. To get a good initialization for these pixels as well, we inpaint the disparities of these pixels from the left side. The experiments show that this significantly increases the performance of the model (see Table 2).

5 Learning

In this section we describe our learning procedure for the collaborative denoising model. To remove scaling ambiguities we require the filter kernels $\kappa_k^{l,t}$ to be zero-mean and to have an ℓ_2 norm ≤ 1. Moreover, we constrain the weights of the RBF kernels to have an ℓ_2 norm ≤ 1, too. This is defined with the following convex set:

$$\Theta = \{\theta_t : \|\kappa_k^{l,t}\| \leq 1, \; \sum_{j=1}^{J} \kappa_{k,j}^{l,t} = 0, \; \|w_k^{l,t}\| \leq 1\} \tag{23}$$

For learning, we define a loss function that measures the error between the last iterate of the disparity map u_T^d and the ground-truth disparity d^*. Note that we do not have a loss

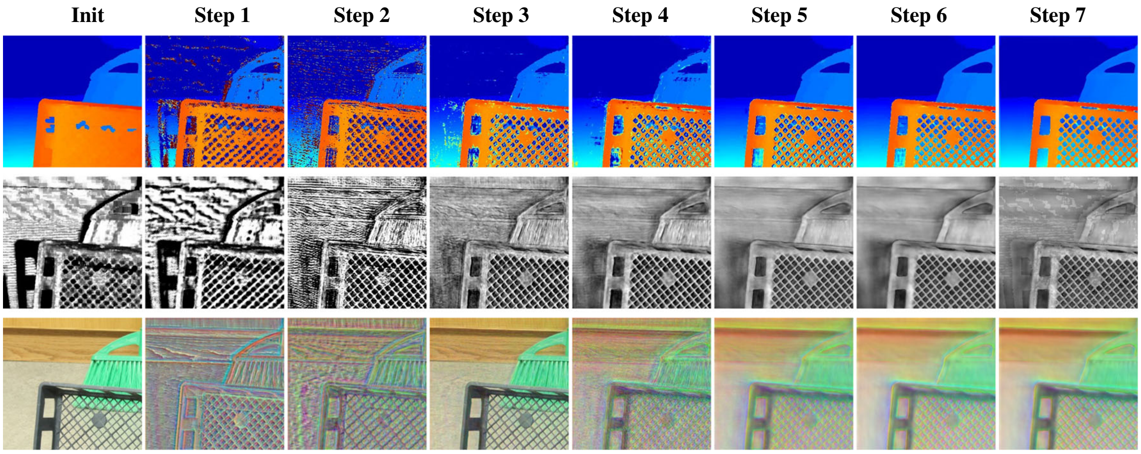


Fig. 4 Visualization of steps in the VN. Top to bottom: disparity map, confidence map, image. Left to right: Initialization, VN Steps 1–7. Note how the color image and the confidence map help to restore very fine details in the disparity map

function for the confidence and the color image. Their aim is rather to support the disparity map to achieve the lowest loss. We use a truncated Huber function of the form

$$\min_{\theta\in\Theta} \sum_{s=1}^{S} \sum_{x\in\Omega} \min\left(|u^{d}_{s,T}(x,\theta) - d^{*}_{s}(x)|_{\delta},\ \tau \right) \tag{24}$$

where τ is a truncation value, s denotes the index of the training sample and

$$|r|_{\delta} = \begin{cases} \frac{r^2}{2\delta} & \text{if } |r| \le \delta \\ |r| - \frac{\delta}{2} & \text{else} \end{cases} \tag{25}$$

is the Huber function.

Implementation Details We implemented our model in the PyTorch machine learning framework[1]. We train the refinement module for 3000 epochs with a learning rate of 10^{-3} with a modified projected Adam optimizer (Kingma and Ba 2014). While in Kingma and Ba (2014) the stepsize is adjusted element-wise, we use a constant stepsize within each parameter block. This is necessary to ensure an orthogonal projection of the parameter blocks onto the constraint set Θ. After 1500 epochs we reduce the truncation value τ from ∞ to 3.

6 Experiments

We split the experiments into two parts. In the first part we evaluate architectural choices based on the WTA result of a matching network and compare with the Fast Bilateral Solver (FBS) (Barron and Poole 2016) and the StereoNet (SN) refinement method of Khamis et al. (2018). In the second part, we use the best architecture and train a variational network for refining the disparity maps computed by the CNN-CRF method (Knöbelreiter et al. 2017). We use this method to participate in the publicly available stereo benchmarks Middlebury 2014 and Kitti 2015. To ensure a fair comparison we choose methods with similar numbers of parameters and runtimes. Figure 4 shows how our method constructs the final result. The method recovers step-by-step fine details with the guidance of the confidences and the color image. Qualitative results on the official tests sets of Middlebury and Kitti are visualized in Figs. 5 and 6 and additional qualitative results are shown in Figs. 7 and 8.

Kitti 2015 The Kitti 2015 dataset (Menze and Geiger 2015) is an outdoor dataset specifically designed for autonomous driving. It contains 200 images with available ground-truth to train a model and 200 images with withheld ground-truth which is used for testing the models on previously unseen data. The ground-truth is captured using a laser scanner and is therefore sparse in general. The cars are densified by fitting CAD models into the laser point-cloud. We report the *badX* error metric for occluded (occ) and non-occluded (noc) pixels with $X = 3$. In the badX measure the predicted disparity $\hat{d}$ is treated incorrect, if the distance to the ground-truth disparity d^* is larger than X.

Middlebury 2014 The Middlebury 2014 stereo dataset (Scharstein et al. 2014) is orthogonal to the Kitti 2015 dataset. It consists of 153 high resolution indoor images with highly precise dense ground-truth. The challenges in the Middlebury dataset are large, almost untextured regions, huge occluded regions, reflections and difficult lighting conditions. The generalization capability of the method is evaluated on a 15 image test-set with withheld ground-truth data. We report all available metrics, i.e., bad{0.5, 1, 2, 4} errors, the average error (avg) and the root-mean-squared error (rms).

[1] https://pytorch.org

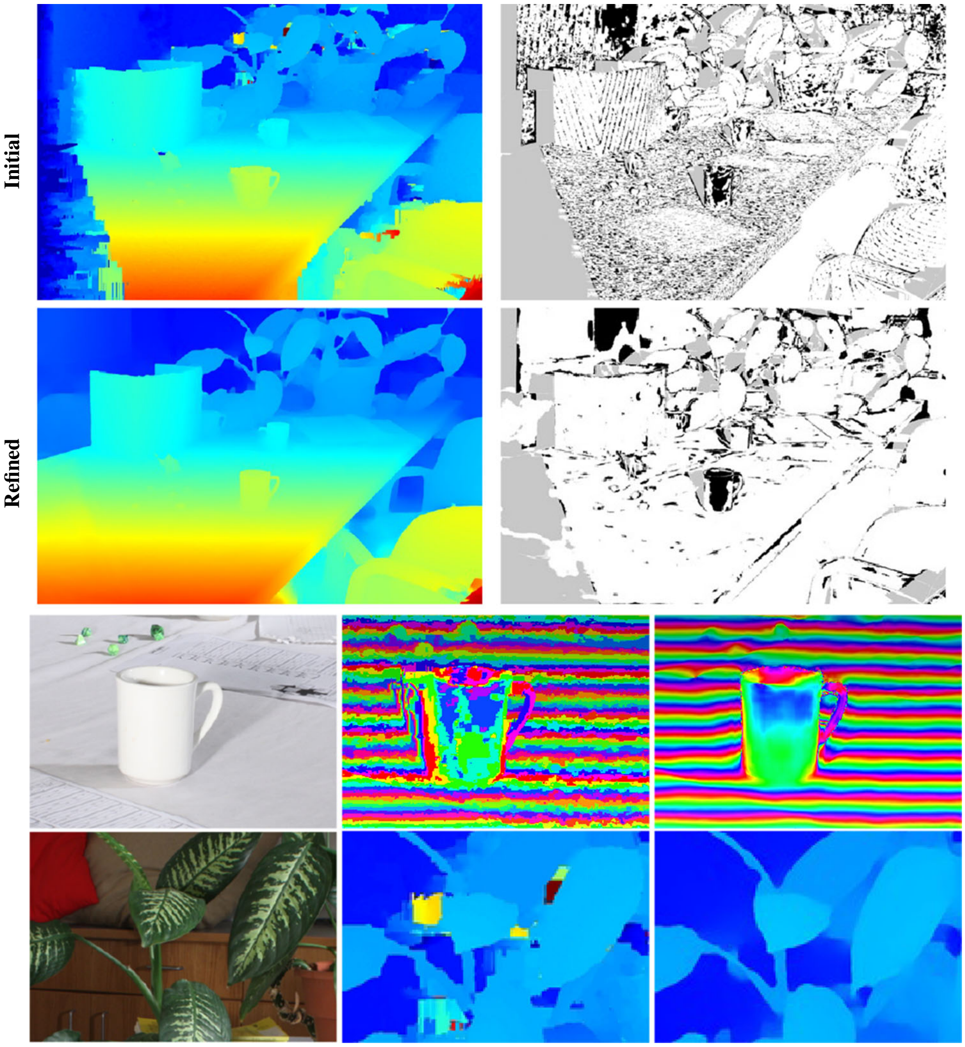


Fig. 5 Qualitative results on the Middlebury test set. Top-group: Left: color-coded disparity maps ranging from blue = far away to red = near. Right: Error maps, where white = correct and black = incorrect. The top row shows the initial disparity map (=input to the VN) and the bottom row shows our refined result. Bottom group: Close-up results with input-image, initial disparity map and refined disparity map from left to right. The second column shows a high-frequency visualization of the disparity map

6.1 Ablation Study

To find the most appropriate hyper parameters for the proposed method, we generate our initial disparity map with a simple feature network. The learned features are then compared using a fixed matching function for a pre-defined number of discrete disparities.

Feature Network Our feature network is a modified version of the U-Net (Ronneberger et al. 2015; Long et al. 2015) which we use to extract features suitable for stereo matching. We keep the number of parameters low by only using 64 channels at every layer. The output of our feature network is thus a 64-dimensional feature vector for every pixel. Table 1 shows the architecture in tabular format.

Feature Matching Next, we use the extracted features ψ^0 from the left image and ψ^1 from the right image to compute a matching score volume $\tilde{p} : \Omega \times \{0, \dots, D-1\} \to \mathbb{R}$ with

$$\tilde{p}(x, d) = \langle \psi^0(x),\ \psi^1(x - \tilde{d}) \rangle. \tag{26}$$

We follow Sect. 4 to compute the inputs for the variational network.

Ablation Study We systematically remove parts of our method in order to show how the final performance is influenced by the individual parts. Table 2 shows an overview of all experiments. First, we investigate the influence of our data-terms, the disparity data-term, the confidence data-term and the RGB image data-term. The study shows that each of the data-terms positively influences the final performance.

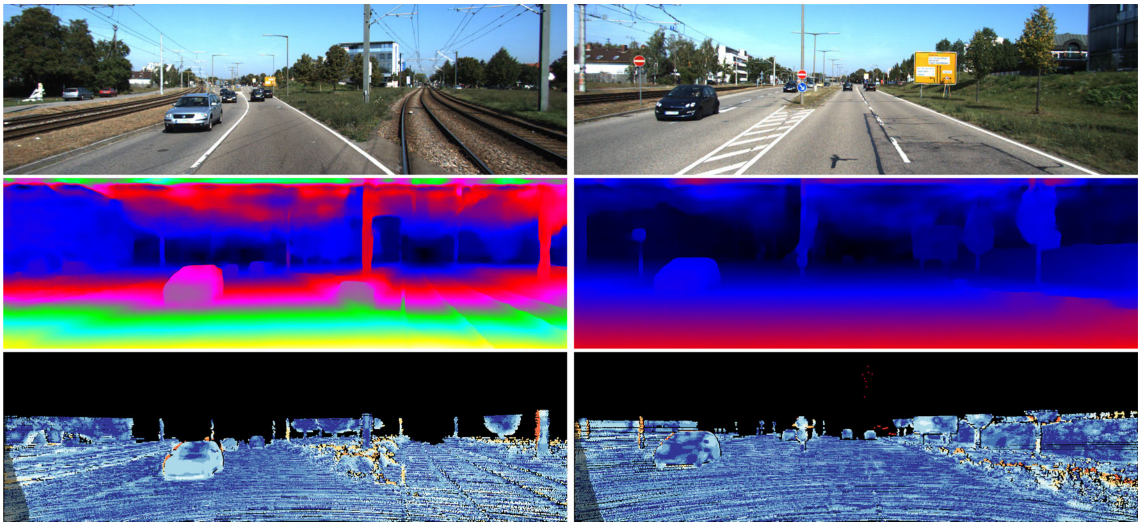


Fig. 6 Qualitative results on the Kitti 2015 test set. Top-to-bottom: Reference image, disparity map which is color coded with blue = far away to yellow = near, error map, where blue = correct disparity, orange = incorrect disparity

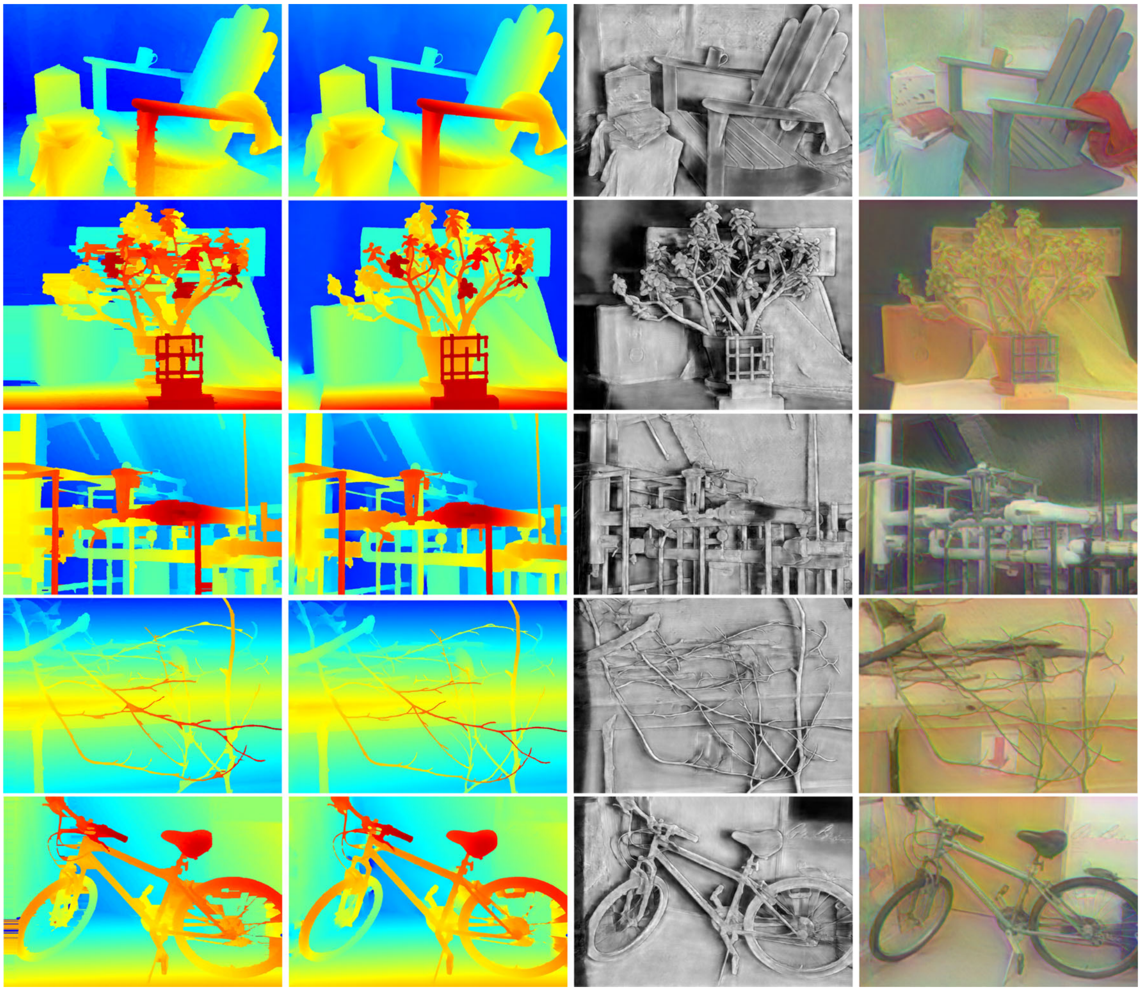


Fig. 7 Results of $\text{VN}_4^{7,11}$ on half size (H) Middlebury images. Left to right: initial disparity map, refined disparity map, confidences and color image. Our model learns to use object edges to guide the denoising of the disparity map. Best viewed with zoom on the PC

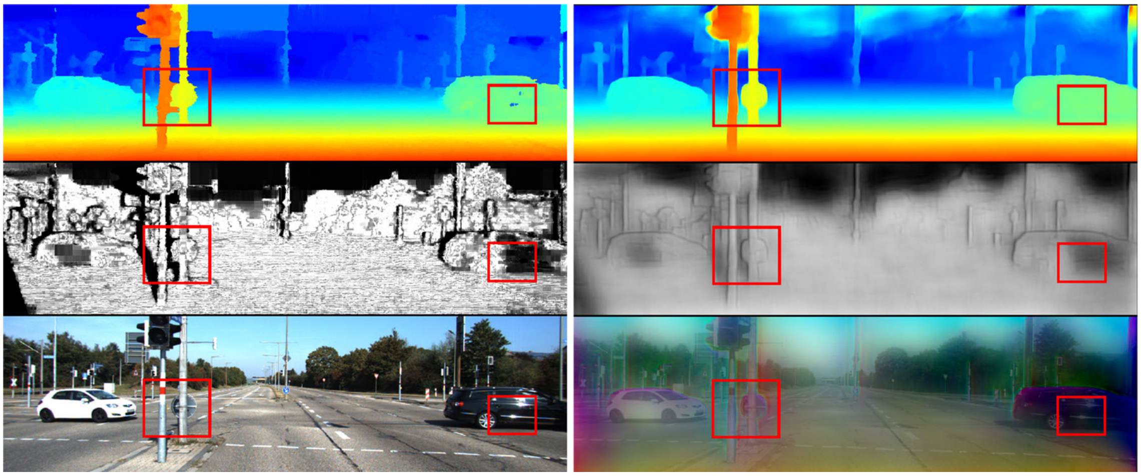


Fig. 8 Refinement on Kitti. Top to bottom are the disparity map, the confidence map and the color image. Left: initial results, right refined results. Note especially the highlighted boxes, where artefacts are corrected and fine details are recovered

Table 1 Detailed architecture of our multi-level feature network

Layer	KS	Resolution	Channels	Input
conv00	3	$W \times H / W \times H$	3/64	Image
conv01	3	$W \times H / W \times H$	64/64	conv00
pool0	2	$W \times H / \frac{W}{2} \times \frac{H}{2}$	64/64	conv01
conv10	3	$\frac{W}{2} \times \frac{H}{2} / \frac{W}{2} \times \frac{H}{2}$	64/64	pool0
conv11	3	$\frac{W}{2} \times \frac{H}{2} / \frac{W}{2} \times \frac{H}{2}$	64/64	conv10
pool1	2	$\frac{W}{2} \times \frac{H}{2} / \frac{W}{4} \times \frac{H}{4}$	64/64	conv10
conv20	3	$\frac{W}{4} \times \frac{H}{4} / \frac{W}{4} \times \frac{H}{4}$	64/64	pool1
conv21	3	$\frac{W}{4} \times \frac{H}{4} / \frac{W}{4} \times \frac{H}{4}$	64/64	conv20
deconv1	3	$\frac{W}{4} \times \frac{H}{4} / \frac{W}{2} \times \frac{H}{2}$	64/64	conv21
conv12	3	$\frac{W}{2} \times \frac{H}{2} / \frac{W}{2} \times \frac{H}{2}$	128/64	{deconv1, conv11}
conv13	3	$\frac{W}{2} \times \frac{H}{2} / \frac{W}{2} \times \frac{H}{2}$	64/64	conv12
deconv0	3	$\frac{W}{2} \times \frac{H}{2} / W \times H$	64/64	conv12
conv02	3	$W \times H / W \times H$	128/64	{deconv0, conv01}
conv03	3	$W \times H / W \times H$	64/64	conv02

KS denotes the kernel size, *Resolution* contains the spatial resolution of the input and output, respectively and *Channels* contain the number of input and output feature channels, respectively. We use curly brackets to indicate a concatenation of feature maps. We use the LeakyReLU activation function after every convolution layer

Especially, adding the original input image significantly increases the performance. This can be e.g. seen in Fig. 4, where the information of how the basket needs to be reconstructed, is derived from the input image. In the second part of the study, we evaluate different variational network architectures. To make the comparison as fair as possible, we chose the variants such that the total number of parameters is approximately the same for all architectures. The experiments show, that a compromise between number of steps, pyramid levels and filter-size yields the best results. The best performing model is the model $\text{VN}_4^{7,5}$, where the filter-size is set to 5 × 5 for 4 pyramid levels and 7 steps. The average runtime of this VN is as low as 0.09s on an NVidia 2080Ti graphics card.

We use the model $\text{VNS}_4^{30,5}$ to run another experiment where we share the parameters over all iterations in the VN. This shows that we can use the same procedure also in a pure optimization setting. Here, we have significantly less parameters, i.e. we have only 20k parameters in the VN while the non-shared version has 140K parameters. We trained the shared model for $T = 30$ iterations and show the result in

Table 2 Ablation study on the Kitti 2015 dataset

Model	Conf	Img	OccIp	Joint	Error [bad3]		#P
					occ	noc	
WTA					8.24	6.78	480k
WTA + $\text{VN}_4^{7,5}$			✓		5.42	4.68	50k
WTA + $\text{VN}_4^{7,5}$	✓	✓			5.12	3.98	140k
WTA + $\text{VN}_4^{7,5}$	✓		✓		4.43	3.90	73k
WTA + $\text{VN}_4^{7,5}$		✓	✓		3.77	3.07	118k
WTA + $\text{VN}_4^{7,5}$	✓	✓	✓		3.46	2.72	140k
WTA + $\text{VN}_4^{7,5}$	✓	✓	✓	✓	**3.37**	**2.55**	140k
WTA + $\text{VN}_3^{5,7}$	✓	✓	✓	✓	3.43	2.58	133k
WTA + $\text{VN}_2^{8,7}$	✓	✓	✓	✓	3.62	2.97	141k
WTA + $\text{VN}_4^{14,3}$	✓	✓	✓	✓	4.37	3.71	136k
WTA + $\text{VN}_5^{11,3}$	✓	✓	✓	✓	4.25	3.49	134k
WTA + $\text{VNS}_4^{30,5}$	✓	✓	✓		5.24	4.35	20k
WTA + FBS (Barron and Poole 2016)	✓	✓	✓		7.48	6.08	–
WTA + SN (Khamis et al. 2018)		✓	✓		4.02	3.11	114k
WTA + SN (Khamis et al. 2018)	✓	✓	✓		3.78	2.88	114k

Conf = Confidences, Img = Image, OccIp = Occlusion inpainting, Joint = joint training, Shared = shared VN parameters, #P = number of parameters. The super-script indicates the number of steps and the filter-size while the sub-script indicates the number of levels in the variational network. $\text{VN}_4^{7,5}$ is therefore a variational network with 7 steps and 4 levels

Table 2. The shared model needs more iterations to converge to a good result.

Additionally, we compare with the FBS, because the FBS is defined via a similar optimization problem as our VN. We therefore use exactly the same inputs as we did in our method, i.e., the refined WTA solution $\check{d}$, our confidence measure c and the RGB input image. To ensure the best performance for the FBS, we performed a grid-search over its hyper-parameters on the Kitti dataset. As shown in Table 2 the FBS clearly improves the performance upon the initial solution, but the FBS cannot compete with the proposed method.

The next method we want to directly compare with is the StereoNet (Khamis et al. 2018). StereoNet performs a hierarchical refinement on top of initial disparity maps and is similar lightweight as our model. The refinement in the StereoNet approach is performed with a refinement module consisting of 6 residual blocks and an input and an output mapping layer. While our model contains residual connections implicitly through the optimization structure the authors of StereoNet explicitly designed them in their architecture. The receptive field is similar to ours, but instead of downsampling the authors used dilated convolutions. The inputs to the StereoNet are the RGB color image and the initial disparity map. We will investigate the performance of StereoNet on top of our feature net in the original setting i.e. without the confidences and additionally we show the benefit of using confidences in the StereoNet as well in Table 2. The ablation study shows that the proposed VN compares favorable to the StereoNet in both variants, with and without additional confidences as input. Thus we can conclude that the structure arising from an optimization problem is also beneficial in terms of final performance in the learning setting.

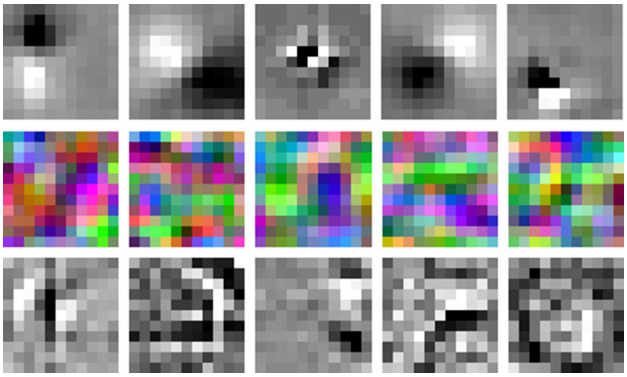


Fig. 9 Visualization of the learned filters of our model. Top to bottom: filters of the disparity map, filters of the RGB color image and filters of the confidence map

6.2 Benchmark Performance

We use our method on top of the CNN-CRF (Knöbelreiter et al. 2017) stereo method for the official test set evaluation

Table 3 Performance on the Middlebury 2014 benchmark

Method	Middlebury 2014 (Train)						
	bad0.5	bad1	bad2	bad4	avg	rms	time
PSMNet (Chang and Chen 2018)	90.0 (90.8)	78.1 (79.9)	58.5 (61.8)	32.2 (37.3)	9.60 (13.3)	21.7 (27.1)	2.62
PDS (Tulyakov et al. 2018)	54.2 (58.2)	26.1 (31.9)	11.4 (16.7)	5.10 (9.09)	1.98 (3.26)	9.10 (12.7)	10
MC-CNN (Žbontar and LeCun 2016)	42.1 (49.0)	20.5 (29.8)	11.7 (21.5)	7.94 (17.7)	3.87 (12.8)	16.5 (37.5)	1.26
CNN-CRF (Knöbelreiter et al. 2017)	56.1 (60.5)	25.1 (32.5)	10.8 (18.9)	6.12 (13.7)	2.30 (9.57)	9.89 (32.0)	3.53
(Knöbelreiter et al. 2017) + VN (ours)	***41.8*** (***46.6***)	***17.1*** (***23.0***)	***7.05*** (***12.1***)	***2.96*** (***6.49***)	***1.21*** (***2.06***)	***5.80*** (***8.57***)	4.06
Method	**Middlebury 2014 (Test)**						
	bad0.5	bad1	bad2	bad4	avg	rms	time
PSMNet (Chang and Chen 2018)	81.1 (82.9)	63.9 (67.3)	42.1 (47.2)	23.5 (27.2)	6.68 (8.78)	19.4 (23.3)	2.62
PDS (Tulyakov et al. 2018)	58.9 (62.8)	21.1 (38.3)	14.2 (21.0)	6.98 (**12.6**)	3.27 (6.90)	15.7 (27.5)	10.3
MC-CNN (Žbontar and LeCun 2016)	**41.3** (***48.5***)	**18.0** (**28.4**)	**9.47** (**20.6**)	6.7 (17.7)	4.37 (19.3)	22.4 (55.7)	1.26
CNN-CRF (Knöbelreiter et al. 2017)	60.9 (65.1)	31.9 (39.4)	12.5 (21.9)	**6.61** (15.9)	3.02 (15.7)	14.4 (49.0)	3.53
(Knöbelreiter et al. 2017) + VN (ours)	*56.2* (*61.0*)	*30.0* (*37.5*)	14.2 (22.4)	7.71 (*14.6*)	***2.49*** (***4.98***)	***10.8*** (***17.3***)	4.06

We report the numbers of the official online system for non-occluded (all) pixels. Top = Official training set, Bottom = Official test set. Bold font: Overall best. Italic font = improvement of baseline. Note especially the significant improvement of the continuous error metrics avg and rms on all pixels

Table 4 Performance on the Kitti 2015 benchmark

Method	Kitti 2015 (train)		Kitti 2015 (test)	
	noc	all	noc	all
PSMNet (Chang and Chen 2018)	–	**1.83**	**2.14**	**2.32**
PDS (Tulyakov et al. 2018)	–	–	2.36	2.58
MC-CNN (Žbontar and LeCun 2016)	–	–	3.33	3.89
CNN-CRF (Knöbelreiter et al. 2017)	–	4.04	4.84	5.50
(Knöbelreiter et al. 2017) + VN (ours)	***1.90***	*2.04*	*4.45*	*4.85*

We provide the official bad3 error metric on non-occluded (noc) and all pixels on the training set (left) and on the test set (right). The VN improves the baseline method on both metrics

(see Tables 3 and 4). We set the temperature parameter $\eta = 0.075$ in all experiments.

We used the model $\text{VN}_4^{7,5}$ on the Kitti dataset, since this model performed best in the ablation study. As shown in Tabel 4 we reduce the bad3 error in both, occluded and in non-occluded regions. The relative improvement brought by the VN is 8% for occluded pixels and 12% for all pixels. Thus, the experiment shows that the VN is especially beneficial in occluded pixels. Figure 6 shows qualitative results with the corresponding error maps on the Kitti test set.

On the Middlebury benchmark we use the model $\text{VN}_4^{7,11}$ for all evaluations, where we have choosen a larger filter size to account for the high-resolution images in this benchmark. We compare the errors on the training set with the errors on the test set (Table 3) and observe first that our method shows a significant improvement over the baseline method on the continuous error metrics avg and rms on both the test set and the training set in non-occluded and all pixels. This is understandable, because we have used the continuous Huber loss (24) for training the VN. The Huber loss is a combination of the ℓ_1 and ℓ_2 error and thus minimizes the continuous error metrics. At the time of submission the VN ranks 8^{th} out of 147 methods on the continuous rms error metric which confirms the good performance. However, we can also see that minimizing the continuous error metric does not necessarily yield better results for the badX error measure, which can be explained by the fact that the Huber loss does not provide a good proxy for the badX measures. While the VN can at least slightly improve the results on bad$\{0.5,1,4\}$, the error is slightly increased on the bad2 error on the test set compared to Knöbelreiter et al. (2017). This is in contrast to the training set, where the VN can improve on all badX error measures as well. Similar to the behavior on the Kitti dataset, the benefit of the VN is significant especially in occluded regions, where we have reduced the average error from 15.7 to 4.98 which is a relative improvement of almost 70% over the baseline method. This is also noticeable visually in Fig. 7, where the VN is often able to perfectly fill in occluded regions. To conclude, we have seen that the VN yields state-of-the-art (SOTA) results using continuous error metrics for evaluation, but trails SOTA on the badX error metric. Figure 5 shows a qualitative example of the Middlebury test set. Note that the tabletop is nice and smooth while the sharp edges of the objects are very well preserved.

7 Analyzing the VN

One of the main benefits of a variational network compared to other CNNs is the interpretability of the VN. Due to the optimization-like architecture, we can visualize the individual steps, interpret the learned filters and activation functions, compute eigen-disparity maps, which are non-linear eigenvectors of our learned regularizer, and investigate the quality of our confidence maps. We address all these properties of our model in the next sub-sections.

7.1 Learned Filters and Activation Functions

In this section we investigate the structure of our learned filters and plot the learned activation functions. Visualizing the filters can be easily done in the VN, because our filters always operate in the 2D image space directly. Note that this visualization technique is not possible in other CNNs, because the filters are usually 3D in convolution layers and thus, they can not be directly plotted. For our visualization we split up the five learned spatial 2D filters into three parts which can then be interpreted as the filters for the disparity map, for the color image and for the confidence maps. Note that the RGB color filter uses three of the five channels. Figure 9 shows selected filter kernels. The first row contains filters for the disparity map, the second row contains filters for the RGB color image and the third row contains filters for the confidence map. Note that the learned filters contain structure which makes them interpretable. The structure can be clearly seen in the disparity filters, which look like Gabor filters. The color filters contain structure as well and can be interpreted as texture filters. The middle filter could be an ellipsoidal blob detector. The confidence filters seem to capture the edge information between low confident and high confident regions around edges. The color- and confidence-filters are not as smooth as the disparity filters, which can be explained by the fact that we did not use any loss function on the color and confidence channel. The structure in the filters suggests that our model actually captures statistics of how to appropriately refine disparity maps, confidence maps and color images jointly.

Figure 10 shows the learned activation functions. We can integrate the learned activation functions (blue) to get the potential functions (green) used in our energy. Similar as for the learned filters, the learned activation functions can also be interpreted. We plot in Fig. 10 prototypical learned potential functions of our model. Starting from left to right we can see instances of a Student-t potential, the Mexican hat function, a truncated Huber function and a double-well potential. For comparison, we show the analytic potential functions in the last row in red and state the corresponding analytic expressions. We can e.g. see that our model has learned to be robust against outliers with the first (Student-t) and the third (truncated Huber) potential function. We also observe that we have found similar functions as e.g. Chen et al. (2015) for denoising and Zhu et al. (1998).

7.2 Shared Parameters

In this section we restrict the parameters to be shared for all iterations of the VN. Since we are in a pure optimization setting we can perform additional experiments such as computing eigen disparity maps, eigen image and eigen confidence maps.

Using shared parameters during the iterations of the VN requires us to change Equation 14 to

$$\mathbf{u}_{t+1} = \operatorname{prox}_{\alpha\mathcal{D}(\cdot,\theta)}(\mathbf{u}_t - \alpha\nabla\mathcal{R}(\mathbf{u}_t,\theta)),\ 0 \leq t \leq T-1 \quad (27)$$

where we removed the index t in all parameters. Next, we compute the eigenmodes of the learned regularizer. We therefore use the same shared model as in the ablation study in Table 2, i.e. $\mathrm{VNS}_4^{30,5}$.

7.3 Eigenmodes of the VN

We show how we can compute eigenmodes of our learned regularizer in the refinement VN by adapting the approach of Effland et al. (2020). This allows us to visualize the eigenmodes of our regularizer as images and we can thus interpret them. The eigenimages give insights into what the regularizers has learned, since they reveal prototypical structures yielding a low energy of the regularizer.

Recall the classical eigenvalue/eigenvector problem

$$Au = \lambda u \quad (28)$$

with $A \in \mathcal{S}^{N\times N}$, where $\mathcal{S}$ is the space of symmetric positive definite matrices. λ and u are the sought eigenvalue/eigenvector pairs. To motivate the way we compute our eigenmodes, we note that the left hand side of Equation 28 can also be derived using the gradient of a quadratic function $\mathcal{Q}(u) = \frac{1}{2}u^T Au$, where we get

$$\nabla\mathcal{Q}(u) = \lambda u. \quad (29)$$

In order to apply the eigenmode analysis to our refinement VN, we replace the quadratic function $\mathcal{Q}(u)$ with our non-

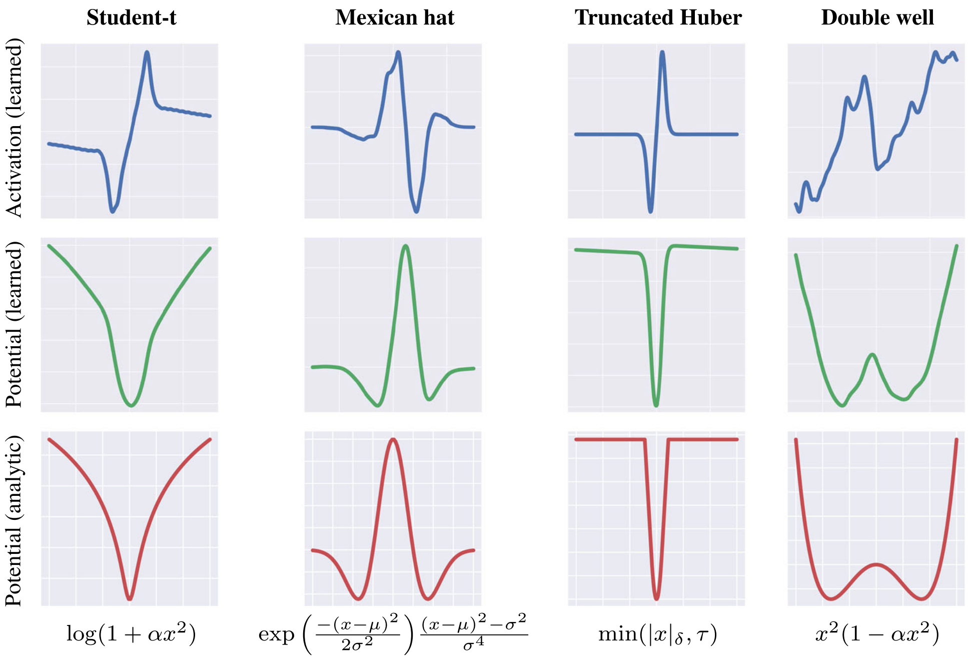


Fig. 10 Visualization of learned activation functions of our model. The first row (blue) shows the activation functions ρ in the derivative space. The second row (green) shows the corresponding potential functions ϕ in the energy domain. For comparison, the third row (red) shows analytic potential functions corresponding to the expressions shown below

linear regularizer and get

$$\nabla \mathcal{R}(\mathbf{u}) = \lambda \mathbf{u}, \tag{30}$$

which corresponds to a non-linear eigenvalue/eigenvector problem. Since we cannot compute a solution to (30) in closed form, we propose to compute approximate solutions by solving the nonlinear least squares optimization problem

$$\min_{\mathbf{u},\lambda} \frac{1}{2} \|\nabla \mathcal{R}(\mathbf{u}) - \lambda \mathbf{u}\|^2. \tag{31}$$

First, we observe that we can actually solve for λ^* in closed form by setting the derivative w.r.t. λ to zero. Thus, we get

$$\lambda^* = \frac{\langle \mathbf{u}, \nabla \mathcal{R}(\mathbf{u})\rangle}{\|\mathbf{u}\|^2}, \tag{32}$$

which is known as the Rayleigh quotient. Substituting (32) back into (30) yields the new optimization problem

$$\min_{\mathbf{u}} \frac{1}{2} \left\| \nabla \mathcal{R}(\mathbf{u}) - \frac{\langle \mathbf{u}, \nabla \mathcal{R}(\mathbf{u})\rangle \mathbf{u}}{\|\mathbf{u}\|^2} \right\|^2, \tag{33}$$

where we have additionally restricted the elements of the variable $\mathbf{u}$ to the interval [0, 1].

Now we are ready to move on to the eigenmode computation of the VN. Therefore, we solve problem (33) with Nesterov's proximal accelerated gradient method (Nesterov 1988) in order to actually compute the eigen disparity-maps. We iterate until convergence and get a pixel-wise residual of less than $2 \cdot 10^{-6}$, which indicates that we obtain high quality approximations for the eigenmaps. The computed eigenimages are of particular interest since substituting an eigenmode back into our model (5) yields

$$\begin{aligned} \mathbf{u}_{t+1} &= \mathrm{prox}_{\alpha_t \mathcal{D}}(\mathbf{u}_t - \alpha_t \overbrace{\nabla \mathcal{R}(\mathbf{u}_t)}^{\lambda \mathbf{u}_t}) \\ &= \mathrm{prox}_{\alpha_t \mathcal{D}}(\mathbf{u}_t(1 - \alpha_t \lambda)) \end{aligned} \tag{34}$$

and reveals that the regularizer adapts only the contrast to capture the correct disparity. Thus, the structure contained in the eigenimages is kept and transferred to the outputs of our model.

Figure 11 shows two examples of all three eigen maps, where we have used different initializations to compute different eigenmaps. It can be easily seen that our regularizer learned the stucture of local disparity maplets, i.e. local parts of natural disparity maps. We see that our regularizer prefers to accurately align the edges in all three components, the eigendisparity, eigenimage and eigenconfidence. They consist of e.g. pole-like and car-like structures as well as slanted surfaces.

Another way to interpret the eigen disparity maps is in terms of energy. Therefore, we observe that the Karush-Kuhn-Tucker condition of the constraint optimization problem

$$\min_{\mathbf{u}} \mathcal{R}(\mathbf{u}) \quad \text{s.t.} \ \frac{1}{2}\|\mathbf{u}\|_2^2 = \rho(\lambda), \tag{35}$$

for an unknown function ρ depending on the eigenvalue λ is given by

$$\nabla \mathcal{R}(\mathbf{u}) = \lambda \mathbf{u}, \tag{36}$$

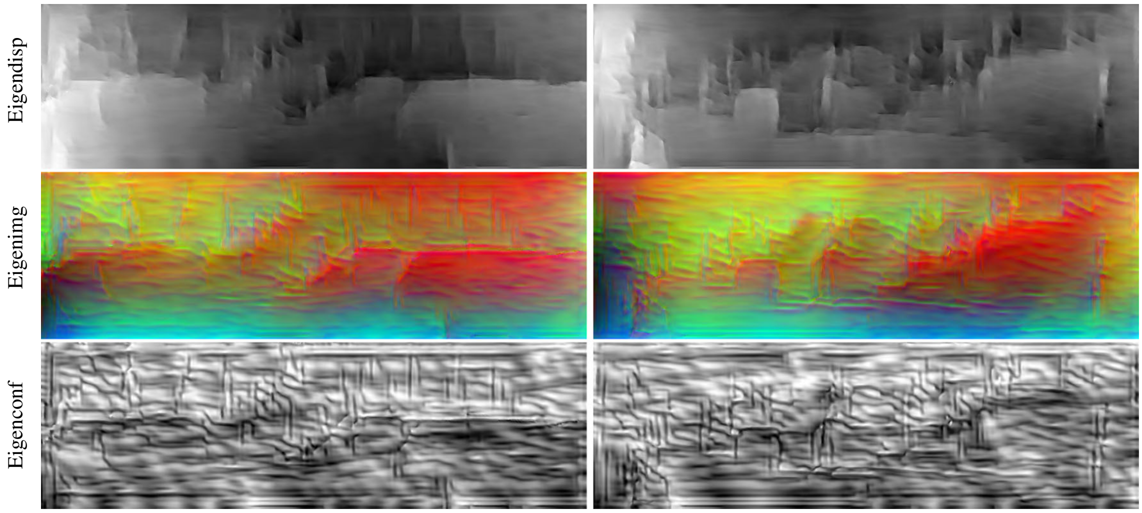


Fig. 11 Eigenimages. Top to bottom shows the eigenimage of our learned regularizer for disparity map, color image and confidence map. Note that the regularizer learned to favour pole-like structures, car parts and slanted surfaces. This fits perfectly to the scenery of the Kitti dataset. Best viewed in color on the screen

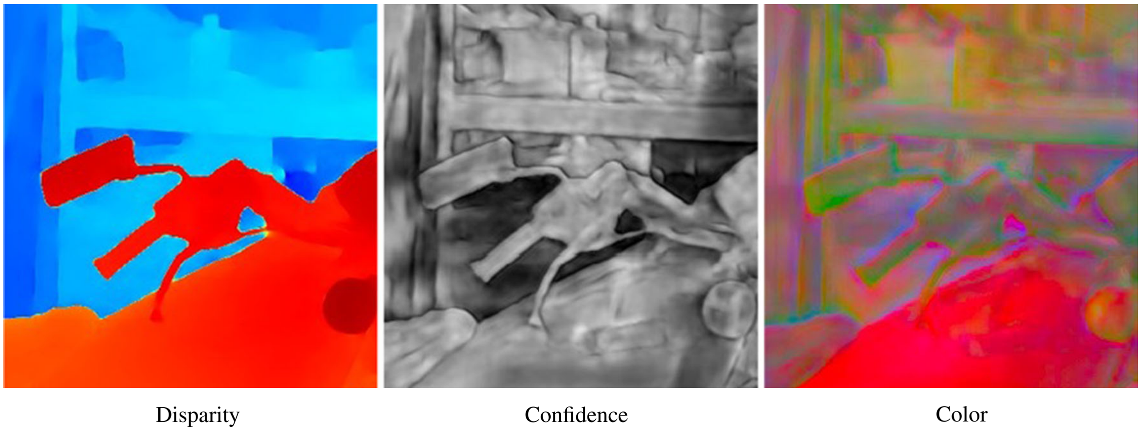


Fig. 12 Detailed view of the outputs of the VN. Note that all three images contain sharp edges at depth discontinuities. The color image is normalized for better visualization. Best viewed zoomed on the PC

which resembles exactly the non-linear eigenvalue problem defined in Equation 30. Thus, we are seeking images which contain structure, but have a low energy in the regularizer. The eigenmaps shown in Fig. 11 contain frequent structures of natural disparity maps and thus confirm that we have learned a regularizer suitable for disparity refinement.

7.4 VN Color Image

In this section we finally provide a possible interpretation of the processed color image. The color image is used to support the VN during refining the disparity map. As visualized in Fig. 4 it provides e.g. edges to guide the disparity refinement process. Figure 12 shows a detailed view of the three outputs of the VN, the refined disparity map, confidence image and color image. We first observe that both the confidence and the color image have edges at depth discontinuities. For the color image, we see on the one hand that the green channel captures depth discontinuities on the right side of object boundaries, where no occlusions exist. On the other hand, the blue channel seems to capture the left side of the object boundaries and can be interpreted as an occlusion detector. Thus, the color image shows a tendency to capture problem specific information in the processed color images. It is therefore used as a memory channel for the VN. Using the color image as an additional input during the refinement process yields not only better quantitative results as shown in Table 2, but contains also abstracted, but still interpretable information about the stereo problem.

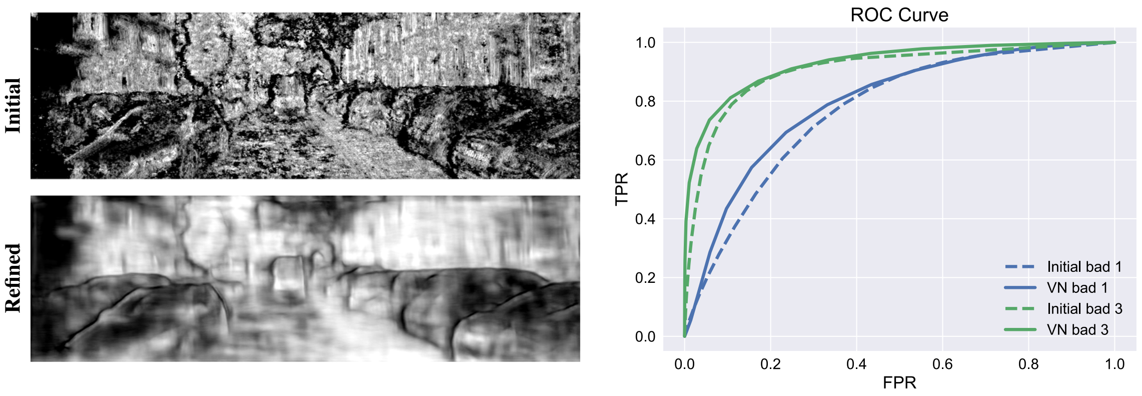


Fig. 13 Left: Initial confidences (top) and VN confidences (bottom). Right: ROC curve for initial confidences and VN confidences. The blue curve shows the confidences provided by the $\mathrm{VN}_4^{7,5}$ and the green curve shows the initial confidences. The larger the area under the curve, the better. Thus, the confidences provided by our VN reflect the actual performance better than the initial confidences. TPR = True positive rate, FPR = False positive rate

7.5 VN Confidences

In our collaborative refinement model we do not only refine the disparity map but we additionally implicitly refine the initial confidences as well. Note that we do not put any loss on the confidences during learning. We show in this section that the refined confidences are more reliable compared to the initial confidence values. Therefore, we compared the initial confidences from our feature network (e.g. Fig. 13, left top) with the confidences generated by the VN (e.g. Fig. 13, left bottom). For showing the reliability of the confidences we compute Receiver Operating Characteristics (ROC) for both confidence maps. Therefore, we compute the True Positive Rate[2] (TPR)

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{37}$$

where TP are the true positives and FN are the false negatives and the False Positive Rate (FPR)

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \tag{38}$$

where FP are the false positives and TN are the true negatives. To compute the all terms in Equation 37 and 38 we use a threshold δ to split all predicted disparities into two two sets $A \geq \delta$ for confident predictions and $B < \delta$ for vague predictions, respectively. Intuitively, the larger δ, the more reliable the predictions in set A should be. Thus, we define the TP and FP as the data points in A having a disparity error less than and larger than 1 or 3 pixels, respectively. Similarly, we use the set B to define the FN and TN as the data points in B having a disparity error less than and larger than 1 or 3 pixels, respectively. The ROC curve can then be computed by evaluating Equation 37 and 38 for a range of thresholds δ.

Figure 13 right shows the ROC curves of the initial confidences and the VN confidences. We construct the plot using the same data as we used in the ablation study. The larger the area under curve (AUC) in this plot, the better the confidences. The solid lines show the ROC curves for the VN and the dashed curves show the ROC curves of the initial confidences. We report the curves for the bad3 (green) and the bad1 (blue) error metric. Consider for example the point $(0.1, 0.82)$ on the green curve (bad3), where we have selected the FPR of 0.1. The ROC curve reveals here that the confidences predicted by our model are reliable with a TPR of 0.82, while the FPR is as low as 0.1. If we decrease δ, we get more points into the set A and have therefore the chance to get a higher TPR, but we will also increase the FPR as can be seen in Fig. 13. The ROC curve of the confidences of the VN are always above the curve of the initial confidences which yields a higher AUC for the VN. Thus, we have shown that the refined confidences are more reliable than the initial confidences.

8 Conclusion & Future Work

We have proposed a learnable variational network for efficient refinement of disparity maps. The learned collaborative and hierarchical refinement method allows the use of information from the joint color, confidence and disparity space from multiple spatial resolutions. In an ablation study, we evaluated a broad range of architectural choices and demonstrated the impact of our design decisions. Our method can be applied on top of any other stereo method and explicitly exploits confidence information contained in a cost volume.

[2] Also known as *Recall*

We demonstrated this by adding the variational refinement network on top of the CNN-CRF method and have shown improved results. We have shown insights and interpretations of our model in terms of visualizing the indermediate steps, the learned filters and activation functions. The optimization like structure of our model additionally allowed us to compute eigen disparity maps, eigen color images and eigen confidence maps. Furthermore, we have proven the effectiveness of our method by participating in the publicly available stereo benchmarks of Middlebury and Kitti. In future work, we would like to include a matching score during the refinement process and perform data augmentation to increase the training set for learning.

Funding Open access funding provided by Graz University of Technology.

References

Barron, J. T., & Poole, B.(2016). The fast bilateral solver. In *European conference on computer vision (ECCV)* (pp. 617–632).

Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal of Imaging and Sciences* pp. 183–202.

Brox, T., Bruhn, A., Papenberg, N., & Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision (ECCV)* (pp. 25–36).

Chambolle, A., & Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision* 120–145.

Chang, J. R., & Chen, Y. S. (2018). Pyramid stereo matching network. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5410–5418).

Chen, Y., Yu, W., & Pock, T.(2015). On learning optimized reaction diffusion processes for effective image restoration. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5261–5269).

Effland, A., Kobler, E., Kunisch, K., & Pock, T. (2020). An optimal control approach to early stopping variational methods for image restoration. *Journal of Mathematical Imaging and Vision* 396–416.

Gidaris, S., & Komodakis, N. (2017). Detect, replace, refine: Deep structured prediction for pixel wise labeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5248–5257).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778).

Hu, X., & Mordohai, P. (2012). A quantitative evaluation of confidence measures for stereo vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2121–2133.

Khamis, S., Fanello, S., Rhemann, C., Kowdle, A., Valentin, J., & Izadi, S. (2018). Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *European conference on computer vision (ECCV)* (pp. 8–14).

Kingma, D. P., & Ba, J.(2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Knöbelreiter, P., & Pock, T. (2019). Learned collaborative stereo refinement. In *German conference on pattern recognition (GCPR)* (pp. 3–17).

Knöbelreiter, P., Reinbacher, C., Shekhovtsov, A., & Pock, T. (2017). End-to-end training of hybrid CNN-CRF models for stereo. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2339–2348).

Kobler, E., Klatzer, T., Hammernik, K., & Pock, T.(2017). Variational networks: Connecting variational methods and deep learning. In *German conference on pattern recognition (GCPR)* (pp. 281–293).

Kuschk, G., & Cremers, D. (2013). Fast and accurate large-scale stereo reconstruction using variational methods. In *IEEE international conference on computer vision workshop* (pp. 700–707).

Liang, Z., Feng, Y., Guo, Y., Liu, H., Chen, W., Qiao, L., Zhou, L., & Zhang, J. (2018). Learning for disparity estimation through feature constancy. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2811–2820).

Long, J., Shelhamer, E., & Darrell, T.(2015). Fully convolutional networks for semantic segmentation. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3431–3440).

Maurer, D., Stoll, M., & Bruhn, A.(2017). Order-adaptive and illumination-aware variational optical flow refinement. In *British machine vision conference*.

Meinhardt, T., Moeller, M., Hazirbas, C., & Cremers, D.(2017). Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *IEEE International conference on computer vision (ICCV)*.

Menze, M., & Geiger, A. (2015). Object scene flow for autonomous vehicles. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3061–3070).

Nesterov, Y. (1988). On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ekonomika i Mateaticheskie Metody*, *24*(3), 509–517.

Pang, J., Sun, W., Ren, J. S., Yang, C., & Yan, Q. (2017). Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *IEEE international conference on computer vision workshop* (pp. 887–895).

Parikh, N., & Boyd, S. (2014). Proximal algorithms. Foundations and trends® in Optimization pp. 127–239.

Ranftl, R., Bredies, K., & Pock, T. (2014). Non-local total generalized variation for optical flow estimation. In *European conference on computer vision (ECCV)* (pp. 439–454).

Ranftl, R., Gehrig, S., Pock, T., & Bischof, H. (2012). Pushing the limits of stereo using variational stereo estimation. In *IEEE intelligent vehicles symposium* (pp. 401–407).

Revaud, J., Weinzaepfel, P., Harchaoui, Z., & Schmid, C. (2015). Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1164–1172).

Riegler, G., Rüther, M., & Bischof, H. (2016). ATGV-Net: Accurate depth super-resolution. In *European conference on computer vision (ECCV)* (pp. 268–284).

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In International conference on medical image computing and computer-assisted intervention (MICCAI) (pp. 234–241).

Roth, S., & Black, M. J. (2009). Fields of experts. *International Journal of Computer Vision*, *82*(2), 205.

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

Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, *47*(1–3), 7–42.

Shekhovtsov, A., Reinbacher, C., Graber, G., & Pock, T.(2016). Solving dense image matching in real-time using discrete-continuous optimization. *Computer vision winter workshop*.

Tulyakov, S., Ivanov, A., & Fleuret, F. (2018). Practical deep stereo (PDS): Toward applications-friendly deep stereo matching. In *Proceedings of advances in neural information processing systems* (pp. 5871–5881).

Vogel, C., Knöbelreiter, P., & Pock, T. (2018). Learning energy based inpainting for optical flow. In *Asian conference on computer vision (ACCV)* (pp. 340–356).

Vogel, C., & Pock, T.(2017). A primal dual network for low-level vision problems. In *German conference on pattern recognition (GCPR)* (pp. 189–202).

Wang, S., Fidler, S., & Urtasun, R. (2016). Proximal deep structured models. In *Proceedings of advances in neural information processing systems* (pp. 865–873).

Zach, C., Pock, T., & Bischof, H.(2007). A duality based approach for realtime TV-L1 optical flow. In *German conference on pattern recognition (GCPR)* (pp. 214–223).

Žbontar, J., & LeCun, Y. (2016). Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, *17*(1), 2287–2318.

Zhu, S. C., Wu, Y., & Mumford, D. (1998). Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, *27*(2), 107–126.