ATGV-Net: Accurate Depth Super-Resolution

Gernot Riegler, Matthias Rüther, Horst Bischof

Institute for Computer Graphics and Vision,
Graz University of Technology
{riegler, ruether, bischof}@icg.tugraz.at

Abstract. In this work we present a novel approach for single depth map super-resolution. Modern consumer depth sensors, especially Time-of-Flight sensors, produce dense depth measurements, but are affected by noise and have a low lateral resolution. We propose a method that combines the benefits of recent advances in machine learning based single image super-resolution, *i.e.* deep convolutional networks, with a variational method to recover accurate high-resolution depth maps. In particular, we integrate a variational method that models the piecewise affine structures apparent in depth data via an anisotropic total generalized variation regularization term on top of a deep network. We call our method *ATGV-Net* and train it end-to-end by unrolling the optimization procedure of the variational method. To train deep networks, a large corpus of training data with accurate ground-truth is required. We demonstrate that it is feasible to train our method solely on synthetic data that we generate in large quantities for this task. Our evaluations show that we achieve state-of-the-art results on three different benchmarks, as well as on a challenging Time-of-Flight dataset, all without utilizing an additional intensity image as guidance.

Keywords: deep networks, variational methods, depth super-resolution

1 Introduction

Over the last decade depth sensors have entered the mass market which substantially improved in package size, energy consumption and price. This made depth data an interesting and important auxiliary input for computer vision tasks, for example in pose estimation [14, 35], or scene understanding [16]. However, current sensors are limited by physical and manufacturing constraints. Hence, depth outputs are affected by degenerations due to noise, quantization and missing values, and typically have a low resolution.

To alleviate the use of depth data, recent methods focus on increasing the spatial resolution of the acquired depth maps. A common approach to tackle this problem is to utilize a high-resolution intensity image as guidance [12, 25, 29]. These methods are motivated by the statistical co-occurrences of edges in intensity images and discontinuities in depth. In practical scenarios, however, a depth sensor is not always accompanied by an additional camera and the depth map has to be projected to the guidance image, which is also problematic due to noisy depth measurements. Therefore, approaches that solely rely on the depth input for super-resolution are becoming popular [1, 13, 20].

In contrast to super-resolution methods for depth data, machine learning based methods for natural images [11, 33, 36, 37] are advancing rapidly and achieve impressive results on standard benchmarks. Those methods learn a mapping from a low-resolution input space to a plausible and visually pleasing high-resolution output space. The inference is performed for small, overlapping patches of the image independently, and are then averaged for the final output. This is not optimal for depth data, as it is characterised by textureless, piece-wise affine regions that have sharp depth discontinuities. In contrast, variational methods are especially suited for this task, because the aforementioned prior information can be exploited in the model's regularization term. A prominent example is the total generalized variation (TGV) [3] that is for example utilized in [12].

In this work we propose a method that combines the advantages of data-driven methods and energy minimization models by combining a deep convolutional network with a powerful variational model to compute an accurate high-resolution output from a single low-resolution depth map input. Deep networks recently demonstrated impressive capabilities in single-image super resolution [22]. We utilize a similar architecture for our network, but instead of just producing the refined depth map as output, we design the network to additionally predict the locations of the depth discontinuities in the high-resolution output space. Both outputs are then used as input for a variational model to refine the high-resolution estimate. The variational model uses an anisotropic TGV pairwise regularization that is weighted by the network output. To integrate the variational method into our network and learn the joint model end-to-end, we unroll all computation steps of the primal-dual optimization scheme [5] that is used for inference with layers of a deep network. Therefore, we name our method *ATGV-Net*. Finally, we deal with the problem of obtaining accurate ground-truth data for training. The training of deep networks requires a large corpus of data. We demonstrate that we can train our model entirely on synthetic depth data that we generate in large quantities and obtain state-of-the-art results on four different benchmark datasets.

Our contributions can be summarized as follows: (i) We integrate a variational model with anisotropic TGV regularization into a deep network by unrolling the optimization steps of the primal-dual algorithm [5] and train the whole model end-to-end (see Sec. 3). (ii) We demonstrate that our joint model can be trained entirely on synthetic data for single depth map super-resolution (see Sec. 4.1). (iii) Finally, we show that our method improves upon state-of-the-art results on four different benchmark datasets (see Sec. 4.2-4.4).

2 Related Work

Depth Super-Resolution In general, the work on super-resolution is roughly divided in approaches that use a series of aligned images to produce a high-resolution output, and single image super-resolution, *i.e.* approaches that use only one low-resolution image as input. We focus in this related work on the latter as our method falls into this category.

Natural images often contain repetitive structures and therefore, a patch might be visible on different scales within the same image. Glasner *et al.* [15] exploit this knowledge in their seminal work. For each image patch they search similar patches across various scales in the image and combine them for a high-resolution estimate. A similar idea is employed for depth data by Hornáček *et al.* [20], but instead of reasoning about 2D patches, they reason in terms of patches containing 3D points. The 3D points of the depth map patch can be translated and rotated with six degree of freedom to find related patches within the same depth map. Aodha *et al.* [1] search for similar patches not within the same image, but in an ancillary database and they formulate a Markov Random Field (MRF) that enforces smooth transition between the candidate high-resolution patches.

More recently, machine learning approaches have become popular for single image super-resolution. They achieve higher accuracy and are at the same time more efficient in testing, because they do not rely on a computational intensive patch search. Sparse coding approaches [40, 42] learn dictionaries for the low- and high-resolution domains that are coupled via a common encoding. To increase the inference speed, Timofte *et al.* [36] replace the ℓ_1 norm in the sparse coding step with the ℓ_2 norm, which can be solved in closed form and replace a single dictionary by man smaller sub-dictionaries to improve accuracy. In [33], Schulter *et al.* substitute the flat code-book of sparse coding methods with a random regression forest. A test patch traverses the trees of the forest and each leaf node stores regression coefficients to predict a high-resolution estimate. Deep learning based approaches recently showed very good results for single image super-resolution, too. Dong *et al.* [11] train a convolutional network of three layers. The input to the network is the bilinear upsampled low-resolution image and the network is trained with the Euclidean loss on the network output and the corresponding ground-truth high-resolution image. This idea was substantially improved by Kim *et al.* [22]. They train a deep network with up to 20 convolutional layers with filters of size 3×3 and therefore, increasing the receptive field to 41×41 pixel from 15×15 pixels of the network in [11]. Further, the network does not output directly the high-resolution estimate, but the residual to the pre-processed input image, aiding training of the very deep networks [19].

These learning based methods have mainly been applied to color images, where a huge amount of training data can be easily obtained. In contrast, large datasets with dense, accurate depth maps have only very recently become available, *e.g.* [17]. Therefore, most methods for depth map super-resolution are not based on machine learning, but utilize a high-resolution intensity image as guidance. One of the first works in this direction is by Diebel and Thrun [9]. They apply a MRF for the upsampling task and weight their smoothness term according to the gradients of the guidance image. Yang *et al.* [41] propose an approach based on a bilateral filter that is iteratively applied to estimate a high-resolution output map. Park *et al.* [29] present a least-squares method, that incorporates edge aware weighting schemes in the regularization term of their formulation. A more recent approach of Ferstl *et al.* [12] utilizes a variational framework for image guided depth upsampling, where they also use the total generalized variation [3] as regularization term. One of the few machine learning based approaches for depth map super-resolution is by Kwon *et al.* [25]. They collect their own training data using KinectFusion [21] and facilitate sparse coding with an additional multi-scale approach and an advanced edge weighting term, that emphasizes intensity edges corresponding to depth discontinuities. Ferstl *et al.* [13] use sparse coding with dictionaries trained on the 31 synthetic depth maps of [1] to predict the depth discontinuities in the high-resolution domain from the low-resolution depth data. Those edge estimates are then used in an anisotropic diffusion tensor of their regularization term.

Deep Network Integration of Energy Minimization Methods Energy minimization methods, such as Markov Random Fields (MRFs), or variational methods have a wide range of applications in computer vision. They consist of unary terms, for example the class likelihood of a pixel for semantic segmentation, or the depth value in depth super-resolution, and pairwise terms, which measure the dependencies on neighbouring pixels. Recently, the integration of those models into deep networks gained a lot of attention, as deep networks jointly trained with energy minimization methods achieve excellent results. For example, Tompson *et al.* [38] propose the joint training of a convolutional network and a MRF for human pose estimation. The MRF is realized by very large convolutional filters to model the pairwise interactions between joints and can be interpreted as one iteration of loopy belief propagation. In [8, 34] the authors show how to compute the derivative with respect to the mean field approximation [24] in MRFs. This allows end-to-end learning and improves results for instance in semantic segmentation. Similarly, Zheng *et al.* [43] show that the computation steps of the mean field approximation can be modeled by operations of a convolutional network and unroll the iterations on top of their network.

While the latter approaches for semantic segmentation are designed for a discrete label space, the variational approach by Ranftl and Pock [31] has a continuous output space. They show that the gradient of a loss function can be back-propagated through the energy functional of a variational method by implicit differentiation, if the functional is smooth enough. This approach has been extended for depth denoising and upsampling by Riegler *et al.* [32]. Recently, Ochs *et al.* [28] propose a technique that allows the back-propagation through non-smooth energy functionals using Bregman proximity functions [6], but did not demonstrate the use in combination with deep networks.

Our approach utilizes a variational method on top of a deep network, but instead of implicitly differentiating the energy functional as in [31, 32] we unroll every step of an exact optimization scheme [5], in the spirit of [10]. This has two major advantages: First, we can incorporate stronger pairwise regularization terms and second, the optimization gets more robust, allowing the successful training of deeper networks. This is similar to [43], but instead of the mean field approximation, we unroll the steps of the primal-dual algorithm by

Chambolle and Pock [5], which converges to the global optimal solution of the convex energy functional. For parametrizing the variational method we use a 10 layer deep network of 3×3 convolutions, and train on the residual similarly to [22]. Additionally, we train the network to predict the depth discontinuities in the high-resolution output space. This output is used to weight the pairwise regularization term of the variational part. Finally, we demonstrate that we can train a deep network for this task by rendering synthetic depth maps in large quantities with a ray-caster running on the GPU.

3 ATGV-Net

In this section we describe our method that takes a single low-resolution, probably noisy depth map as input and computes a high-resolution output. We first introduce the notation used throughout this work and then detail our variational model, how we integrate it on top of a deep network and finally the network itself.

In the remainder of this work we denote the low-resolution depth map input as $s_k^{(\mathrm{lr})} \in \mathbb{R}^{M \times N}$. Further, for training we assume that we have for each input sample an accurate, high-resolution ground-truth depth map $t_k \in \mathbb{R}^{\rho M \times \rho N}$, where $\rho > 1$ is the given upsampling factor. The only preprocessing step in our method is a bilinear upsampling of the low-resolution input depth map $s_k^{(\mathrm{lr})}$ to the size of the ground-truth target depth map. We denote this mid-level representation of the input as $s_k \in \mathbb{R}^{\rho M \times \rho N}$.

Given a training set $\{(s_k, t_k)\}_{k=1}^K$ of K training pairs we follow [31, 32] and formulate the training task as the following bi-level optimization problem:

$$\min_w \frac{1}{K} \sum_{k=1}^{K} L(u^*(f(w, s_k)), t_k) \tag{HL}$$

$$\text{s.t.} \quad u^*(f(w, s_k)) = \arg\min_u E(u; f(w, s_k)). \tag{LL}$$

This optimization problem has an intuitive interpretation: In the higher-level problem (HL) we want to minimize some weights w, such that the minimizer u^* of the energy functional E in the lower-level problem (LL), which is parameterized by a learnable function f, achieves a low loss L over all training samples. We provide more details on the energy functional and on the parametrization in Sec. 3.1 and Sec. 3.2, respectively. For the loss we only impose the restriction that we can compute the gradient with respect to u^*. For the remainder of this work we will use the Euclidean loss:

$$L(u^*(f(w, s_k)), t_k) = \|u^*(f(w, s_k)) - t_k\|_2^2 \,. \tag{1}$$

The authors of [31, 32] have proven that the bi-level optimization problem can be solved by implicit differentiation, if certain assumptions for the energy functional E hold. Namely, E has to be strongly convex, twice differentiable with respect to u and once differentiable with respect to f. Further, the gradient of f has to

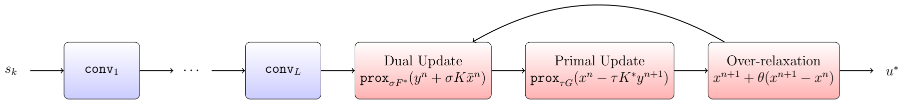

s_k
conv$_1$
$\cdots$
conv$_L$
Dual Update
$\texttt{prox}_{\sigma F^*}(y^n + \sigma K \bar{x}^n)$
Primal Update
$\texttt{prox}_{\tau G}(x^n - \tau K^* y^{n+1})$
Over-relaxation
$x^{n+1} + \theta(x^{n+1} - x^n)$
u^*

Fig. 1. Our model consists of a deep convolutional network with $L = 10$ layers (blue rectangles) that predicts a first high-resolution depth map and depth discontinuities. The output of the network is then feed to an unrolled primal-dual optimization algorithm (red rectangles) realized by operations in a deep network that further refines the result. This enables us to train the joint model end-to-end.

be computable with respect to w. The last constraint is satisfied by construction since the parametrization f is realized by a deep network. However, the first constraints drastically limit the choice of energy functionals and therefore, the authors of [31, 32] had to design smooth approximations. In the following we show that this constraints can be eliminated by unrolling the optimization steps of the lower-level problem (LL) on top of a deep network, similar to [43].

3.1 Unrolling the Optimization

For the energy functional we have the requirement that it should refine the initial high-resolution depth estimate. Therefore, we use a TGV$_2$-ℓ_2 variational model [3] that favors the piecewise affine surfaces apparent in depth maps. In addition, we incorporate an anisotropic diffusion tensor [30, 39] into the regularization and name our model *ATGV-Net*. The optimization of the energy functional in conjunction with a guidance intensity image already provides good results for depth super-resolution [13]. In the following we demonstrate, how we can significantly improve the model by parametrizing the energy functional by a deep network and learn it end-to-end by unrolling the optimization procedure.

In general, our energy functional consists of a pairwise regularization term R and an ℓ_2 data term:

$$E(u; f(w, s_k)) = R(u, h(w_h, s_k)) + \frac{e^{w_\lambda}}{2} \left\| u - g(w_g, s_k) \right\|_2^2 . \tag{2}$$

The functional is parameterized by a function $f(w, s_k) = [h(w_h, s_k), w_\lambda, g(w_g, s_k)]^T$ that has learnable weights w and takes the mid-resolution depth map s_k as input. The functions h and g are realized as a single deep network and described in Sec. 3.2. The parameter w_λ controls the trade-off between data and regularization term and is also learned. We take the exponential of w_λ to ensure convexity of the energy functional. For the pairwise regularization term we utilize the total generalized variation (TGV) [3] of second order that favors piecewise affine solutions and is therefore ideal for depth maps:

$$R(u, h(w_h, s_k)) = \min_v \alpha_1 \left\| T(h(w_h, s_k))(\nabla_u u - v) \right\|_1 + \alpha_0 \left\| \nabla_v v \right\|_1 , \tag{3}$$

where α_0 and α_1 are user defined parameters. In the regularization term, an anisotropic diffusion tensor T enforces a low degree of smoothness across depth discontinuities and vice versa, more smoothness in homogeneous regions. This anisotropic diffusion tensor is based on the Nagel-Enkelmann operator [27]:

$$T(h(w_h, s_k)) = \exp(-\beta \left\| h(w_h, s_k) \right\|_2^\gamma) nn^T + n_\perp n_\perp^T \,, \tag{4}$$

with β and γ being adjustable parameters weighting the magnitude and sharpness of the tensor. The gradient normal of h is given by

$$n = \frac{h(w_h, s_k)}{\left\| h(w_h, s_k) \right\|_2} \,, \quad n_\perp \cdot n = 0 \,. \tag{5}$$

To optimize this energy functional we chose the first-order primal-dual algorithm by Chambolle and Pock [5], as it guarantees fast convergence. To apply the optimization algorithm, we first reformulate Eq. (2) as saddle-point problem with dual variables p, q as

$$\min_{u,v} \max_{p,q} \alpha_1 \left\langle T(h(w_h, s_k))(\nabla_u u - v), p \right\rangle + \alpha_0 \left\langle \nabla_v v, q \right\rangle + \frac{e^{w_\lambda}}{2} \left\| u - g(w_g, s_k) \right\|_2^2 \tag{6}$$

$$\text{s.t.}\, p \in \left\{ p \in \mathbb{R}^{2 \times \rho M \times \rho N} \mid \left\| p_{:,i,j} \right\|_2 \leq 1 \right\}, q \in \left\{ q \in \mathbb{R}^{4 \times \rho M \times \rho N} \mid \left\| q_{:,i,j} \right\|_2 \leq 1 \right\} \,, \tag{7}$$

where ∇_u and ∇_v denote operators in the discrete setting that compute the forward differences of u and v. A single iteration of the optimization procedure to obtain u^* is then given by:

$$\begin{aligned}
p^{n+1} &= \text{proj}(p^n + \sigma_p \alpha_1 (T(h(w_h, s_k))(\nabla_u \bar{u}^n - \bar{v}^n))) && (8)\\
q^{n+1} &= \text{proj}(q^n + \sigma_q \alpha_0 \nabla_v \bar{v}^n) && (9)\\
u^{n+1} &= \frac{u^n + \tau_u (\alpha_1 \nabla_u^T T(h(w_h, s_k)) p^{n+1} + e^{w_\lambda} g(w_g, s_k))}{1 + \tau_u e^{w_\lambda}} && (10)\\
v^{n+1} &= v^n + \tau_v (\alpha_0 \nabla_v^T q^{n+1} + \alpha_1 T(h(w_h, s_k)) p^{n+1}) && (11)\\
\bar{u}^{n+1} &= u^{n+1} + \theta(u^{n+1} - u^n) && (12)\\
\bar{v}^{n+1} &= v^{n+1} + \theta(v^{n+1} - v^n) \,, && (13)
\end{aligned}$$

with $u^0 = g(w_g, s_k)$, $v^0, p^0, q^0 = 0$, $\sigma_p, \sigma_q, \tau_u, \tau_v > 0$, $\theta \in [0, 1]$, and $\text{proj}(p) = \frac{p}{\max(1, \|p\|_2)}$ is the point-wise projection to the unit hyper-sphere:

The key observations are: (i) The single computation steps in this optimization algorithm can be realized by operations of a deep network, *i.e.* individual network layers, and (ii) given a fixed number of iterations, the algorithm can be unrolled like a recurrent neural network, similar to [43]. This allows us to use the back-propagation algorithm to train the optimization procedure, *i.e.* all hyper-parameters, jointly with the parametrization, *i.e.* the deep network. See Figure 1 for a visualization of the concept. In the following we detail how the individual computation steps are realised within our model. We provide a graphical representation of a single iteration of the optimization procedure in terms of deep network operations in the supplemental material.

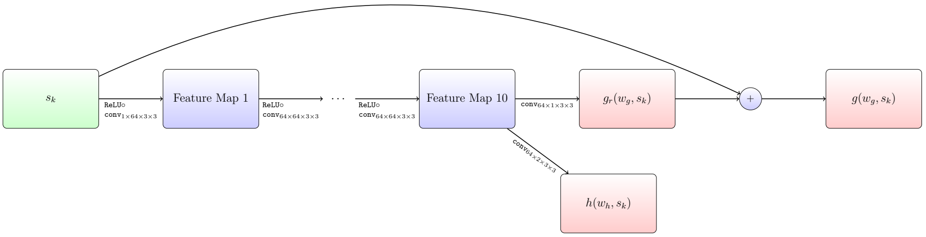

s_k
ReLU∘ conv$_{1\times64\times3\times3}$
Feature Map 1
ReLU∘ conv$_{64\times64\times3\times3}$
$\cdots$
ReLU∘ conv$_{64\times64\times3\times3}$
Feature Map 10
conv$_{64\times1\times3\times3}$
$g_r(w_g, s_k)$
+
$g(w_g, s_k)$
conv$_{64\times2\times3\times3}$
$h(w_h, s_k)$

Fig. 2. Overview of our deep network architecture. Our network consists of 10 convolutional layers with 3×3 filters and 64 feature maps in the hidden layers (blue rectangles). The input to the network (green rectangle) is the mid-resolution depth map and the output is (i) the residual that after adding to the mid-resolution input produces the high-resolution estimate $g(w_g, s_k)$ and (ii) the estimates of the depth discontinuities in the high-resolution output $h(w_h, s_k)$ (red rectangles).

Dual Update The gradient ascent of the dual variables in Eq. (8) and (9) consists of scalar multiplication, point-wise addition and multiplication, the gradient operators ∇_u, ∇_v, and the projection. The scalar multiplication and the point-wise operations are trivial operations and are implemented in most deep learning frameworks. The ∇-operator is basically a convolution with two filters, $\nabla_x = [-1, 1]$ and $\nabla_y = [-1, 1]^T$. Therefore, it can be implemented with a standard convolutional layer that has fixed filter coefficients. Additionally, we have to ensure a reflecting padding of the layer input, *i.e.* Neumann boundary conditions. Finally, the proj-operator is a composition of a point-wise division, a max-operator and the ℓ_2 norm. We implemented the max-operator as shifted ReLU, and the ℓ_2 norm as custom layer.

Primal Update The gradient descent of the primal variables in Eq. (10) and (11) consists of similar operations as the dual update, and therefore, can be implemented with the same building blocks. Additional operators are ∇_u^T, ∇_v^T. These operators are defined as $\nabla^T p = \nabla_x p_x + \nabla_y p_y$. From this definition we can see that this operation can again be implemented with a convolutional layer that has fixed filter coefficients. However, we have to ensure a negative symmetric padding of the layer input, *i.e.* Dirichlet boundary conditions.

Over-Relaxation The over-relaxation step of the primal variables in Eq. (12) and Eq. (13) can be simplified to a weighted sum of two terms, *i.e.* $\bar{u} = (1 + \theta)u^{n+1} - \theta u^n$ and $\bar{v} = (1 + \theta)v^{n+1} - \theta v^n$.

3.2 Parametrization

After we have described the variational model and how to integrate it on top of a deep network, we now detail the parametrization functions $h(w_h, s_k)$ and $g(w_g, s_k)$. Inspired by the recent success in single image super-resolution for color images [22], we implement $g(w_g, s_k)$ as a deep convolutional neural network with 10 convolutional layers. Each convolutional filter has the size of 3×3 and each hidden layer of the network has 64 feature maps. As $g(w_g, s_k)$ is used in the data term of our energy functional it should provide a good initial estimate of the high-resolution depth map. However, the output of this network is not the estimate of the high-resolution depth map itself, but the residual $g_r(w_g, s_k)$, such that $g(w_g, s_k) = g_r(w_g, s_k) + s_k$. Learning the residual instead of the full output aids the training procedure of the network [22], and has been applied before in other super-resolution methods [33, 36, 37].

The parameterization function $h(w_h, s_k)$ is used for weighting the pairwise regularization term. As we argued before, the regularization should be small near depth discontinuities and high in smooth areas. Therefore, we implemented $h(w_h, s_k)$ as an additional network output of size $2 \times \rho M \times \rho N$ and train it to estimate the gradient of the high-resolution target ∇t_k. This method has two benefits: First, we get more accurate estimates for the depth discontinuities than what we would get from the gradient of the high-resolution estimate $g(w_g, s_k)$. Secondly, the joint training of both objectives in a single deep network improves the performance of both tasks, because the weights w_h and w_g share the majority of parameters and only the parameters of the last layer, the output, differ. A graphical depiction of our deep network parametrization is shown in Figure 2.

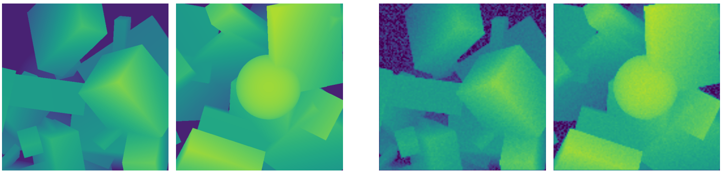
3.3 Training

In the previous sections we presented the description of our model. In this section we detail how we train it given a large set of training samples $\{(s_k, t_k)\}_{k=1}^{K}$. The training procedure is two-fold: In a first step we initialize the deep convolutional network, *i.e.* the functions g and h. Therefore, we train the network by mini-batch gradient descent with momentum term on the following loss function:

$$L_p(\{(s_k, t_k)\}_{k=1}^{K}) = \frac{1}{K} \sum_{k=1}^{K} \|g(w_g, s_k) - t_k\|_2^2 + \|h(w_h, s_k) - \nabla t_k\|_2^2 \,. \qquad (14)$$

In the following evaluations we set the learning rate to 0.001 and the momentum parameter to 0.9 for the initializing of the network. With this setting we train the network for 30 epochs on non-overlapping patches of size 32 × 32 pixel.

In the second step of the training procedure we add the unrolled primal-dual optimization algorithm as introduced in Sec. 3.1 on top of the network. Then, we train the joint model end-to-end on the Euclidean loss stated in Eq. (1) with mini-batch gradient descent. We set the learning rate to 0.001 and the momentum parameter 0.9 to train the whole model for 5 epochs on non-overlapping patches of size 128 × 128 pixel. In contrast to the method of implicit differentiation [31, 32], our method is still robust if we use a high learning rate, and as a consequence converges in fewer training iterations. Further, it enables us to optimize the parameter w_λ, as well ass all hyper-parameters of the optimization procedure.



(a) rendered high-resolution ground-truth (b) corresponding mid-resolution input

Fig. 3. Examples of our generated depth maps. (a) visualizes the high-resolution ground-truth data. By resampling those depth maps with a scale factor ρ and adding depth dependent noise we create the low-resolution input. (b) shows the mid-resolution input, which is the bilinear upsampled low-resolution data. Best viewed magnified in the electronic version.

4 Evaluation

In this section we present an exhaustive experimental evaluation of the proposed *ATGV-Net*. First, we show how we generate a huge amount of training data with accurate ground-truth needed to train the deep network. Then, we demonstrate evaluation results on four standard benchmark datasets for depth map super-resolution: Following [1, 13, 20], we evaluate our method on the noise-free Middlebury disparity maps *Teddy*, *Cones*, *Tsukuba* and *Venus*. Additionally, we show results for the Laserscan dataset as proposed in [1]. In a second evaluation we compare our results on the noisy Middlebury 2007 dataset as proposed in [29] and finally, we demonstrate the real-world applicability of our method on the challenging ToFMark dataset [12].

We set the initial parameters of our model to $\alpha_1 = 17$, $\alpha_0 = 1.2$ for the regularization term, $\beta = 9$, $\gamma = 0.85$ for the anisotropic diffusion tensor, and $w_\lambda = 0.01$ for all experiments. Further, we fix the number of iterations of the primal-dual algorithm to 10.

4.1 Training Data

One challenge in training very deep networks is the need for a huge amount of training data. In [1, 13] the authors use a small set, *i.e.* 31 depth maps, of synthetic rendered images for training and in [32] the authors trained and tested their method on the synthetic New Tsukuba dataset [26]. Only very recently larger datasets with accurate depth maps have been released [17], or have been added to existing benchmarks [4]. In our method we also make use of synthetically rendered data, but produce them in a much larger quantity.

For this purpose we implemented a ray-caster [2] that runs on the GPU and enables us to generate thousands of synthetic depth maps of high quality in a few minutes. For each image we randomly place between 24 and 42 rectangular cuboids and up to 3 spheres in a predefined volume. Further, we randomly scale and rotate each solid to achieve an infinitely number of possible constellations. Then, we place a virtual camera at the origin of the coordinate system and cast

Table 1. Results on the noise-free Middlebury and Laserscan data. We report the error as root mean squared error (RMSE) in pixel disparity for the Middlebury data and in *mm* for the Laserscan data, respectively. We highlight the best result in boldface and the second best in italic.

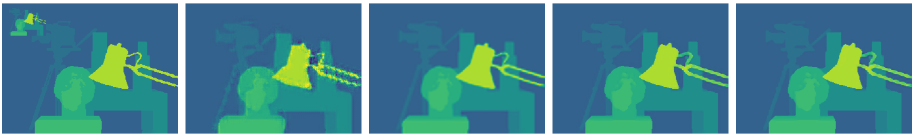
	×2				×4				×4		
	Cones	Teddy	Tsukuba	Venus	Cones	Teddy	Tsukuba	Venus	Scan21	Scan30	Scan42
NN	4.3772	3.2596	9.7968	2.1408	6.1236	4.5168	13.3248	2.9432	0.0177	0.0163	0.0396
Bicubic	3.8392	2.7668	8.3648	1.8192	4.9544	3.5744	10.6960	2.3504	0.0132	0.0125	0.0326
Diebel & Thrun [9]	2.9588	2.1060	6.4208	1.3624	4.5624	3.2040	8.7840	1.9408	–	–	–
Ferstl *et al.* [12]	2.8240	2.1408	7.0592	1.2840	3.6372	2.5068	10.0128	1.4624	–	–	–
Zeyde *et al.* [42]	2.7680	1.9616	6.1936	1.3200	3.8468	2.7812	8.7632	1.7592	0.0100	0.0093	0.0246
Timofte *et al.* [36]	2.7872	1.9816	6.1280	1.3328	3.0256	3.0256	9.6304	1.9616	0.0106	0.0101	0.0264
Aodha *et al.* [1]	4.5076	3.2988	9.6192	2.2088	6.0168	4.1036	13.3328	2.6920	0.0175	0.0170	0.0452
Hornáček *et al.* [20]	3.9744	3.1640	9.2832	2.0592	5.5944	4.7828	11.6352	3.6008	0.0205	0.0179	0.0299
Ferstl *et al.* [13]	2.4988	1.7588	5.6064	1.1464	3.7336	2.6680	7.8416	1.8096	0.0085	0.0083	0.0190
CNN only	1.0275	*0.8201*	*2.3610*	*0.2266*	3.0015	1.5330	*6.4361*	0.4219	*0.0083*	*0.0082*	*0.0120*
CNN + ATGV-L2	*1.0145*	0.8374	**2.3197**	0.2720	*2.9832*	*1.5175*	**6.4223**	*0.4124*	0.0084	0.0083	0.0120
ATGV-Net	**1.0021**	**0.8155**	2.3846	**0.1991**	**2.9293**	**1.5029**	6.6327	**0.3764**	**0.0081**	**0.0081**	**0.0117**

a ray for each pixel of the camera image. For each ray we compute the distance between the image plane and the closest surface it hits, or in the case it does not hit any surface, we return a maximum distance value for the background. In Figure 3 we illustrate two random examples of the more than 40,000 depth maps that we have generated with this method.

Given this generated depth maps as noise free ground-truth, we create the low-resolution depth maps $s_k^{(\mathrm{lr})} = \downarrow_\rho t_k$ for the network training by resampling the generated ground-truth depth maps t_k by the scale factor of ρ that is used in the evaluation. Depending on the dataset, we additionally add depth-dependent noise $\eta(s_k^{(\mathrm{lr})})$ to the low-resolution depth map. Finally, we upsample this low-resolution, probably noisy depth maps with bilinear interpolation to obtain our mid-level representation $s_k = \uparrow_\rho (s_k^{(\mathrm{lr})} + \eta(s_k^{(\mathrm{lr})}))$.

4.2 Clean Middlebury & Laserscan

In this first experiment we evaluate the performance of our proposed method on the images *Teddy*, *Cones*, *Tsukuba* and *Venus* of the Middlebury dataset as in [1, 13, 20]. The disparity is interpreted as depth and we test upsampling factors of ×2 and ×4. Additionally, we evaluate on the Laserscan dataset images *Scan21*, *Scan30* and *Scan42* with an upsampling factor of ×4 as in [1, 13]. We compare our results to simple upsamling methods, such as nearest neighbor and bicubic upsampling, as well as to state-of-the-art depth upsampling methods that rely on an additional guidance image as input [9, 12]. Further, we show the results of recent sparse coding based approaches for single image super-resolution [42, 36], two approaches based on a Markov Random Field [1, 20] and a recent variational approach that uses sparse coding to estimate edge priors [13]. To demonstrate the effect of our variational model on top of the deep network, we show the results of the high-resolution estimates of the network only (CNN only), the results, where we add the variational model, but without joint training (CNN + ATGV-L2), and the results after end-to-end training (*ATGV-Net*).



(a) Input & GT (b) Timofte *et al.* [36] (c) Ferstl *et al.* [13] (d) CNN only (e) ATGV-Net

Fig. 4. Qualitative results for the noise-free Middlebury image *Tsukuba*, $\rho = 4$. (a) depicts the ground-truth and the input data. (b) and (c) show the results of state-of-the-art methods. (d) and (e) present the results of the deep network only and our proposed model trained end-to-end. Best viewed magnified in the electronic version.

The results in terms of the root mean squared error (RMSE) are summarized in Table 1.[1] We can clearly see that the deep network already achieves a significant performance improvement compared to the other methods on both datasets and upsampling factors. Interestingly, we obtain even better results as the methods [9, 12] that utilize an additional guidance image for the upsampling. This is especially pronounced in test samples with structures that are well simulated in the training data, such as *Venus*. Further, the variational model on top of the network slightly increases the performance and training the whole model end-to-end gives the overall best results. One exception is the *Tsukuba* sample, where the results get slightly worse after end-to-end training. An explanation might be that fine, elongated structures, *e.g.* near at the lamp of *Tsukuba*, are not well represented in the training data. In the qualitative results, see Figure 4, we can further observe that the deep network with 10 layers achieves already very good results with sharper depth discontinuities compared to other methods. However, the improvement of the variational model on top of the deep network is hardly visible. This becomes more apparent in the next experiment.

4.3 Noisy Middlebury

In this experiment we evaluate our method on the Middlebury disparity maps *Art*, *Books* and *Moebius* with added depth dependent Gaussian noise to simulate the acquisition process of a Time-of-Flight sensor, as proposed by Park *et al.* [29]. Therefore, we add to our low-resolution synthetic training data $s_k^{(\mathrm{lr})}$ depth dependent Gaussian noise of the form $\eta(x) = \mathcal{N}(0, \sigma s_k^{(\mathrm{lr})}(x)^{-1})$, with $\sigma = 651$. Exemplar training images are depicted in Figure 3. We report quantitative results in Table 2 and visualize qualitative results in Figure 5.

We again compare our method to simple upsampling methods, such as nearest neighbor and bilinear interpolation. We compare our proposed method to other approaches that utilize an additional intensity image as guidance. Those methods include the Markov Random Field based approach in [9], the bilateral filtering

[1] Note that we present our results over the full disparity range $[0, 255]$, as opposed to *e.g.* [13], where the disparities are scaled to a narrower range.

Table 2. Results on noisy Middlebury data. We report the error as RMSE in pixel disparity and highlight the best result in boldface and the second best in italic.

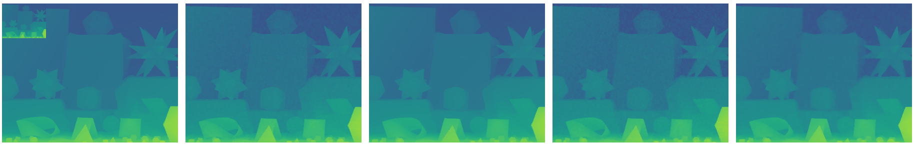
	×2			×4		
	Art	Books	Moebius	Art	Books	Moebius
NN	6.55	6.16	6.59	7.48	6.31	6.78
Bilinear	4.58	3.95	4.20	5.62	4.31	4.56
Yang *et al.* [41]	3.01	1.87	1.92	4.02	2.38	2.42
He *et al.* [18]	3.55	2.37	2.48	4.41	2.74	2.83
Diebel & Thrun [9]	3.49	2.06	2.13	4.51	3.00	3.11
Chan *et al.* [7]	3.44	2.09	2.08	4.46	2.77	2.76
Park *et al.* [29]	3.76	1.95	1.96	4.56	2.61	2.51
Ferstl *et al.* [12]	3.19	1.52	1.47	4.06	*2.21*	*2.03*
CNN only	2.02	1.27	1.50	3.55	2.41	2.68
CNN + ATGV-L2	*1.93*	*1.14*	*1.37*	*3.40*	2.24	2.51
ATGV-Net	**1.84**	**1.13**	**1.24**	**2.98**	**1.72**	**1.95**

with cost volume in [41], the guided image filter in [18], the noise-aware bilateral filter in [7], the non-local means filter in [29] and the variational model in [12]. To evaluate the influence of the variational model on top of the deep network, we report the results of the network only (CNN only), results with the variational model on top of the network, but without joint training (CNN + ATGV-L2), and the results after end-to-end training (ATGV-Net).

From the quantitative results in Table 2 we observe that the *CNN only* already performs better than state-of-the-art methods that utilize an additional guidance input for most images and upsampling factors. Further, the variational model on top of the deep network slightly improves the results, but end-to-end training of the whole model results in significant improvement. This improvement of *ATGV-Net* over the network only is also apparent in the qualitative results (Figure 5). We observe less noise in homogeneous areas in the *ATGV-Net* estimates, especially in the background, compared to the *CNN only* estimates. The results of [12] look also very sharp, but produce errors near depth discontinuities and in-between fine structures. In contrast, our method preserves those finer structures. We refer to the supplemental material for additional qualitative results, as well as quantitative results in terms of mean absolute error (MAE).

4.4 ToFMark

In our final experiment we evaluate our method on the challenging ToFMark dataset [12]. This dataset consists of three time-of-flight (ToF) depth maps of three different scenes. For each scene there exists an accurate high-resolution structured-light scan as ground-truth. The ToF depth maps have a resolution of 120×160 pixel and the target resolution, given by the guidance intensity image (that we do not use in our method) is 610×810 pixel. This corresponds to an upsampling factor of approximately $\rho = 5$. As the target high-resolution depth-map is given in the camera coordinate system of the structured light scanner, we prepare our training data accordingly. We project our high-resolution synthetic training depth maps to the ToF coordinate system using the provided projection matrix. In the low-resolution depth maps we add depth dependent noise and back project the remaining points to the target camera coordinate system. This yields



(a) Input & GT (b) He *et al.* [18] (c) Ferstl *et al.* [12] (d) CNN only (e) ATGV-Net

Fig. 5. Qualitative results for the noisy Middlebury image *Moebius*, $\rho = 4$. (a) depicts the ground-truth and the input data. (b) and (c) show the results of state-of-the-art methods. (d) and (e) present the results of the deep network only and our proposed model trained end-to-end. Best viewed magnified in the electronic version.

	Books	Devil	Shark
NN	30.46	27.53	38.21
Bilinear	29.11	25.34	36.34
Kopf *et al.* [23]	27.82	24.30	34.79
He *et al.* [18]	27.11	23.45	33.26
Ferstl *et al.* [12]	**24.00**	*23.19*	*29.89*
ATGV-Net	*24.67*	**21.74**	**28.51**

Table 3. Results on real Time-of-Flight data from the ToFMark benchmark dataset. We report the error as RMSE in mm and highlight the best result in boldface and the second best in italic.

a very sparse depth map that we subsequently inpaint with bilinear interpolation to obtain our final mid-resolution training inputs.

We compare our results to simple nearest neighbour and bilinear interpolation, and three state-of-the-art depth map super-resolution methods that utilize an additional guidance image as input. The quantitative results are shown in Table 3 as RMSE in mm. Please see the supplemental material for qualitative results. Even on this difficult dataset we are at least on par with state-of-the-art methods that utilize an additional intensity image as guidance input.

5 Conclusion

We presented a combination of a deep convolutional network with a variational model for single depth map super-resolution. We designed the convolutional network to compute the high-resolution depth map, as well as the depth discontinuities. The network output was utilized in our variational model to further refine the result. By unrolling the optimization procedure of the variational model, we were able to optimize the joint model end-to-end, which lead to improved accuracy. Further, we demonstrated the feasibility to train our method on a massive amount of synthetic generated depth data and obtain state-of-the-art results on four different benchmarks. Our model is especially useful if the low-resolution depth map contains noise, which is the case for most consumer depth sensors. In future work we plan to extend our model to depth data that contain larger areas of missing pixels, *e.g.* from structured light sensors. This is straight-forward by setting $w_\lambda = 0$ for areas where depth measurements are missing.

Acknowledgment: This work was supported by *Infineon Technologies Austria AG* and the Austrian Research Promotion Agency under the *FIT-IT Bridge* program, project #838513 (TOFUSION).

References

1. Aodha, O.M., Campbell, N.D., Nair, A., Brostow, G.J.: Patch Based Synthesis for Single Depth Image Super-Resolution. In: European Conference on Computer Vision (ECCV) (2012)
2. Apple, A.: Some techniques for shading machine renderings of solids. In: Proceedings of the April 30–May 2, 1968, spring joint computer conference (1968)
3. Bredies, K., Kunisch, K., Pock, T.: Total Generalized Variation. SIAM Journal on Imaging Sciences 3(3), 492–526 (2010)
4. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: European Conference on Computer Vision (ECCV) (2012)
5. Chambolle, A., Pock, T.: A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. Journal of Mathematical Imaging and Vision 40(1), 120–145 (2011)
6. Chambolle, A., Pock, T.: On the ergodic convergence rates of a first-order primal-dual algorithm. Mathematic Programming pp. 1–35 (2015)
7. Chan, D., Buisman, H., Theobalt, C., Thrun, S.: A Noise-aware Filter for Real-time Depth Upsampling. In: European Conference on Computer Vision Workshops (ECCVW) (2008)
8. Chen, L.C., Schwing, A.G., Yuille, A.L., Urtasun, R.: Learning Deep Structured Models. In: Proceedings of the International Conference on Machine Learning (ICML) (2015)
9. Diebel, J., Thrun, S.: An Application of Markov Random Fields to Range Sensing. In: Proceedings of Conference on Neural Information Processing Systems (NIPS) (2005)
10. Domke, J.: Generic Methods for Optimization-Based Modeling. In: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS) (2012)
11. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a Deep Convolutional Network for Image Super-Resolution. In: European Conference on Computer Vision (ECCV) (2014)
12. Ferstl, D., Reinbacher, C., Ranftl, R., Rüther, M., Bischof, H.: Image Guided Depth Upsampling using Anisotropic Total Generalized Variation. In: IEEE International Conference on Computer Vision (ICCV) (2013)
13. Ferstl, D., Rüther, M., Bischof, H.: Variational Depth Superresolution using Example-Based Edge Representations. In: IEEE International Conference on Computer Vision (ICCV) (2015)
14. Girshick, R., Shotton, J., Kohli, P., Criminisi, A., Fitzgibbon, A.W.: Efficient Regression of General-Activity Human Poses from Depth Images. In: IEEE International Conference on Computer Vision (ICCV) (2011)
15. Glasner, D., Bagon, S., Irani, M.: Super-Resolution from Single Image. In: IEEE International Conference on Computer Vision (ICCV) (2009)
16. Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning Rich Features from RGB-D Images for Object Detection and Segmentation. In: European Conference on Computer Vision (ECCV) (2014)
17. Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., Cipolla, R.: Understanding Real World Indoor Scenes With Synthetic Data. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

18. He, K., Sun, J., Tang, X.: Guided Image Filtering. In: European Conference on Computer Vision (ECCV) (2010)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
20. Hornáček, M., Rhemann, C., Gelautz, M., Rother, C.: Depth Super Resolution by Rigid Body Self-Similarity in 3D. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
21. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A.: KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In: ACM Symposium on User Interface Software and Technology (2011)
22. Kim, J., Lee, J.K., Lee, K.M.: Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
23. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint Bilateral Upsampling. ACM Transactions on Graphics (TOG) 26(3), 96 (2007)
24. Krähenbühl, P., Koltun, V.: Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In: Proceedings of Conference on Neural Information Processing Systems (NIPS) (2012)
25. Kwon, H., Tai, Y.W., Lin, S.: Data-Driven Depth Map Refinement via Multi-scale Spare Representations. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
26. Martull, S., Peris, M., Fukui, K.: Realistic CG Stereo Image Dataset with Ground Truth Disparity Maps. In: International Conference on Pattern Recognition Workshops (ICPRW) (2012)
27. Nagel, H.H., Enkelmann, W.: An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields from Image Sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 8(5), 565–593 (1986)
28. Ochs, P., Ranftl, R., Brox, T., Pock, T.: Bilievel Optimization with Nonsmooth Lower Level Problems. In: Scale Space and Variational Methods in Computer Vision (SSVM) (2015)
29. Park, J., Kim, H., Tai, Y.W., Brown, M.S., Kweon, I.S.: High Quality Depth Map Upsampling for 3D-TOF Cameras. In: IEEE International Conference on Computer Vision (ICCV) (2011)
30. Ranftl, R., Gehrig, S., Pock, T., Bischof, H.: Pushing the Limits of Stereo Using Variational Stereo Estimation. In: IEEE Intelligent Vehicles Symposium (2012)
31. Ranftl, R., Pock, T.: A Deep Variational Model for Image Segmentation. In: German Conference on Pattern Recognition (GCPR) (2014)
32. Riegler, G., Ranftl, R., Rüther, M., Bischof, H.: Joint Training of an Convolutional Neural Net and a Global Regression Model. In: Proceedings of the British Machine Vision Conference (BMVC) (2015)
33. Schulter, S., Leistner, C., Bischof, H.: Fast and Accurate Image Upscaling with Super-Resolution Forests. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
34. Schwing, A.G., Urtasun, R.: Fully Connected Deep Structured Networks. In: arXiv preprint arXiv:1503.02351 (2015)
35. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time Human Pose Recognition in Parts from Single Depth Images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011)

36. Timofte, R., Smet, V.D., Gool, L.V.: Anchored Neighborhood Regression for Fast Example-Based Super-Resolution. In: IEEE International Conference on Computer Vision (ICCV) (2013)
37. Timofte, R., Smet, V.D., Gool, L.V.: A+: Adjusted Anchored Neighborhood Regression for Fast Super-Resolution. In: Asian Conference on Computer Vision (ACCV) (2014)
38. Tompson, J., Jain, A., LeCun, Y., Bregler, C.: Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. In: Proceedings of Conference on Neural Information Processing Systems (NIPS) (2014)
39. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L1 Optical Flow. In: Proceedings of the British Machine Vision Conference (BMVC) (2009)
40. Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image Super-Resolution Via Sparse Representation. IEEE Transactions on Image Processing 19(11), 2861–2873 (2010)
41. Yang, Q., Yang, R., Davis, J., Nistér, D.: Spatial-Depth Super Resolution for Range Images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007)
42. Zeyde, R., Elad, M., Protter, M.: On Single Image Scale-Up Using Sparse-Representations. In: Curves and Surfaces (2010)
43. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.: Conditional Random Fields as Recurrent Neural Networks. In: IEEE International Conference on Computer Vision (ICCV) (2015)