AUTOMATIC 2D-TO-3D CONVERSION USING MULTI-SCALE DEEP NEURAL NETWORK

Jiyoung Lee, Hyungjoo Jung, Youngjung Kim, and Kwanghoon Sohn

School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea

ABSTRACT

In this paper, we address a problem of generating a virtual right-view from a single left image. Traditional methods usually have separate stages, i.e, depth (or disparity) estimation for a given single image and depth image-based rendering (DIBR), and require ground-truth depth as supervision. In contrast, using spatial transformer module, our method trains a deep convolutional neural network (CNN) directly on stereo image pairs captured in outdoor environments. This makes it possible to exploit orders of magnitude data, in where high-quality depth recording is challenging, and significantly increases performance. To capture large displacements between images, we further propose multi-scale deep architecture that works from coarse to fine. The idea is that the displacements are always less than a few pixels at each scale. Experimental results demonstrate the effectiveness of the proposed method over state-of-the-art approaches both qualitatively and quantitatively.

Index Terms— Automatic 2D-to-3D conversion, view extrapolation, multi-scale deep neural network, depth image-based rendering.

1. INTRODUCTION

Estimating 3D structure from images is a fundamental task in image processing, computer vision and graphics. In this work, we aim to solve the related problem of automatic 2D-to-3D conversion, where the goal is to synthesize a virtual right-view by warping a given single left-view. It can be used for a variety of applications such as 3D-TV, virtual reality, autonomous vehicle [1], and video stabilization [2]. The majority of existing techniques for 2D-to-3D conversion consists of two steps: single image depth estimation and depth-based image rendering (DIBR) in order to form a stereo pair. In the following, we briefly review these steps separately.

The problem of 2D-to-3D conversion is strongly related to the problem of predicting depth from a single image. In recent years, machine learning approaches have greatly advanced the accuracy of depth prediction problem. Saxena *et al.* [3] modeled monocular cues based on the MRF whose edges encode a simple smoothness assumption between neighboring superpixels. Karsch *et al.* [4] devised the depth transfer algorithm using retrieved similar images in the training set. Their retrieval is performed using the GIST descriptor at a whole image level, followed by dense scene alignment [5]. A global optimization step is then utilized to combine depth from the aligned images. Konrad *et al.* [6] argued that dense scene alignment of [4] is computationally expensive and does not necessarily improve the quality of depth estimation. Instead, they directly fused the retrieved depth by computing a median value for each pixel. More recent methods have used convolutional neural networks (CNN) and Kinect data [7] for supervised training. Eigen *et al.* applied the CNNs in multiple stages to generate features and refine depth prediction to higher resolution [8]. In [9], relative depth annotations rather than metric depth were used to improve the performance in unconstrained settings. Laina *et al.* [10] devised a fast up-projection layer and combined it with the deep residual learning [11]. These methods have achieved state-of-the-art performance on several benchmarks, but require large-scale ground-truth depth maps for training. Recording high-quality depth maps in a range of environments (especially for outdoor) is very difficult.

The DIBR is one of the most important techniques to synthesize virtual view at different viewpoints using a 3D warping process. When a virtual view is located between two real cameras, i.e., view interpolation, most occluded regions can be handled by combining image and depth from multiple views. In automatic 2D-to-3D conversion, however, the problem becomes even more complicated since the view is extrapolated from a single view. Early methods used a Gaussian filter to smooth the depth image [12,13]. The main drawback of this is that it smooths depth discontinuities, introducing geometric distortions. Azzari *et al.* [14] filled the occluded region using non-local means of similar patches. Recently, Choi *et al.* used the random walker segmentation for extrapolating a virtual view [15].

In this paper, we combine the single image depth estimation and DIBR process into a unified deep CNN framework. Using a spatial transformer module [17], the problem of 2D-to-3D conversion is reformulated as an image reconstruction problem. This allows our model to be trained end-to-end using stereo image pairs only (without ground-truth disparity as supervision). We further propose multi-scale deep architecture to capture large displacements between stereo images. The disparity map estimated by the network is optimized only to synthesize a good virtual right-view, performing occlusion

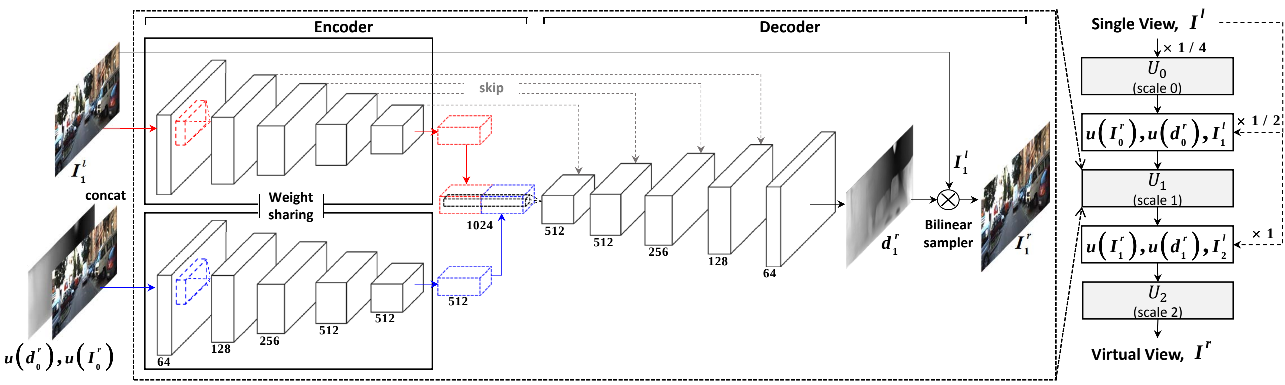

Fig. 1. The architecture of the proposed multi-scale CNNs. Each scale of network consists of a pair of encoder-decoder network. Encoder parts require both of the resized left view and the outputs of previous scale network (including disparity and virtual right view), except the coarsest scale.

filling implicitly. Note that our method is highly related to recent work of Deep3D [16] that integrates the DIBR process into the CNN using an probabilistic disparity representation. But, our approach uses a multi-scale design to directly represent the disparity maps. We will show that the proposed method gives much better results than the Deep3D [16].

2. PROPOSED METHOD

A typical 2D-to-3D conversion process consists of two steps: estimating a depth (or disparity) map from the given left-view I^l and rendering the virtual right-view I^r using a depth image-based rendering algorithm. We combine these two steps into a unified deep CNN framework, which can be trained by standard back-propagation. Our approach uses the spatial transformer module [17] in a coarse-to-fine manner, compensating large displacements between left-right views. Note that our approach does not require disparity maps as supervision for training. Next, we describe the network and training procedure in detail.

2.1. Spatial Transformer and 2D-to-3D Conversion

We propose to use the spatial transformer module [17] in the CNNs for automatic 2D-to-3D conversion. It was originally introduced by Jaderberg *et al.* [17] and aimed to find an affine transformation for spatially invariant classification. In contrast, our transformation is defined by per-pixel disparity map. The task is then to estimate the best disparity map d^r relating a right-view I^r with a given left-view I^l. The disparity map is assumed to be pixel-wise dense, allowing displacing each pixel of I^l to a new position aligned to I^r. The resulting pixel displacement requires interpolation back onto a regular grid. We use bilinear interpolation $\mathcal{B}\{\cdot, d\}$, and express the synthesized right-view as:

$$I^r = \mathcal{B}\{I^l, d^r\}, \tag{1}$$

i.e., through backward warping. Our formulation of (1) is fully differentiable with respect to d^r, and therefore allows back-propagation of the error.

2.2. Inference

We adopt a multi-scale design to represent the disparity d^r from single left-view I^l. Let $\{U_0, \dots, U_K\}$ denote a set of trained CNNs with downsampling factors $\{2^K, \dots, 2^0\}$. In this work, we use 3-scale architecture ($K = 2$) and the corresponding schematic of design is illustrated in Fig. 1.

Since our objective is to synthesize I^r from single I^l, each network has slightly different input configurations. The CNN U_0 takes I_0^l only as input, and computes the disparity d_0^r and I_0^r as follows:

$$d_0^r = U_0(I_0^l), \quad I_0^r = \mathcal{B}\{I_0^l, d_0^r\}. \tag{2}$$

Whereas the others U_1, U_2 use the estimated disparity map and synthesized view from the previous scale:

$$\begin{aligned} d_k^r &= U_k(I_k^l, u(I_{k-1}^r), u(d_{k-1}^r)), \\ I_k^r &= B\{I_k^l, d_k^r\}, \end{aligned} \tag{3}$$

where $u(\cdot)$ is a upsampling operator that increases the spatial resolution ($\times 2$). That is, we upsample the resulting disparity $u(d_{k-1}^r)$ and right-view $u(I_{k-1}^r)$, and pass these to the next scale U_k along with I_k^l. Note that at each scale, we synthesize right-view I_k^r using (1). The problem in finer scale (U_1, U_2) becomes similar to stereo matching since the network takes the left and synthesized right views as inputs. It makes the problem much easier than using I_k^l only.

2.3. Network Architecture

Each network U_k has a fully convolutional encoder-decoder architecture [18] that takes input of arbitrary size and produces correspondingly-sized output. The encoder consists of

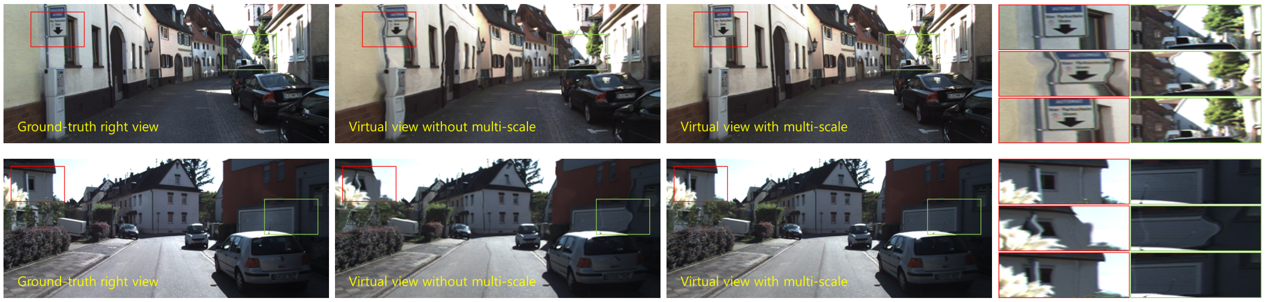

Fig. 2. Comparison between single- and multi-scale networks. Multi-scale network outperforms the single-scale network, especially foreground regions (large disparity values). For the visibility, highlighted boxes are enlarged in the last column.

the repeated application of three 3×3 convolutions and rectified linear unit (ReLU), followed by (stride 2) max-pooling. For the extra inputs ($u(d^r_{k-1})$, $u(I^r_{k-1})$), we first concatenate them and adapt the late fusion strategy [19] using the same encoder architecture (see Fig. 1). The decoder part progressively enlarges the spatial resolution of CNN features through a sequence of deconvolution (a factor of 2) and convolution layers[1].

There is a trade-off between localization accuracy of the output and the use of global context. The encoder-decoder architecture requires a series of convolution and max-pooling layers to robustly estimate the disparity map d^r_k. The subtle details of the disparity map, however, are lost during these process. Inspired by [20], we add skip connections between two corresponding convolution and deconvolution layers, as shown in Fig. 1. The feature maps shuttled by skip connections carry much details, which help the network to estimate accurate disparity maps.

2.4. Training

We train each of the CNNs $\{U_0, \ldots, U_{K=2}\}$ independently and sequentially to synthesize the right-view I^r_k. Given a collection of M training stereo paris, the mean absolute error (MAE) criterion is applied to every scale of networks. Hence, the loss function is defined as follows

$$\mathcal{L}_k = \frac{1}{M} \sum_{p=1}^{M} \frac{1}{c_k w_k h_k} \left\| I^{r,(p)}_k - r^{(p)}_k \right\|_1, \quad (4)$$

where r_k indicates the ground-truth right view downsampled at scale k. The loss at each scale is normalized by the number of channels c_k, width w_k, and height h_k. We use the standard stochastic gradient descent (SGD) to minimize (4). The derivative for the back-propagation is obtained as follows:

$$\frac{\partial \mathcal{L}_k}{\partial I^{r,(p)}_k} \propto sgn \left(I^{r,(p)}_p - r^{(p)}_k \right), \quad (5)$$

[1]We add the batch normalization [18] after every convolution layers in the decoder

Table 1. Average PSNR and SSIM on 1,578 test images from KITTI dataset [23].

	Deep3D [16]	Ours ($K=0$)	Ours ($K=2$)
PSNR	30.18	29.23	**31.85**
SSIM	0.8677	0.8569	**0.9038**

where $sgn(\cdot)$ denotes the signum function. Each network U_k uses the previous scale U_{k-1} as initialization, except U_0. The encoder and decoder parts of U_0 are initialized using pretrained VGG-16 model [21] and normal distribution with zero mean, respectively.

3. EXPERIMENTS

3.1. Implementation Details

We implement the proposed method using the VLFeat MatConvNet[2] [22]. We modify the *BilinearSampler* layer in MatConvNet [22] to realize the spatial transformer module [17]. The source codes for training and testing will be made publicly available. Training is done on a standard desktop with 12GB NVIDIA Titan GPU using 35 thousand stereo pairs from the KITTI dataset [23]. The training images are resized to 784×256[3] for the efficient training. Data augmentation is performed on the fly, applying random transform to the training data. It includes in-plane rotation, translation, flip, and brightness shift. We use a learning rate of 10^{-4} for the first 10 epoch and decrease it to 10^{-6} until the networks converge (40 epoch). For each scale, the batch sizes are set to 32, 16 and 8, respectively. In all cases, the momentum and weight decay parameters are set to 0.9 and 0.0005, respectively.

We compare the proposed method to the Deep3D [16] model, which also does not require the ground-truth disparity maps for training. Since the Deep3D [16] is not trained on

[2]http://www.vlfeat.org/matconvnet/

[3]The input sizes of each scale are 196×64, 392×128, and 784×256 respectively.

Fig. 3. Qualitative results for 2D-to-3D conversion: (From left to right) ground-truth right view, Deep3D [16], and ours. For accurate comparison, highlighted boxes are stacked in the last column.

Fig. 4. Disparity maps and virtual views from Deep3D [16] and ours.

the KITTI dataset [23], we retrain it using the source code[4] provided by the authors.

3.2. Results on KITTI

We evaluate the proposed method on $1,578$ KITTI test images [23]. Figure 2 demonstrates the effectiveness of our multi-scale architecture for auto-matric 2D-to-3D conversion. Both the single- and the multi-scale networks can synthesize the background regions correctly (the corresponding disparity is relatively small). However, the former has a difficulty in reconstructing the foreground object which has the large disparity value. The proposed multi-scale network successfully synthesizes the foreground object and thin structures (the blue and red boxes in Fig. 2).

Fig. 3 shows the visual comparison of our method with the Deep3D [16]. Since the Deep3D [16] synthesize the virtual view by calculating weighed averages of shifted input left-view, the resulting image becomes smooth. In contrast, the proposed method samples the textures from the input left-view directly, and shows more natural results. The estimated disparity maps both of the Deep3D and ours have incorrect values in homogenous regions (Fig. 4). This is because both methods optimize the CNNs to estimate the good virtual views, not disparity maps. However, the proposed method still visually plausible disparity maps. We measure peak signal-to-noise ratio (PSNR) and SSIM as quantitative comparison on the KITTI dataset [23]. The results are summarized in Table 1. We observe that the proposed method is more accurate than the Deep3D [16] in both metrics. Finally, in Fig. 4 we show stereo anaglyph generated by the Deep3D and ours, which are constructed from the input left view and virtual right view.

[4]https://github.com/piiswrong/deep3d

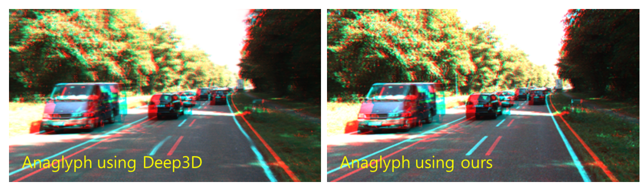

Fig. 5. Stereo Anaglyph generated by the Deep3D [16] and the proposed method.

4. CONCLUSION

In this paper, we propose multi-scale convolutional neural networks for automatic 2D-to-3D conversion. Different from existing methods, we combine the single image depth estimation and DIBR process into a unified deep CNN framework. Using the spatial transform module, we train our model end-to-end on a large-scale stereo image dataset. We also proposed multi-scale deep architecture to capture large displacements between stereo images. Intensive experiments demonstrate the superiority of the proposed method over state-of-the-art methods both qualitatively and quantitatively.

5. REFERENCES

[1] H. Yoo, J. Son, B. Ham, and K. Sohn, ”Real-time rear obstacle detection using reliable disparity for driver assistance,” in *Expert Systems with Application*, vol. 56, no. 1, 2016.

[2] J. Kopf, M. F. Cohen, and R. Szeliski, ”First-person hyperlapse videos,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012.

[3] A. Saxena, M. Sun, and A.Y. Ng, “Make3D: Learning 3D-scene structure from a single still image,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, 2009.

[4] K. Karsch, C. Liu, and S.B. Kang, “Depth extraction from video using non-parametric sampling,” in *Proc. Eur. Conf. Comput. Vis.*, pp. 775-788, 2012.

[5] C. Liu, J. Yuen, and A. Torralba, “SIFTFLOW: Dense correspondence across scenes and its applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, 2011.

[6] J. Konrad, M. Wang, and P. Ishwar, “Learning-based, auto-matic 2d-to-3d image and video conversion,” *IEEE Trans. Image Process.*, vol. 22, no. 9, 2013.

[7] N. Silberman, P. Kohli, D. Hoiem, and R. Fergus, “Indoor segmentation and support inference from RGB-D images,” in *Proc. Eur. Conf. Comput. Vis.*, pp. 746-760, 2012.

[8] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in *IEEE Int. Conf. on Computer Vision*, pp. 2650-2658, 2015.

[9] W. Chen, Z. Fu, D. Yang, and J. Deng, “Single image depth perception in the wild,” in *Advances in Neural Information Processing Systems*, 2016.

[10] I. Laina, C. Rupprecht, and V. Belagiannis, “Deeper depth prediction with fully convolutional residual networks,” in *Proc. Int. Conf. on 3D Vision*, 2016.

[11] J. Kim, J. Lee, and K. Lee, “Accurate image super-resolution using very deep convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[12] S. Zinger, D. Ruijters, L. Do, and H.N. Peter, “View interpolation for medical images on autostereoscopic displays,” in *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 1, 2012.

[13] C. Fehn, ”Depth-image-based rendering DIBR, compression and transmission for a new approach on 3D-TV,” *Proc. SPIE*, vol. 5291, pp. 93-104, Jan. 2004.

[14] L. Azzari, F. Battisti, and A. Gotchev, “Comparative analysis of occlusion-filling techniques in depth image-based rendering for 3D videos,” in *Proc. 3rd Workshop Mobile Video*, 2010.

[15] S. Choi, B. Ham, and K. Sohn, “Space-time hole filling with random walks in view extrapolation for 3D video,” *IEEE Trans. Image Process.*, vol. 22, no. 6, 2013.

[16] J. Xie, R. Girshick, and A. Farhadi, “Deep3d: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks,” in *Proc. Eur. Conf. Comput. Vis.*, pp. 842-857, 2016.

[17] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial transformer networks,” in emphAdvances in Neural Information Processing Systems, pp. 2017-2025, 2015.

[18] H. Noh, S. Hong, and B. Han, “Learning deconvolutional network for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015.

[19] J. Liu, S. Zhang, S. Wang, and D.N. Metaxas, “Multispectral deep neural networks for pedestrian detection,” in *Proc. British Machine Vision Confrence*, 2016

[20] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015

[21] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *IEEE Int. Conf. on Learning Representations*, 2015.

[22] http://www.vlfeat.org/matconvnet.

[23] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in emphIEEE Int. Conf. on Computer Vision, pp. 3354-3361, 2012.