Deep Stereoscopy from a Single Image

Anonymous Author(s)

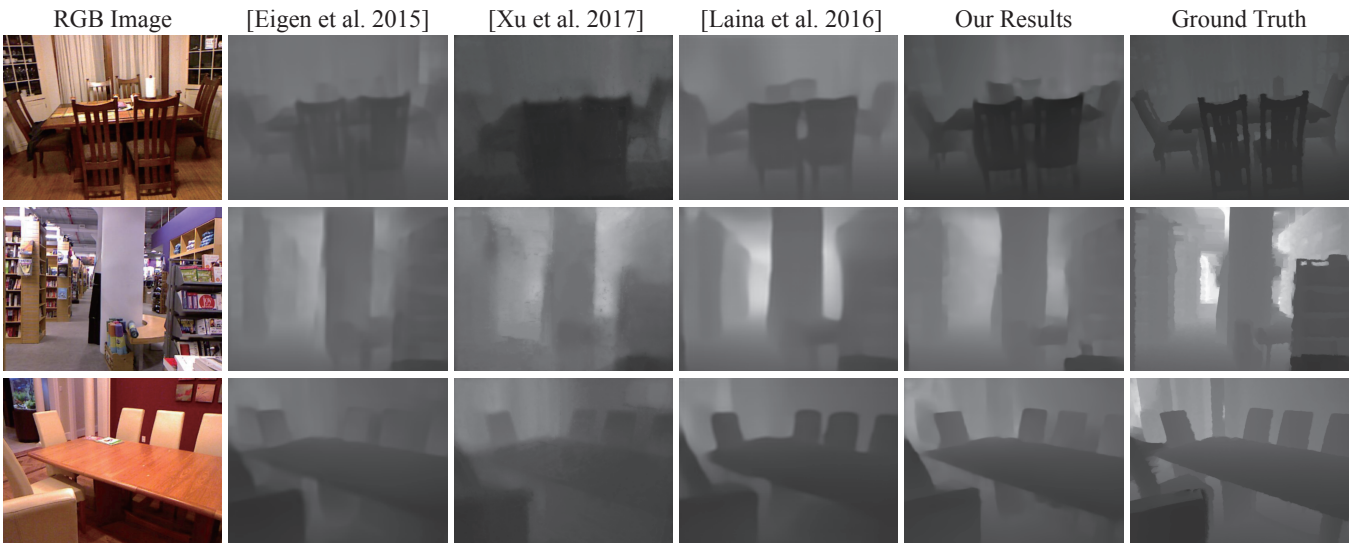

Figure 1: Comparison with state-of-the-art depth prediction methods based on convolutional neural network. The input to the networks is a color image shown in the first column. We train a fully convolutional neural network with designed side-input layers to enhance the quality of predicted depth edges, as shown at the depth edges of chairs in the first row and the sharper depth edges in the last two rows. The enhanced depth edges can help to prevent the distortion of object boundaries in the generation of stereo images as shown in the experimental results.

ABSTRACT

In this paper, we present a novel deep-learning based pipeline to create stereo image pairs from a single image. A fully convolutional neural network is designed for the depth prediction task. We integrate scale-invariant L1, L2 loss and rank loss in the training to improve the prediction accuracy. Since the depth values along object boundaries are critical to reduce the distortion of edges in the synthesized images at new viewpoints, we design side-input layers to directly learn proper scale features for depth edge enhancement. Residual blocks are used in the network to facilitate the gradient back-propagation. The input image is then warped to a new viewpoint using the disparity information computed via the predicted depth, and the large dis-occluded areas are inpainted by a trained generative adversarial network. Experimental results show that the pipeline is effective in generating stereo image pairs from RGB images with the predicted depth information.

CCS CONCEPTS

•**Computing methodologies** → **Image processing;** *Image-based rendering;*

KEYWORDS

Deep convolutional neural network, Depth from a single image, Stereoscopy

ACM Reference format:
Anonymous Author(s). 2016. Deep Stereoscopy from a Single Image. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 11 pages.
DOI: 10.475/123_4

1 INTRODUCTION

Stereoscopy is a technique to simulate binocular vision of human eyes: it presents observers with two different images with disparities to create the illusion of depth. It has been widely used in 3D display systems, such as VR glasses and naked eye 3D display. Through RGB-D images, i.e., color images with depth information, it is straightforward to compute the disparity information required in the synthesis of stereo images. The depth information can also be manipulated for special stereoscopic viewing effects. However, specialized devices, such as depth cameras, are necessary to capture RGB-D images.

Our target is to automatically augment color images with depth information so as to significantly enlarge the application scope of the stereoscopic viewing of images. Although depth or 3D from a single image has been widely investigated, the existing researches often assume planar or cylindrical structures in a scene or handle specific types of objects through data-driven methods, which limit

Conference'17, Washington, DC, USA
2016. 123-4567-24-567/17/06…$15.00
DOI: 10.475/123_4

their application scenarios[Huang et al. 2015; Saxena et al. 2009; Zeng et al. 2015]. Another line of research works predicts pixel-level depth information through the training of deep convolutional neural network (CNN) on RGB-D datasets [Eigen and Fergus 2015; Kendall and Gal 2017; Laina et al. 2016; Liu et al. 2015; Xu et al. 2017]. Although the training is time-consuming, such algorithms can handle complicated scene images and are fast in prediction, rendering them attractive in the application of stereoscopy.

In this paper, we present a novel deep-learning based pipeline for stereoscopy. The main contribution is a carefully designed fully convolutional nerual network using side-input layers for the depth prediction task [Shelhamer et al. 2017], and residual blocks are used to facilitate the gradient back-propagation [He et al. 2016]. Our improvements are two-fold. First, we integrate scale-invariant L1, L2 loss and rank loss in [Chen et al. 2016] in the training to improve the prediction accuracy. L1 and L2 loss are imposed to reduce the reconstruction error at pixels. In contrast, the rank loss allows us to specify the relative depth between non-adjacent pixels. It is able to further regularize the training with non-local information. Such combination can balance between local view and non-local view of the network and improve the accuracy of depth prediction. Second, we design side-input layers to learn proper scale features for depth edge enhancement, which is important to reduce the distortion of the edges in view sythesis at new viewpoints. They are inspired by multi-scale convolution network in [**??**], while we significantly extend it in the selection of feature scale. Particularly, the side-input layers are strided convolution layer and only apply for half-size depth image reconstruction, since we found that its usage in full-size depth image reconstruction might lead to undesirable noises. The reason is that the color image edges matched with depth edges are usually object boundaries, whose scale is larger than texture edges. Thus small disturbance should not be introduced to disturb the prediction.

In the generation of stereo image pairs, the input image is first warped with disparity information to simulate viewpoint translation. Although our method has ehanced the reconstructed depth endges, there still might be slight mis-match between color edges and depth edges at some object boundaries. It will lead to errorneous warped pixels. To this end, we remove these pixels via foreground object protection in [Lu et al. 2012], which can be viewed as a special type of depth smoothing at foreground object boundaries. Second, the dis-occluded areas in a generated image form holes and they can be filled with background information. We thus train a globally and locally consistent generative adversarial network (GAN) presented in [Iizuka et al. 2017] to predict the missing information at holes from their surrounding pixel values. The depth prediction and GAN-based inpainting networks form a pipeline for depth-based stereoscopy.

We have tested the designed pipeline on NYU indoor dataset and synthetic SYNTHIA outdoor driving scene dataset. It is effective in depth prediction and the subsequent stereo image pairs generation. To better demonstration the disparity effect, we also produce videos to show how the images are changed according to the viewpoint translation.

2 RELATED WORK

Depth from a single image: Estimating depth from a single image with depth cues, such as perspective effect, occlusion and vanishing lines, is a challenging task. The Make3D system predicted the depth information at super-pixels through Markov random field, using plane paramemters probablity conditioned on image features as variables [Saxena et al. 2009]. Karsch et al. [2014] proposed a kNN depth transfer algorithm based on SIFT flow, where the depth of an input image is tranferred from its similar images in a RGBD database. Zhang et al. [2015] adopted segmentation and depth ordering to compute depth information from a single image.

CNN architecture provides an efficient way to group global and local depth cues to regress on pixel-level depth values. Eigen et al. [2014] proposed a multi-scale deep CNN network for depth regression. It first estimates global structure of the depth image at coarse scale and then refines its details at second scale. They further improved the prediction accuracy of the multi-scale network by training the network to regress on segmentation, normal and depth simultaneously [Eigen and Fergus 2015]. Conditional random field can also be end-to-end trained with the CNN to fuse multi-scale or super-pixel level depth prediction results [Liu et al. 2015; Xu et al. 2017]. State-of-the-art convolutional network architectures, such ResNet and DenseNet, are also tuned for depth prediction [Kendall and Gal 2017; Laina et al. 2016]. Chen et al. [2016] adopted rank loss to train a CNN to predict relative depth for images in the wild.

In our work, we use fully convolutional neural network structure with side-input layers and combine L1, L2 and rank loss to improve the depth prediction accuracy.

Stereoscopy: Presenting the recorded scene with stereo images or videos can create the illusion of depth, which is important to the immersive user experience in virtual reality applications. One way of stereoscopy is to capture stereo images directly with stereo cameras, such as Bumblebee and ZED stereo cameras [1394a 2017; Camera 2017; Lee and Kweon 2000]. There are also cameras to capture stereo panaroma videos, such as Facebook surround 360 and Nokia OZO cameras [Facebook 2017; Nokia 2017]. Recently, Anderson et al. [2016] developed a omnidirectional stereo video capture system.

Given a left view image with depth information, the right view image of stereoscopy can be synthesized through image forward mapping. However, the dis-occluded areas will be holes in the right image. Asymmetric smoothing of depth map can reduce the hole area to ease the subsequent hole filling step [Zhang and Tam 2005]. In 3DTV, Image inpainting techniques integrated with depth information are also widely investigated [Cheng et al. 2008; Daribo and Pesquet-Popescu 2010; han Lu et al. 2012; Köppel et al. 2016; Ndjiki-Nya et al. 2011] in the production of stereo videos.

Stereo images can be synthesized through image-based rendering or view synthesis techniques [Buehler et al. 2001; Goesele et al. 2010; Levoy and Hanrahan 1996]. Given multi-view images and the reconstructed sparse 3D points, Chaurasia et al. proposed silhouette-aware global image warping and super-pixel level local warping to synthesize new views [Chaurasia et al. 2013, 2011]. Deep CNN is also applied to predict new views from light field or posed multi-view images [Flynn et al. 2016; Kalantari et al. 2016]. They are end-to-end trained to obtain the new view images directly. Zhou et al. [2016] proposed to learn appearance flow to synthesize new views. It works well for single object images but produces distorted scene images. The reason might be that the geometric constraints required in view synthesis are not strongly enforced.

Xie et al. [2016] trained a deep CNN on 3D movies to predict right eye views directly. While it avoids the difficulty to estimate depth information, it is also limited by the baseline setting in the dataset and not suited to the application that needs to tune the baseline or depth information for special effects.

CNN-based image inpainting: The target of image inpainting or image completion algorithms is to reconstruct the lost parts in images. In [Bertalmio et al. 2000], the pixel values are extended to the lost part along isophotes. Patch-based inpainting methods are developed [Barnes et al. 2009; Hays and Efros 2008; Huang et al. 2014; Sun et al. 2005] to better reconstruct the contextual structures in images. Please refer to [Ravi et al. 2013] for a comprehensive survey of image inpainting methods.

CNN-based image inpainting directly learns a mapping to predict the missing information [Köhler et al. 2014; Ren et al. 2015; Xie et al. 2012]. By interpreting images as samples from a high-dimensional probability distribution, image inpainting can be realized by generative adversarial networks [Arjovsky et al. 2017; Goodfellow et al. 2014; Mao et al. 2016; Mroueh and Sercu 2017; Radford et al. 2015]. The context encoder method trained a CNN network with both least squares loss and adversarial loss to measure the quality of reconstructed images [Pathak et al. 2016]. A recent contribution in [Iizuka et al. 2017] proposed a globally and locally consistent GAN based inpainting method. It further improves the quality of the image inpainting results, which is employed in our pipeline.

3 DEPTH PREDICTION NETWORK

The deep CNN network designed for depth prediction has two features: 1) We design side-input layers so as to learn features of proper scale to enhance the edges in the reconstructed depth. 2) We combine $L1$, $L2$ and rank loss at the training stage to improve the prediction accuracy. In this section, we describe the details of the network structure and loss functions for training.

3.1 Model Structure

We employ fully convolutional network for the pixel-level depth regression task. The network we design consists of two parts: a ResNet50 down-sampling network and an up-sampling network to reconstruct output depth images. The input to our network is 320×240 RGB images, and the output is a 320×240 depth image. Figure 2 illustrates the overall network structure. The network parameters of ResNet50 can be found in [He et al. 2016].

Up-sampling network: It consists of 4 building blocks, and each building block consists of un-pooling with kernel 2, convolutional layers with stride 1, side-in and side-output layers. The network gradually enlarges the feature resolution and finally performs 1×1 convolution to convert the 64-channel features to the output depth image. Since the caffe implementation of the un-pooling layer allows to specify its output feature resolution, it is convenient to handle input features with odd size. We then use it to enlarge 2D feature resolution by two times and add possible paddings to recover the down-sampled feature size. Inside each block, a skip connection is also designed to add the output of the un-pooling layer and the last convolutional layer, which facilitates the back propagation of gradient information. Please refer to Tab.1 for its detailed parameters.

The side-input layers are just several strided convolutional layers to generate corresponding features for each up-sampling block. Connecting them to the up-sampling blocks can supply image-level features to improve the details of the output depth images. In contrast, the side-output layers in our network are the input to the multi-scale loss function. It is used to control the behavior of the up-sampling blocks at different scales, similar to [Xie and Tu 2017].

3.2 Loss functions

Since it is difficult to obtain scale-correct depth values from single images, our training loss function is a combination of scale-invariant L2 and L1 loss and rank loss to regularize relative depth. We found such combination improved the prediction accuracy and speculated on two reasons for the improvement: 1) It is well known that the L2 norm can penalize high deviation from the data while L1 norm can tolerate outliers. After testing a large number of weights to combine this two loss, we can achieve a balance between these two norms on the depth regression problem. 2) The ranks loss allows us to pick arbitrarily two points on the image and specify their depth order, which can be viewed as a non-local loss to help the network to discover non-local depth order in the data.

Let us first denote one training RGBD image by I and its predicted depth image by Z. For a pixel i, its RGB value is denoted by I_i^c, depth value in the input RGBD image by I_i^d, and predicted depth value by Z_i. Also, according to [Eigen et al. 2014], the scale-invariant difference between the depth value and the predicted depth value at pixel i can be defined as:

$$D(I,i) = log(Z_i) - log\left(I_i^d\right) - \frac{1}{N}\sum_{j=1}^{N}\left(logZ_j - logI_j^d\right) \quad (1)$$

Similarly, we define scale-invariant depth gradient difference function G_x and G_y:

$$\begin{aligned} G_x(I,i) &= log\left(\nabla_x Z_i\right) - log\left(\nabla_x I_i^d\right) - \tfrac{1}{N}\sum_{j=1}^{N}\left(\nabla_x Z_j - \nabla_x I_j^d\right) \\ G_y(I,i) &= log\left(\nabla_y Z_i\right) - log\left(\nabla_y I_i^d\right) - \tfrac{1}{N}\sum_{j=1}^{N}\left(\nabla_y Z_j - \nabla_y I_j^d\right) \end{aligned} \quad (2)$$

L1 and L2 Loss: With $D(I,i), G_x(I,i), G_y(I,i)$, the L1 loss for an image I then is:

$$L_1(I) = \frac{1}{N}\sum_{i=0}^{N}\{|D(I,i)| + |G_x(I,i)| + |G_y(I,i)|\} \quad (3)$$

And the L2 loss can be written into:

$$L_2(I) = \frac{1}{N}\sum_{i=0}^{N}\{D(I,i)^2 + G_x(I,i)^2 + G_y(I,i)^2\} \quad (4)$$

Rank loss:In [Chen et al. 2016], the rank loss is used to train a CNN to predict relative depth of internet images. Its input includes randomly drawn pairs of selected pixels (i,j) with label $r_{ij} = \{+1,-1,0\}$. where r_{ij} encodes three relationships between Z_i and Z_j: $+1$ for $Z_i > Z_j$, -1 for $Z_i < Z_j$, and 0 for $Z_i = Z_j$. Then, The rank loss for a particular pair can be written into:

$$L_r(I,i,j) = \begin{cases} (Z_i - Z_j)^2 & r_{ij} = 0 \\ \log\left(1 + \exp\left(-r_{ij} * (Z_i - Z_j)\right)\right) & r_{ij} \neq 0 \end{cases} \quad (5)$$

To incorporate the rank loss into the training process, we also randomly sample pixel pairs at each training RGBD image and determine their depth orders according to its depth data. If the difference

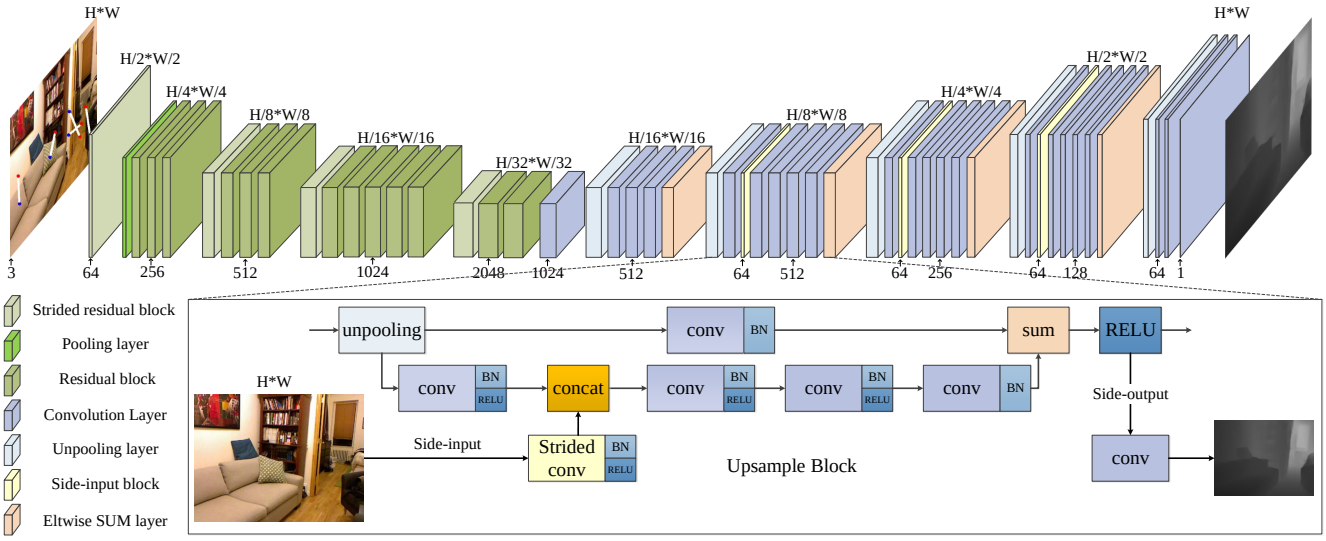

Figure 2: Our fully convolutional neural network for depth prediction. The down-sampling network (in green color) is same with ResNet50 [He et al. 2016] except that we use convolution layers to replace the average pooling layer and the fully connected layer. The pixel pairs marked on input image are used in the rank loss. The side-input layers connected to the up-sampling network are trained to supply features for depth edge enhancement. Note that the last up-sampling blocks do not have features from the side-input layer.

between a pair of depth values is less than a threshold 0.01, its label is set to be 0. Otherwise, it is set to be ± 1. The sampled pairs are the input to a rank loss layer in our caffe implementation.

The training loss: It is a weighted combination of L_1, L_2 and L_r, which is:

$$L_t = \sum_{j=0}^{M} L_1\left(I_j\right) + \alpha L_2\left(I_j\right) + \beta \frac{1}{K} \sum_{k,l} L_r\left(I_j, k, l\right) \tag{6}$$

where (k,l) denotes a pixel pair at a training image j, and K the number of sampled pairs.

The multi-scale L1 and L2 training loss: It is additional L1 and L2 loss between the side-output and the input depth image resized to the corresponding scale. Such loss values are used to further regularize the behavior of the network behavior at different up-sampling scales.

4 STEREO IMAGE GENERATION

The goal of stereo image generation is to synthesize image pairs to simulate the influence of viewpoint translation to the input color image. With the predicted depth, our system creates stereo images in two steps. First, the input image is warped to the target viewpoint using the disparity values computed from the depth. Our system generates two stereo images that are left and right to the input image along local X axis. Then, the large dis-occluded regions in the new images are restored via GAN-based image inpainting.

Disparity calculation: In our setting, the disparities of pixels on created stereo images are along the X direction, which is:

$$disparity = x - x' = \frac{B \times f}{Z} \tag{7}$$

where the new pixel position is denoted by x and its original position on the input image by x'. The baseline, i.e. the translation of the viewpoint chosen in the algorithm, is denoted by B. The camera focal length f is set according to the dataset, and Z is the depth at x'. Thus, the new image at the translated viewpoint can be easily achieved through forward image warping with the disparity information. The dis-occluded regions because of viewpoint translation form holes on the new image to be inpainted, as illustrated in Fig.3, and the holes are usually close to the boundary of the objects in the original image. The sparse, 1 pixel width holes not on the boundary are because of the discretization error of continuous depth values.

Foreground object protection: It extends the depth edges of foreground objects to match their color edges and then smooth the depth transition from foreground to background [Fieseler and Jiang 2009; Lu et al. 2012]. Suppose that the view right to the input RGB-D image is to be generated, the holes will then appear along the right side of foreground objects. In this case, the depth discontinuity between a foreground object and background can be identified by:

$$B_d\left(x,y\right) = \begin{cases} 1 & Z\left(x+1,y\right) - Z\left(x,y\right) > \theta \\ 0 & otherwise \end{cases} \tag{8}$$

where $Z\left(x,y\right)$ indicates the depth value at pixel $\left(x,y\right)$, and the value increases with its distance to the viewpoint. The binary function $B_d\left(x,y\right)$ is used to specify whether a depth discontinuity exists with value one.

The protection algorithm first determines the depth discontinuity with Eq.8, and then search the color edge in the pixels right to the depth edge. Once color edges detected, the depth of the foreground object is extended to the color edge so that the foreground object can be warped smoothly without tearing effect. To reduce the hole size to ease the inpainting, a narrow band is used to smooth the depth transition. Fig.4 illustrates the concept of the protection method.

GAN-based inpainting: We employ globally and locally consistent GAN to fill the holes after foreground object detection [Iizuka et al.

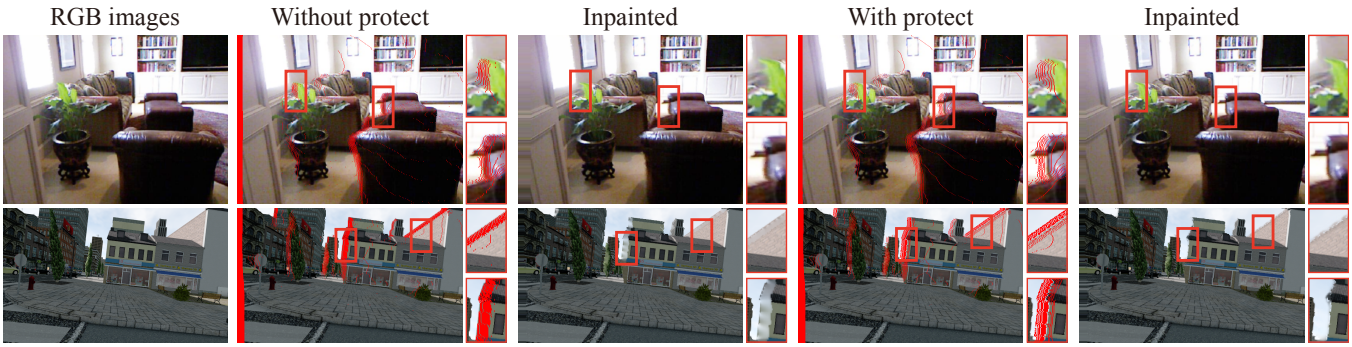

Figure 3: Examples of image warping using the predicted depth information. The dis-occluded areas are drawn in red, after moving the viewpoints at original images to the left. After foreground object protection, the object boundaries are better protected as shown the third and fifth columns.

Table 1: The up-sampling network Details

Layer name	*type*	*kernel size*	*stride*	*outputs*
res_down	Convolution	1	1	1024
unpool5	Unpooling	2	2	1024
up5_branch1	Convolution	5	1	512
up5_branch2a	Convolution	5	1	512
up5_branch2b	Convolution	5	1	512
up5	Eltwise Sum	-	-	512
side-output5	Convolution	5	1	1
unpool4	Unpooling	2	2	512
up4_branch1	Convolution	5	1	512
up4_branch2a	Convolution	5	1	512
up4_branch2b	Convolution	5	1	512
up4_branch2c	Convolution	5	1	512
up4_branch2d	Convolution	5	1	512
up4	Eltwise Sum	-	-	512
side-output4	Convolution	5	1	1
unpool3	Unpooling	2	2	512
up3_branch1	Convolution	5	1	256
up3_branch2a	Convolution	5	1	256
up3_branch2b	Convolution	5	1	256
up3_branch2c	Convolution	5	1	256
up3_branch2d	Convolution	5	1	256
up3	Eltwise Sum	-	-	256
side-output3	Convolution	5	1	1
unpool2	Unpooling	2	2	256
up2_branch1	Convolution	5	1	128
up2_branch2a	Convolution	5	1	128
up2_branch2b	Convolution	5	1	128
up2_branch2c	Convolution	5	1	128
up2_branch2d	Convolution	5	1	128
up2	Eltwise Sum	-	-	128
side-output2	Convolution	5	1	1
unpool1	Unpooling	2	2	128
up1_conv1	Convolution	5	1	64
up1_conv2	Convolution	5	1	64
depth	Convolution	5	1	1

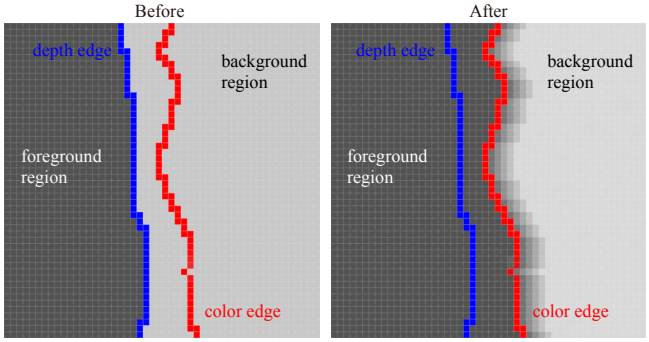

Figure 4: Illustration of foreground protection. The mismatched depth and color edges in the left image are corrected and smoothed as shown in the right image. Note that foreground object protection can not eliminate the edge distortion due to depth error.

2017]. Since the width of hole size in our case is usually less than 20 pixels, we thus train the network on RGB images in the RGB-D dataset using random squared holes of size $1-32$, which is smaller than the hole size $96-128$ in the original paper. In addition, if the hole is of only 1-2 pixel width, we directly fill it using the interpolated color from its neighbour, and the rest large holes are then inpainted by the trained GAN.

5 EXPERIMENTS

We train the depth prediction network on a PC with two NVIDIA 1080Ti GPUs and 12GB memory. The parameters of ResNet50 network are from its pre-trained parameters on ImageNet[Deng et al. 2009], while other layers are initialized randomly using Gaussian distributions. We employ SGD method to train the network using a batch size of 8 RGB-D images and randomly sampled 3000 pixel pairs for each image. The entire training process lasts for 12 epochs. We warm-up the training by setting the learning rate to be 0.001 for several thousand iterations. Afterwards, the learning rate is changed to 0.01 to start the formal training process, and it is decreased to

110 every 5 epochs. In the following, we report the details of the dataset and the evaluation of the depth-prediction network and the inpainting results.

5.1 Dataset

The depth prediction network is trained on two RGB-D image datasets which cover the indoor and outdoor scenes.

NYU Depth Dataset: NYU Depth v2 is an indoor scene RGBD dataset captured by Microsoft Kinect [Nathan Silberman and Fergus 2012]. For training, we sample raw Kinect frames with equal interval on its 249 training scenes and get 40K unique RGB-D images. After data augmentation of horizontally flip, rotation and scale following [Eigen et al. 2014], the final training set has 120K samples. Each image and corresponding depth map are downsampled to 320×240. The missing values and invalid regions are all masked out during training. To combine with relative depths, we sample 3000 pixel pairs on the valid ground truth depth maps randomly and mark their relations. The predicted values of the network are mapped back to the scale of 10 meters by exponential function. All the evaluations are applied on the commonly-used test subset of 654 RGB-depth image pairs with the invalid pixels in ground truth filled.

SYNTHIA Dataset: SYNTHIA is a synthetic collection of outdoor driving images [Ros et al. 2016]. The old European town scene SEQS-04 and highway scenario SEQS-06 of all seasons are used. All the samples of left and right images are all treated as irrelevant images. We split all the images to train and test set in proportion of 9:1 and get 54K training pairs and 6K testing pairs. Each pair is downsampled to $320 * 190$ with bilinear interpolation. Random pixel pairs are also sampled on ground truth depth maps.

5.2 Evaluation of Depth Prediction Results

Comparison with base-line algorithms: We first compare our results with state-of-the-art depth prediction methods with open-sourced implementation using deep convolutional neural network:

- The multi-scale convolutional neural network for depth prediction in [Eigen and Fergus 2015].
- The sequential deep network with multi-scale continuous conditional random fields(CRFs) for monocular depth estimation in [Xu et al. 2017].
- The fully convolutional residual network for depth prediction in [Laina et al. 2016]

The reconstructed depth images are visualized as a gray-scale image, where darker color indicates a closer distance to the viewpoint. As shown in Fig. 1 and 5, the depth edges are better reconstructed by our network. Although multi-scale features are also considered in base-line algorithms, the side-input layer approach is empirically more effective. In Tab. 2, our algorithm achieves best structural similarity (SSIM) score on test images of NYU indoor v2 dataset (Since the depth regression network in [Kendall and Gal 2017] is not open sourced, its SSIM score is skipped in our evaluation). We also note that the high quality of depth edges are important to the quality of the warped image in the simulation of viewpoint translation (see Fig.10).

Tab. 2 lists the mean error and accuracy statistics using the standard metrics as described in [Eigen et al. 2014]. Our algorithm achieves highest prediction accuracy 82.1% with threshold 1.25. It also shows that the integration of rank loss improves the prediction accuracy, since the confusion of local features can be effectively regularized by non-local relationship considered in rank-loss. However, the mean error of our method is not so impressive. We speculate that it is because of the uncertainty of the depth values around object boundaries. As shown in the third row of Fig. 5, there exist mismatched depth edges between our result and the ground truth image when the scene is far from the viewpoint. The depth difference there should be large. It is also an example to show the ambiguity of depth prediction from RGB features.

Fig. 6 illustrates the comparison with the Bayesian deep learning approach in [Kendall and Gal 2017]. Since the implementation of their network is not publicly available, we directly compare with the available depth prediction results in their paper. Since the approach focuses on the modelling of the uncertainties of data and network weights, it reduces the mean error through attenuation while the depth edges is still more blurred comparing to our results.

Comparison to relative depth learning in [Chen et al. 2016]: Fig. 7 illustrates comparison with relative depth prediction using rank loss in [Chen et al. 2016]. The depth prediction results is obtained by the publicly available implementation of their network. Since their training loss is only considered for sampled pixel pairs, there might be missing details in the reconstructed depth. For example, the depth of paper close to the viewpoint is missing, while it is well captured by our network.

Side-input: We assemble side-input layers in upsample residual blocks to restore details. Fig.8 shows the comparison of reconstructed depth with/without side-input layers. The depth edges are blurred in the results obtained by network without side-input layers. In contrast, they are much sharper after the concatenation of features learned through the side-input layers.

Training on SYNTHIA dataset: It is used to test our network on different type of dataset. After training, the accuracy within threshold 1.25,1.25^2,1.25^3 are 83.6%,92.3%,95.5%. The depth prediction results is presented in Fig. 9. The SYNTHIA dataset is relatively new and it still contains erroneous depth values, especially for trees in the scene. Also, the object depth close to the viewpoint is suppressed. It leads to unsatisfactory accuracy number with large threshold.

5.3 Evaluation of Stereo Image Generation

GAN training: The GAN in [Iizuka et al. 2017] is also trained on the NYU and SYNTHIA datasets. We use the same network structure and feed the network with 256×256 image cropped from the RGB images in the dataset, which results in $120,000$ training images from NYU dataset and $50,000$ images from SYNTHIA dataset. The hole size is chosen to be between $1-32$. The network is trained similarly in [Iizuka et al. 2017]. First, the generator is trained with batch size 32 and 10 epochs, and then the discriminator network is trained with batch size 96 and 5 epochs. Finally, both the generator and discriminator networks are trained alternatively until the convergence with batch size 32 and 12 epochs. We use Adadelta algorithm by setting its parameter to be $lr = 1.0$ and $rho = 0.9$.

Image warping results: Fig. 10 shows the image warping results using the disparity computed from the predicted depth. For indoor

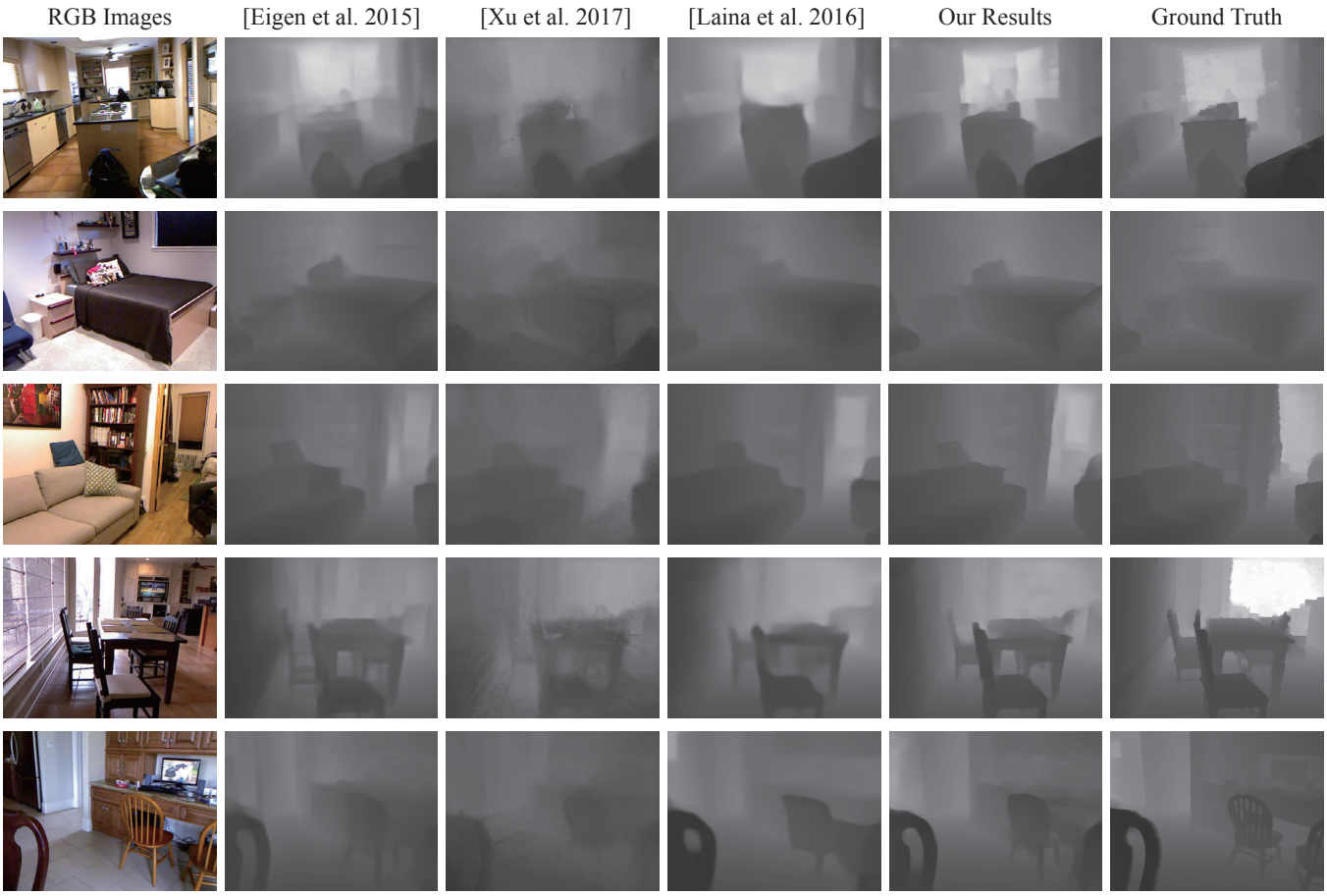

Figure 5: Results comparison with previous works. From left to right:rgb images,[Eigen and Fergus 2015],[Xu et al. 2017],[Laina et al. 2016],our results and ground truth depth map.

Method	SSIM	Error			Accuracy with threshold		
		rel	rms	$\log_{10}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
[Eigen et al. 2014]	0.914	0.215	0.907	-	61.1%	88.7%	97.1%
[Liu et al. 2015]	-	0.230	0.824	0.095	61.4%	88.3%	97.1%
[Eigen and Fergus 2015]	0.931	0.158	0.641	-	76.9%	95.0%	98.8%
[Laina et al. 2016]	0.935	0.127	0.573	0.055	81.1%	95.3%	98.8%
[Xu et al. 2017]	0.914	0.121	0.586	0.052	81.1%	95.4%	98.7%
[Kendall and Gal 2017]	-	0.110	0.506	0.045	81.7%	95.9%	98.9%
Our method(without side input)	0.936	0.157	0.650	0.068	76.8%	95.4%	98.9%
Our method(without rank loss)	0.940	0.143	0.669	0.063	79.8%	95.6%	98.8%
Our method	0.946	0.132	0.598	0.057	82.2%	95.7%	98.8%

Table 2: Comparisons with baseline algorithms on NYU depth v2 dataset.

scene images, the baseline between viewpoints is set to be around two times of the eye distance to reveal the warping effect. The top 3 rows show the images warped to the right view, which corresponds to the three depth predicton results in Fig. 1. It can be seen that the artefacts in the warped image using the depth from three baseline algorithms can be eliminated or greatly reduced using our depth prediction results. As shown in the first row, the back of the chairs is well protected as in the image warped using ground truth depth information.

Stereo image pair generation results: In Fig. 11, we show four generated image pairs. The top 2 rows are results from the test image in our datasets, and the bottom 2 rows shows the results of our

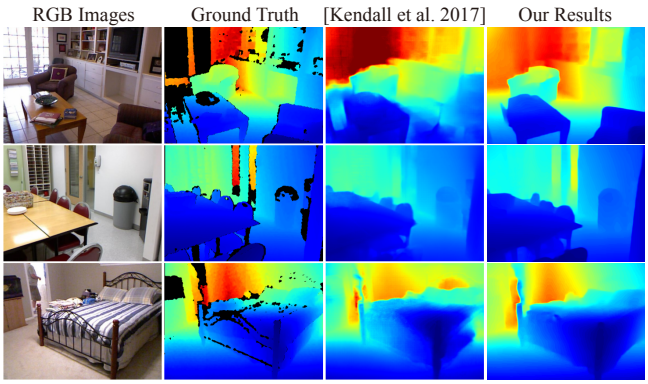

Figure 6: Comparison with bayesian deep learning approach in [Kendall and Gal 2017]. The attenuation of prediction at pixels with high uncertainty does not prevent the blurring at the depth edges.

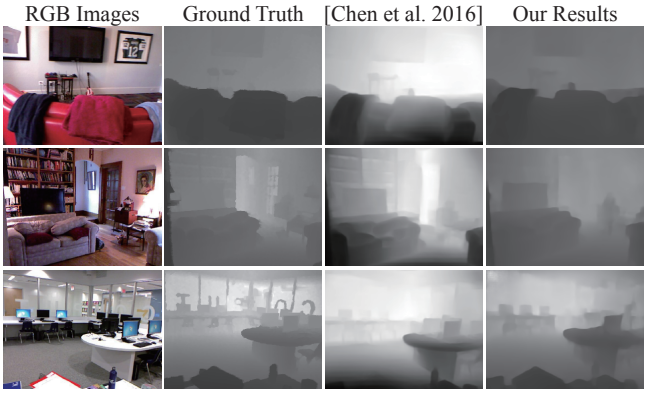

Figure 7: Comparison with depth perception in the wild method in [Chen et al. 2016]. The errors in results from [Chen et al. 2016] are reduced, and the depth if paper close the viewpoint on the last row missed by [Chen et al. 2016] is captured by our network.

pipeline applied to images from the internet. The depth transition region in foreground protection is selected to be 3 and 5 pixels respectively. The red boxes in the images are used to show parallax effects. In addition, we produce ?? videos by generating in-between images to further demonstrate the effectiveness of our pipeline, and they are included in the supplementary materials. A comparison on image inpainting is shown in Fig. 12. We use 9×9 patch size to obtain the result from patch-based image inpainting in [Daribo and Pesquet-Popescu 2010]. However, it distorts the edge, since the order of patches selected in the inpainting is vulnerable. In contrast, the trained GAN recovers the edge. The application of GAN to images not in the dataset is also shown in the supplemented videos.

User study:

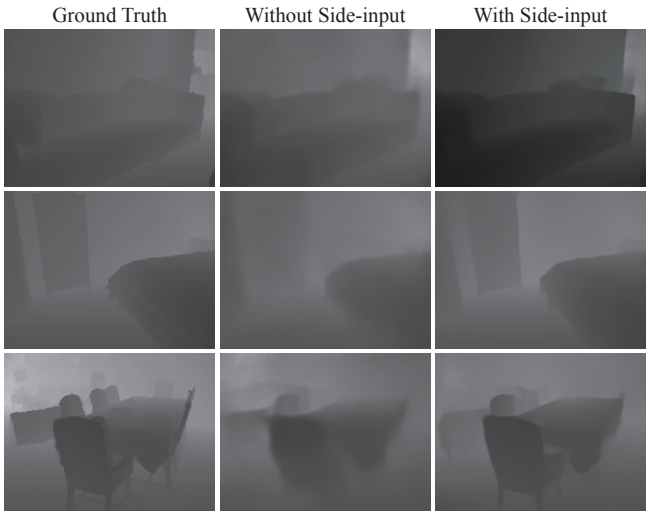

Figure 8: Comparison of depth prediction results with/without side-input layers. The blurred depth edges in the results obtained without side-input layers are much sharper after using side-input layers.

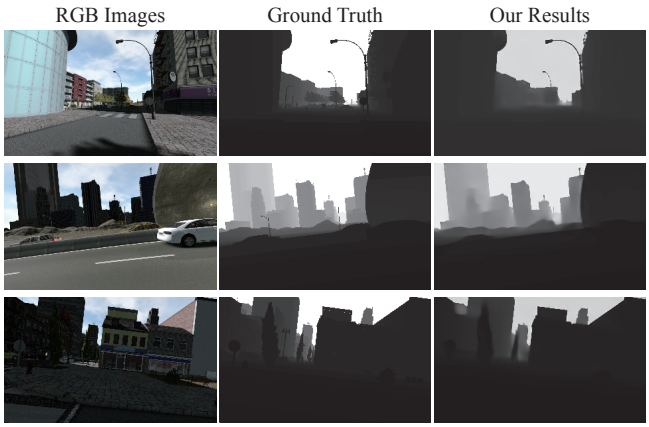

Figure 9: SYNTHIA dataset.

6 CONCLUSION AND FUTURE WORK

We have developed a deep-learning based pipeline for the stereoscopic viewing of RGB images. It consists of two steps. First, a fully convolutional neural network with side-input layers is employed to predict the depth information from an input image. The network for depth prediction is trained on NYU indoor scene and SYNTHIA outdoor scene RGB-D image datasets. Second, the input image is warped to the new viewpoint using the disparity information computed via the predicted depth. Afterwards, the dis-occluded areas are inpainted via a globally and locally consistent generative adversarial network. We also tested the pipeline on the images not in the dataset to verify its effectiveness.

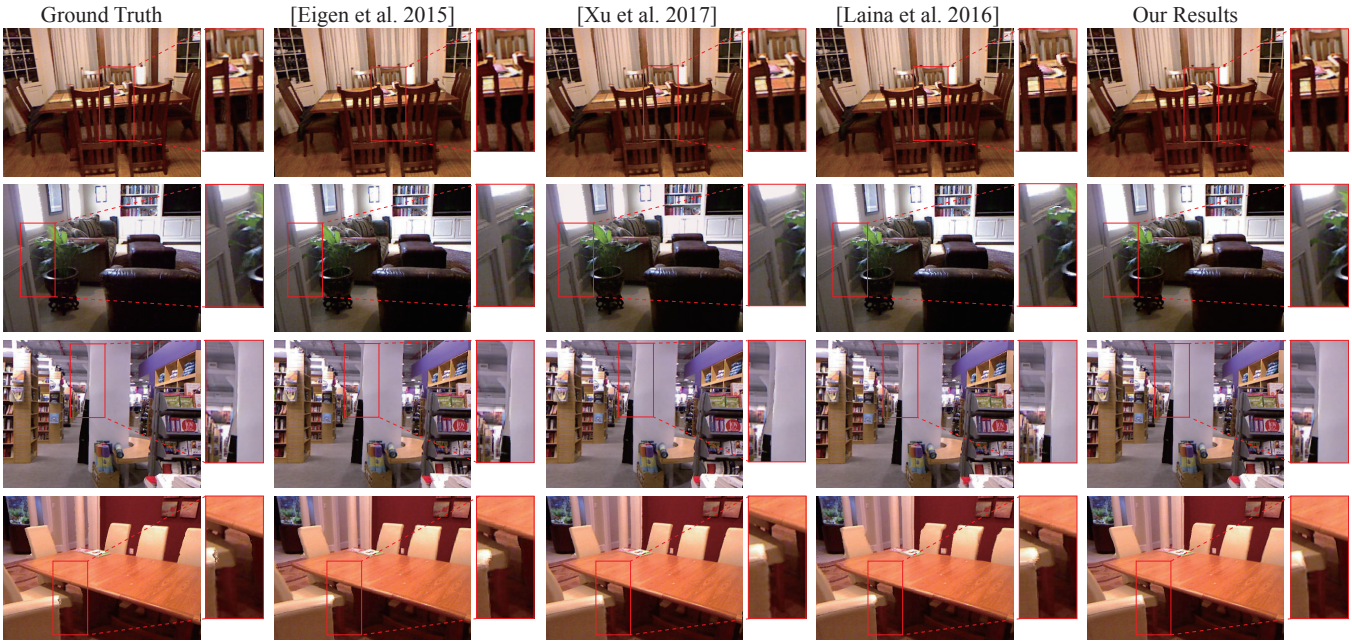

Figure 10: Disparity-based image warping results. The images are warped using the depth from (from left to right): ground truth, [Eigen and Fergus 2015], [Xu et al. 2017], [Laina et al. 2016], our results. The object distortion artefacts in the warped images using baseline depth prediction results are removed or significantly reduced. The distortion at the sofa handle using ground truth depth (see the third row) is because of the interpolated depth in NYU indoor v2 dataset.

Limitation and future work: Although the depth prediction network can produce good quality depth images, there still exist prediction errors which are difficult to be completely removed through the pre-training. It is desirable to train a network that can incorporate user interaction to correct such errors. The reconstructed depth image is of 320×240, which is not enough for applications. The network should also be extended to support high resolution depth reconstruction to synthesize high resolution stereo image pairs for good user experiences. In addition, since the dis-occluded areas are close to object boundaries and the areas should be mainly filled by background information, it is interesting to investigate how non-symmetric convolution kernel can help to improve the inpainting results in the stereoscopy applications.

REFERENCES

Bumblebee2 1394a. 2017. https://www.ptgrey.com/bumblebee2-firewire-stereo-vision-camera-systems?countryid=237. (2017).

Robert Anderson, David Gallup, Jonathan T. Barron, Janne Kontkanen, Noah Snavely, Carlos Hernández, Sameer Agarwal, and Steven M. Seitz. 2016. Jump: Virtual Reality Video. *ACM Trans. Graph.* 35, 6, Article 198 (2016), 13 pages.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. *CoRR* abs/1701.07875 (2017).

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. 2009. PatchMatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* 28, 3 (2009).

Marcelo Bertalmio, Guillermo Sapiro, Vicent Caselles, and Coloma Ballester. 2000. Image inpainting. ACM Trans. Graph., 417–424.

Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2001. Unstructured Lumigraph Rendering. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. ACM, New York, NY, USA, 425–432.

ZED Stereo Camera. 2017. https://www.stereolabs.com/. (2017).

Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. 2013. Depth Synthesis and Local Warps for Plausible Image-based Navigation. *ACM Trans. Graph.* 32, 3, Article 30 (July 2013), 12 pages.

Gaurav Chaurasia, Olga Sorkine, and George Drettakis. 2011. Silhouette-Aware Warping for Image-Based Rendering. *Comput. Graph. Forum* 30, 4 (2011), 1223–1232.

Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. 2016. Single-Image Depth Perception in the Wild.. In *NIPS*. 730–738.

Chia-Ming Cheng, Shu-Jyuan Lin, Shang-Hong Lai, and Jinn-Cherng Yang. 2008. Improved novel view synthesis from depth image with large baseline.. In *ICPR*. 1–4.

Ismaël Daribo and Béatrice Pesquet-Popescu. 2010. Depth-aided image inpainting for novel view synthesis.. In *MMSP*. IEEE, 167–170.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

David Eigen and Rob Fergus. 2015. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture.. In *ICCV*. IEEE Computer Society, 2650–2658.

David Eigen, Christian Puhrsch, and Rob Fergus. 2014. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. *CoRR* abs/1406.2283 (2014).

Facebook. 2017. https://facebook360.fb.com/facebook. (2017).

Michael Fieseler and Xiaoyi Jiang. 2009. Registration of depth and video data in depth image based rendering. In *2009 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*. 1–4.

John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. 2016. Deep Stereo: Learning to Predict New Views from the World's Imagery.. In *CVPR*. IEEE Computer Society, 5515–5524.

Michael Goesele, Jens Ackermann, Simon Fuhrmann, Carsten Haubold, Ronny Klowsky, Drew Steedly, and Richard Szeliski. 2010. Ambient Point Clouds for View Interpolation. *ACM Trans. Graph.* 29, 4, Article 95 (July 2010), 6 pages.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets.. In *NIPS*. 2672–2680.

Xiao han Lu, Fang Wei, and Fang min Chen. 2012. Foreground-Object-Protected Depth Map Smoothing for DIBR.. In *ICME*. IEEE Computer Society, 339–343.

James Hays and Alexei A. Efros. 2008. Scene completion using millions of photographs. *Commun. ACM* 51, 10 (2008), 87–94.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. IEEE Computer Society, 770–778.

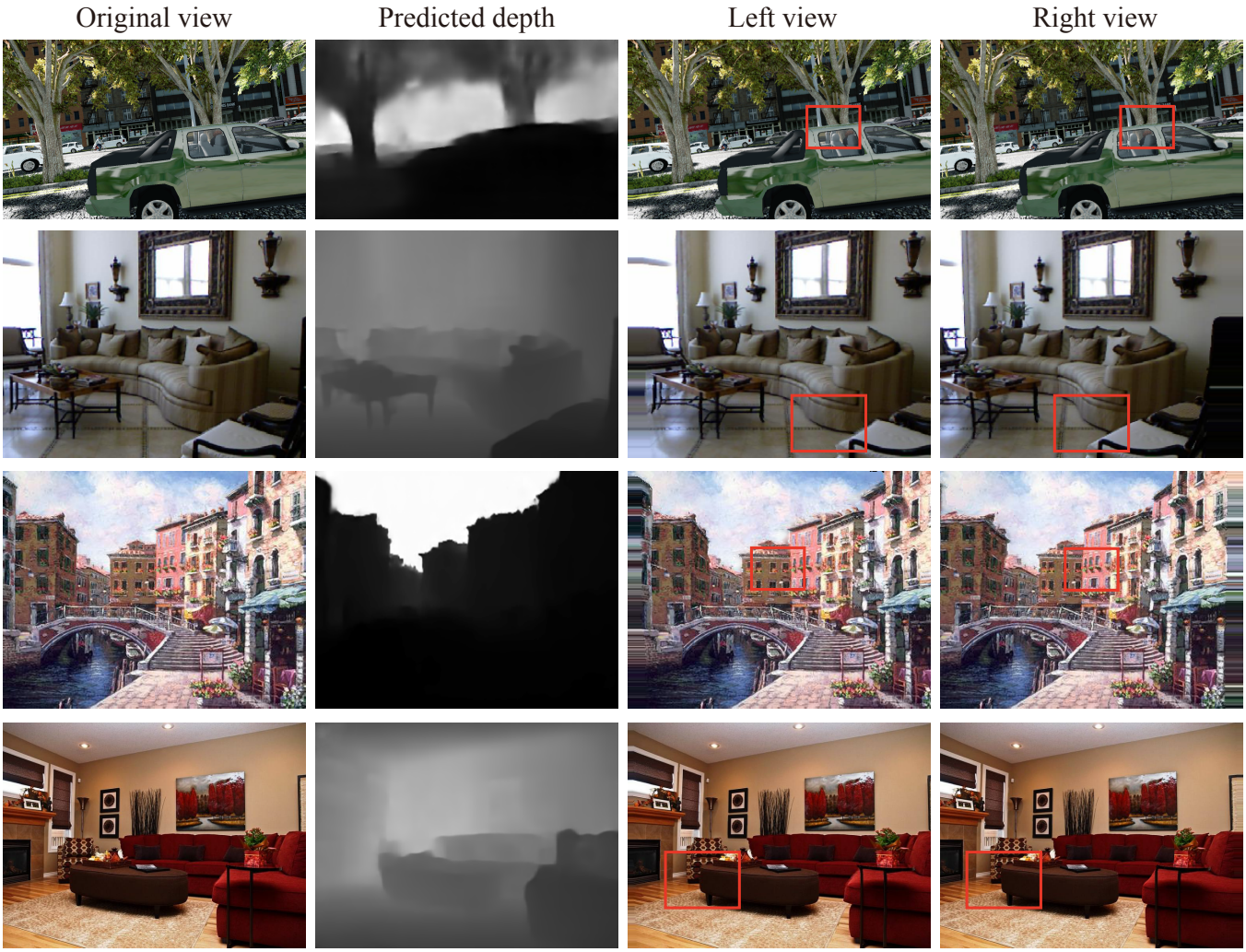

Figure 11: Illustration of stereo image generation results. We test the pipeline on both test images in the dataset on the first two rows and internet images on the last two rows. The red rectangles indicate the parallax effect in the generated image pairs. More results can be found in the supplementary material.

Figure 12: GAN-based vs. Patch-based image inpainting. The red lines in RGB images show the depth edge along the cabinet. The edges are well inpainted in the hole by GAN based inpainting in [Iizuka et al. 2017], but distorted by the patch-based inpainting method in [Daribo and Pesquet-Popescu 2010].

Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. 2014. Image completion using planar structure guidance. *ACM Trans. Graph.* 33, 4 (2014), 129:1–129:10.

Qixing Huang, Hai Wang, and Vladlen Koltun. 2015. Single-view reconstruction via joint analysis of image and shape collections. *ACM Trans. Graph.* 34, 4 (2015), 87.

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and locally consistent image completion. *ACM Trans. Graph.* 36, 4 (2017), 107:1–107:14.

Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. 2016. Learning-based View Synthesis for Light Field Cameras. *ACM Trans. Graph.* 35, 6, Article 193 (Nov. 2016), 10 pages.

Kevin Karsch, Ce Liu, and Sing Bing Kang. 2014. Depth Transfer: Depth Extraction from Video Using Non-Parametric Sampling. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 11 (2014), 2144–2158.

Alex Kendall and Yarin Gal. 2017. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?. In *NIPS*. 5580–5590.

Rolf Köhler, Christian J. Schuler, Bernhard Schölkopf, and Stefan Harmeling. 2014. Mask-Specific Inpainting with Deep Neural Networks.. In *GCPR (Lecture Notes in Computer Science)*, Vol. 8753. Springer, 523–534.

Martin Köppel, Karsten Müller, and Thomas Wiegand. 2016. Filling Disocclusions in Extrapolated Virtual Views Using Hybrid Texture Synthesis. *TBC* 62, 2 (2016), 457–469.

Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. 2016. Deeper Depth Prediction with Fully Convolutional Residual Networks. In *3DV*. IEEE Computer Society, 239–248.

DooHyun Lee and InSo Kweon. 2000. A novel stereo camera system by a biprism. *IEEE Trans. Robotics and Automation* 16, 5 (2000), 528–541.

Marc Levoy and Pat Hanrahan. 1996. Light Field Rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. ACM, 31–42.

Fayao Liu, Chunhua Shen, and Guosheng Lin. 2015. Deep convolutional neural fields for depth estimation from a single image.. In *CVPR*. IEEE Computer Society, 5162–5170.

Xiaohan Lu, Fang Wei, and Fangmin Chen. 2012. Foreground-Object-Protected Depth Map Smoothing for DIBR. In *2012 IEEE International Conference on Multimedia and Expo*. 339–343.

Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, and Zhen Wang. 2016. Least Squares Generative Adversarial Networks. (2016). http://arxiv.org/abs/1611.04076

Youssef Mroueh and Tom Sercu. 2017. Fisher GAN. (2017). http://arxiv.org/abs/1705.09675

Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. 2012. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*.

Patrick Ndjiki-Nya, Martin Koppel, Dimitar Doshkov, Haricharan Lakshman, Philipp Merkle, Karsten Muller, and Thomas Wiegand. 2011. Depth Image-Based Rendering With Advanced Texture Synthesis for 3-D Video. *IEEE Trans. Multimedia* 13, 3 (2011), 453–465.

Nokia. 2017. https://ozo.nokia.com/. (2017).

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. 2016. Context Encoders: Feature Learning by Inpainting. *CoRR* abs/1604.07379 (2016).

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *CoRR* abs/1511.06434 (2015).

S. Ravi, P. Pasupathi, S. Muthukumar., and N. Krishnan. 2013. Image in-painting techniques - A survey and analysis. In *2013 9th International Conference on Innovations in Information Technology (IIT)*. 36–41.

Jimmy S. J. Ren, Li Xu, Qiong Yan, and Wenxiu Sun. 2015. Shepard Convolutional Neural Networks.. In *NIPS*. 901–909. http://dblp.uni-trier.de/db/conf/nips/nips2015.html#RenXYS15

German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. 2016. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

A. Saxena, Min Sun, and Andrew.Y. Ng. 2009. Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (2009), 824–840.

Evan Shelhamer, Jonathan Long, and Trevor Darrell. 2017. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 4 (2017), 640–651.

Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. 2005. Image completion with structure propagation. *ACM Trans. Graph.* 24, 3 (2005), 861–868.

Junyuan Xie, Ross B. Girshick, and Ali Farhadi. 2016. Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks.. In *ECCV (4) (Lecture Notes in Computer Science)*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.), Vol. 9908. 842–857.

Junyuan Xie, Linli Xu, and Enhong Chen. 2012. Image Denoising and Inpainting with Deep Neural Networks.. In *NIPS*. 350–358.

Saining Xie and Zhuowen Tu. 2017. Holistically-Nested Edge Detection. *International Journal of Computer Vision* 125, 1-3 (2017), 3–18.

Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. 2017. Multi-scale Continuous CRFs as Sequential Deep Networks for Monocular Depth Estimation.. In *CVPR*. IEEE Computer Society, 161–169.

Qiong Zeng, Wenzheng Chen, Huan Wang, Changhe Tu, Daniel Cohen-Or, Dani Lischinski, and Baoquan Chen. 2015. Hallucinating Stereoscopy from a Single Image. *Comput. Graph. Forum* 34, 2 (2015), 1–12.

Liang Zhang and Wa James Tam. 2005. Stereoscopic image generation based on depth images for 3D TV. *TBC* 51, 2 (2005), 191–199.

Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. 2016. View Synthesis by Appearance Flow.. In *ECCV (4) (Lecture Notes in Computer Science)*, Vol. 9908. Springer, 286–301.