

Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks

Junyuan Xie¹(✉), Ross Girshick¹, and Ali Farhadi^{1,2}

¹ University of Washington, Seattle, USA
{jxie,rbg,ali}@cs.washington.edu

² Allen Institute for Artificial Intelligence, Seattle, USA

Abstract. As 3D movie viewing becomes mainstream and the Virtual Reality (VR) market emerges, the demand for 3D contents is growing rapidly. Producing 3D videos, however, remains challenging. In this paper we propose to use deep neural networks to automatically convert 2D videos and images to a stereoscopic 3D format. In contrast to previous automatic 2D-to-3D conversion algorithms, which have separate stages and need ground truth depth map as supervision, our approach is trained end-to-end directly on stereo pairs extracted from existing 3D movies. This novel training scheme makes it possible to exploit orders of magnitude more data and significantly increases performance. Indeed, Deep3D outperforms baselines in both quantitative and human subject evaluations.

Keywords: Monocular stereo reconstruction · Deep convolutional neural networks

1 Introduction

3D movies are popular and comprise a large segment of the movie theater market, ranging between 14 % and 21 % of all box office sales between 2010 and 2014 in the U.S. and Canada [1]. Moreover, the emerging market of Virtual Reality (VR) head-mounted displays will likely drive an increased demand for 3D content (Fig. 1).

3D videos and images are usually stored in stereoscopic format. For each frame, the format includes two projections of the same scene, one of which is exposed to the viewer’s left eye and the other to the viewer’s right eye, thus giving the viewer the experience of seeing the scene in three dimensions.

There are two approaches to making 3D movies: shooting natively in 3D or converting to 3D after shooting in 2D. Shooting in 3D requires costly special-purpose stereo camera rigs. Aside from equipment costs, there are cinematographic

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46493-0_51) contains supplementary material, which is available to authorized users.

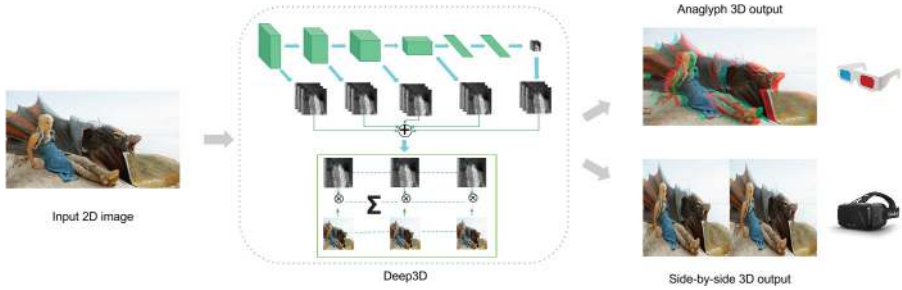


Fig. 1. We propose Deep3D, a fully automatic 2D-to-3D conversion algorithm that takes 2D images or video frames as input and outputs stereo 3D image pairs. The stereo images can be viewed with 3D glasses or head-mounted VR displays. Deep3D is trained directly on stereo pairs from a dataset of 3D movies to minimize the pixel-wise reconstruction error of the right view when given the left view. Internally, the Deep3D network estimates a probabilistic disparity map that is used by a differentiable depth image-based rendering layer to produce the right view. Thus Deep3D does not require collecting depth sensor data for supervision.

issues that may preclude the use of stereo camera rigs. For example, some inexpensive optical special effects, such as forced perspective¹, are not compatible with multi-view capturing devices. 2D-to-3D conversion offers an alternative to filming in 3D. Professional conversion processes typically rely on “depth artists” who manually create a depth map for each frame. Standard Depth Image-Based Rendering (DIBR) algorithms can then be used to combine the original frame with the depth map in order to arrive at a stereo image pair [2]. However, this process is still expensive as it requires intensive human effort.

Each year about 20 new 3D movies are produced. High production cost is the main hurdle in the way of scaling up the 3D movie industry. Automated 2D-to-3D conversion would eliminate this obstacle.

In this paper, we propose a fully automated, data-driven approach to the problem of 2D-to-3D video conversion. Solving this problem entails reasoning about depth from a single image and synthesizing a novel view for the other eye. Inferring depth (or disparity) from a single image, however, is a highly under-constrained problem. In addition to depth ambiguities, some pixels in the novel view correspond to geometry that’s not visible in the available view, which causes missing data that must be hallucinated with an in-painting algorithm.

In spite of these difficulties, our intuition is that given the vast number of stereo-frame pairs that exist in already-produced 3D movies it should be possible to train a machine learning model to predict the novel view from the given view. To that end, we design a deep neural network that takes as input the left eye’s view, internally estimates a soft (probabilistic) disparity map, and then renders

¹ Forced perspective is an optical illusion technique that makes objects appear larger or smaller than they really are. It breaks down when viewed from another angle, which prevents stereo filming.

a novel image for the right eye. We train our model end-to-end on ground-truth stereo-frame pairs with the objective of directly predicting one view from the other. The internal disparity-like map produced by the network is computed only in service of creating a good right eye view and is not intended to be an accurate map of depth or disparity. We show that this approach is easier to train for than the alternative of using a stereo algorithm to derive a disparity map, training the model to predict disparity explicitly, and then using the predicted disparity to render the new image. Our model also performs in-painting implicitly without the need for post-processing.

Evaluating the quality of the 3D scene generated from the left view is non-trivial. For quantitative evaluations, we use a dataset of 3D movies and report pixel-wise metrics comparing the reconstructed right view and the ground-truth right view. We also conduct human subject experiments to show the effectiveness of our solution. We compare our method with the ground-truth and baselines that use state-of-the-art single view depth estimation techniques. Our quantitative and qualitative analyses demonstrate the benefits of our solution.

2 Related Work

Most existing automatic 2D-to-3D conversion pipelines can be roughly divided into two stages. First, a depth map is estimated from an image of the input view, then a DIBR algorithm combines the depth map with the input view to generate the missing view of a stereo pair. Early attempts to estimate depth from a single image utilize various hand engineered features and cues including defocus, scattering, and texture gradients [3, 4]. These methods only rely on one cue. As a result, they perform best in restricted situations where the particular cue is present. In contrast, humans perceive depth by seamlessly combining information from multiple sources.

More recent research has moved to learning-based methods [5–9]. These approaches take single-view 2D images and their depth maps as supervision and try to learn a mapping from 2D image to depth map. Learning-based methods combine multiple cues and have better generalization, such as recent works that use deep convolutional neural networks (DCNNs) to advance the state-of-the-art for this problem [10, 11]. However, collecting high quality image-depth pairs is difficult, expensive, and subject to sensor-dependent constraints. As a result, existing depth data set mainly consists of a small number of static indoor and, less commonly, outdoor scenes [12, 13]. The lack of volume and variations in these datasets limits the generality of learning-based methods. Moreover, the depth maps produced by these methods are only an intermediate representation and a separate DIBR step is still needed to generate the final result.

Monocular depth prediction is challenging and we conjecture that performing that task accurately is unnecessary. Motivated by the recent trend towards training end-to-end differentiable systems [14, 15], we propose a method that requires stereo pairs for training and learns to directly predict the right view from the left view. In our approach, DIBR is implemented using an internal probabilistic

disparity representation, and while it learns something akin to a disparity map the system is allowed to use that internal representation as it likes in service of predicting the novel view. This flexibility allows the algorithm to naturally handle in-painting. Unlike 2D image/depth map pairs, there is a vast amount of training data available to our approach since roughly 10 to 20 3D movies have been produced each year since 2008 and each has hundreds of thousands of frames.

Our model is inspired by Flynn et al.’s DeepStereo approach [16], in which they propose to use a probabilistic selection layer to model the rendering process in a differentiable way so that it can be trained together with a DCNN. Specifically we use the same probabilistic selection layer, but improve upon their approach in two significant ways. First, their formulation requires two or more calibrated views in order to synthesize a novel view—a restriction that makes it impossible to train from existing 3D movies. We remove this limitation by restructuring the network input and layout. Second, their method works on small patches (28×28 pixels) which limits the network’s receptive field to local structures. Our approach processes the entire image, allowing large receptive fields that are necessary to take advantage of high-level abstractions and regularities, such as the fact that large people tend to appear close to the camera while small people tend to be far away.

3 Method

Previous work on 2D-to-3D conversion usually consists of two steps: estimating an accurate depth map from the left view and rendering the right view with a Depth Image-Based Rendering (DIBR) algorithm. Instead, we propose to directly regress on the right view with a pixel-wise loss. Naively following this approach, however, leads to poor results because it does not capture the structure of the task (see Sect. 5.4). Inspired by previous work, we utilize a DIBR process to capture the fact that most output pixels are shifted copies of input pixels. However, unlike previous work we don’t constrain the system to produce an accurate depth map, nor do we require depth maps as supervision for training. Instead, we propose a model that predicts a probabilistic disparity-like map as an intermediate output and combines it with the input view using a differentiable selection layer that models the DIBR process. During training, the disparity-like map produced by the model is never directly compared to a true disparity map and it ends up serving the dual purposes of representing horizontal disparity and performing in-painting. Our model can be trained end-to-end thanks to the differentiable selection layer [16].

3.1 Model Architecture

Recent research has shown that incorporating lower level features benefits pixel wise prediction tasks including semantic segmentation, depth estimation, and optical flow estimation [10, 17]. Given the similarity between our task and depth estimation, it is natural to incorporate this idea. Our network, as shown in Fig. 2,

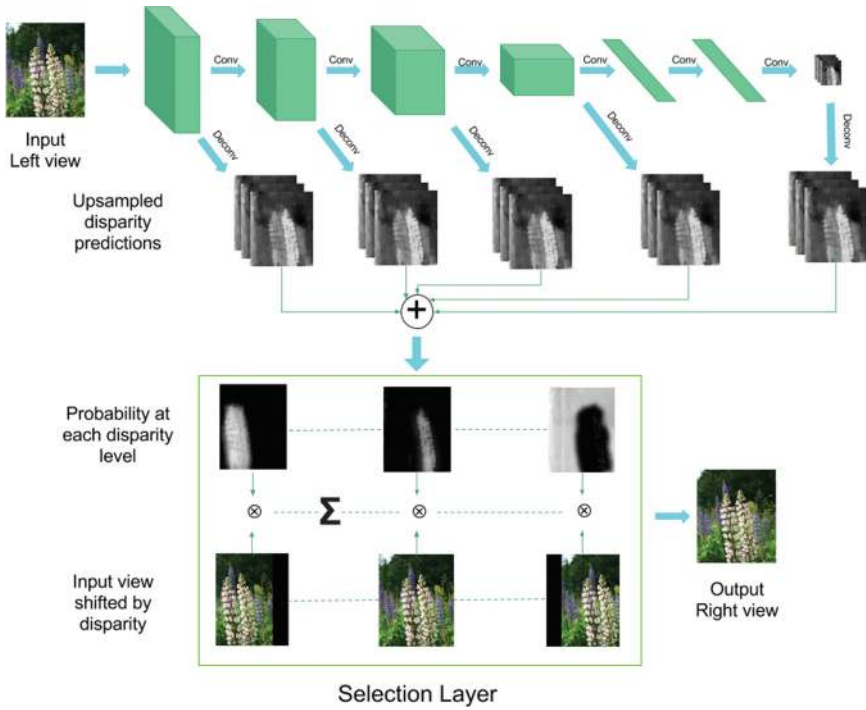


Fig. 2. Deep3D model architecture. Our model combines information from multiple levels and is trained end-to-end to directly generate the right view from the left view. The base network predicts a probabilistic disparity-like map which is then used by the selection layer to model Depth Image-Based Rendering (DIBR) in a differentiable way. This also allows implicit in-painting. (Color figure online)

has a branch after each pooling layer that upsamples the incoming feature maps using so-called “deconvolution” layers (i.e., a learned upsampling filter). The upsampled feature maps from each level are summed together to give a feature representation that has the same size as the input image. We perform one more convolution on the summed feature representation and apply a softmax transform across channels at each spatial location. The output of this softmax layer is interpreted as a probabilistic disparity-like map. We then feed this disparity-like map and the left view to the selection layer, which outputs the right view.

Bilinear Interpolation by Deconvolution. Similar to [17] we use “deconvolutional” layers to upsample lower layer features maps before feeding them to the final representation. Deconvolutional layers are implemented by reversing the forward and backward computations of a convolution layer.

We found that initializing the deconvolutional layers to be equivalent to bilinear interpolation can facilitate training. Specifically, for upsampling by factor S , we use a deconvolutional layer with $2S$ by $2S$ kernel, S by S stride, and $S/2$ by

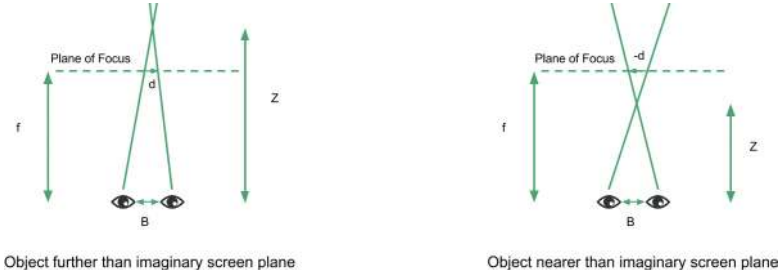


Fig. 3. Depth to disparity conversion. Given the distance between the eyes B and the distance between the eyes and the plane of focus f , we can compute disparity from depth with Eq. 3. Disparity is negative if object is closer than the plane of focus and positive if it is further away.

$S/2$ padding. The kernel weight w is then initialized with:

$$C = \frac{2S - 1 - (S \bmod 2)}{2S} \quad (1)$$

$$w_{ij} = \left(1 - \left|\frac{i}{S - C}\right|\right) \left(1 - \left|\frac{j}{S - C}\right|\right) \quad (2)$$

3.2 Reconstruction with Selection Layer

The selection layer models the DIBR step in traditional 2D-to-3D conversion. In traditional 2D-to-3D conversion, given the left view I and a depth map Z , a disparity map D is first computed with

$$D = \frac{B(Z - f)}{Z} \quad (3)$$

where the baseline B is the distance between the two cameras, Z is the input depth and f is the distance from cameras to the plane of focus. See Fig. 3 for illustration. The right view O is then generated with:

$$O_{i,j+D_{i,j}} = I_{i,j}. \quad (4)$$

However this is not differentiable with respect to D so we cannot train it together with a deep neural network. Instead, our network predicts a probability distribution across possible disparity values d at each pixel location $D_{i,j}^d$, where $\sum_d D_{i,j}^d = 1$ for all i, j . We define a shifted stack of the left view as $I_{i,j}^d = I_{i,j-d}$, then the selection layer reconstructs the right view with:

$$O_{i,j} = \sum_d I_{i,j}^d D_{i,j}^d \quad (5)$$

This is now differentiable with respect to $D_{i,j}^d$ so we can compute an L1 loss between the output and ground-truth right view Y as the training objective:

$$L = |O - Y| \quad (6)$$

We use L1 loss because recent research has shown that it outperforms L2 loss for pixel-wise prediction tasks [18].

We note that D is only an intermediate result optimized for producing low error reconstructions while not intended to be an accurate disparity prediction. In fact, we observe that in practice D serves the dual purpose of depth estimation and in-painting. Low texture regions also tend to be ignored as they do not significantly contribute to reconstruction error.

3.3 Scaling up to Full Resolution

Modern movies are usually distributed in at least 1080p resolution, which has 1920 pixel by 800 pixel frames. In our experiments, We reduce input frames to 432 by 180 to preserve aspect ratio and save computation time. As a result, the generated right view frames will only have a resolution of 432 by 180, which is unacceptably low for movie viewing.

To address this issue, we first observe that the disparity map usually has much less high-frequency content than the original color image. Therefore we can scale up the predicted disparity map and couple it with the original high resolution left view to render a full resolution right view. The right view rendered this way has better image quality compared to the naively 4x-upsampled prediction.

4 Dataset

Since Deep3D can be trained directly on stereo pairs without ground-truth depth maps as supervision, we can take advantage of the large volume of existing stereo videos instead of using traditional scene depth datasets like KITTI [13] and NYU Depth [12]. We collected 27 non-animation 3D movies produced in recent years and randomly partitioned them to 18 for training and 9 for testing. Our dataset contains around 5 million frames while KITTI and NYU Depth only provide several hundred frames. During training, each input left frame is resized to 432 by 180 pixels and a crop of size 384 by 160 pixels is randomly selected from the frame. The target right frame undergoes the same transformations. We do not use horizontal flipping.

5 Experiments

In our main experiments we use a single frame at a time as input without exploiting temporal information. This choice ensures a more fair comparison to single-frame baseline algorithms and also allows applying trained models to static photos in addition to videos. However, it is natural to hypothesize that motion provides important cues for depth, thus we also conducted additional experiments that use consecutive RGB frames and computed optical flow as input, following [19]. These results are discussed in Sect. 5.4.

5.1 Implementation Details

For quantitative evaluation we use the non-upsampled output size of 384 by 160 pixels. For qualitative and human subject evaluation we upsample the output by a factor of 4 using the method described in Sect. 3.3. Our network is based on VGG16, which is a large convolutional network trained on ImageNet [20]. We initialize the main branch convolutional layers (colored green in Fig. 2) with VGG16 weight and initialize all other weights with normal distribution with a standard deviation of 0.01.

To integrate information from lower level features, we create a side branch after each pooling layer by applying batch normalization [21] followed by a 3×3 convolution layer. This is then followed by a deconvolution layer initialized to be equivalent to bilinear upsampling. The output dimensions of the deconvolution layers match the final prediction dimensions. We use batch normalization to connect pretrained VGG16 layers to randomly initialized layers because it solves the numerical instability problem caused by VGG16’s large and non-uniform activation magnitude.

We also connect the top VGG16 convolution layer feature to two randomly initialized fully connected layers (colored blue in Fig. 2) with 4096 hidden units followed by a linear layer. We then reshape the output of the linear layer to 33 channels of 12 by 5 feature maps which is then fed to a deconvolution layer. We then sum across all up sampled feature maps and do a convolution to get the final feature representation. The representation is then fed to the selection layer. The selection layer interprets this representation as the probability over empty or disparity -15 to 16 (a total of 33 channels).

In all experiments Deep3D is trained with a mini-batch size of 64 for 100,000 iterations in total. The initial learning rate is set to 0.002 and reduce it by a factor of 10 after every 20,000 iterations. No weight decay is used and dropout with rate 0.5 is only applied after the fully connected layers. Training takes two days on one NVidia GTX Titan X GPU. Once trained, Deep3D can reconstruct novel right views at more than 100 frames per second. Our implementation is based on MXNet [22] and available for download at <https://github.com/piiswong/deep3d>.

5.2 Comparison Algorithms

We used three baseline algorithms for comparison:

1. Global Disparity: the right view is computed by shifting the left view with a global disparity δ that is determined by minimizing Mean Absolution Error (MAE) on the validation set.
2. The DNN-based monocular depth estimation algorithm of Eigen et al. [10] plus a standard DIBR method as described in Sect. 3.2.
3. DNN-based Monocular depth estimation trained to predict disparity estimation from stereo block matching algorithms, plus standard DIBR method.
4. Ground-truth stereo pairs. We only show the ground-truth in human subject studies since in quantitative evaluations it always gives zero error.

To the best of our knowledge, Deep3D is the first 2D-to-3D conversion algorithm that can be trained directly on stereo pairs, while all previous methods requires ground-truth depth map for training. As one baseline, we take the model released by Eigen et al. [10], which is trained on NYU Depth [12], and evaluate it on our test set. However, it is a stretch to hope that a model trained on NYU Depth will generalize well to 3D movies. Therefore, for a more fair comparison, we also a retrain monocular depth estimation network with estimated depth from stereo block matching algorithms on the same 3D movie dataset. Since Eigen et al. did not release training code, we instead use the same VGG-based network architecture proposed in this paper. This also has the added benefit of being directly comparable to Deep3D.

Because [10] predicts depth rather than disparity, we need to convert depth to disparity with Eq. 3 for rendering with DIBR. However, [10] does not predict the distance to the plane of focus f , a quantity that is unknown and varies across shots due to zooming. The interpupillary distance B is also unknown, but it is fixed across shots. The value of B and f can be determined in two ways:

1. Optimize for MAE on the validation set and use fixed values for B and f across the whole test set. This approach corresponds to the lower bound of [10]’s performance.
2. Fix B across the test set, but pick the f that gives the lowest MAE for each *test* frame. This corresponds to having access to oracle plane of focus distance and thus the upper bound on [10]’s performance.

We do both and report them as two separate baselines, [10] and [10] + Oracle. For fair comparisons, we also do this optimization for Deep3D’s predictions and report the performance of Deep3D and Deep3D + Oracle.

5.3 Results

Quantitative Evaluation. For quantitative evaluation, we compute Mean Absolute Error (MAE) as:

$$MAE = \frac{1}{HW} |y - g(x)|, \quad (7)$$

where x is the left view, y is the right view, $g(\cdot)$ is the model, and H and W are height and width of the image respectively. The results are shown in Table 1. We observe that Deep3D outperforms baselines with and without oracle distance of focus plane.

Qualitative Evaluation. To better understand the proposed method, we show qualitative results in Fig. 5. Each entry starts with a stereo pair predicted by Deep3D shown in anaglyph, followed by 12 channels of internal soft disparity assignment, ordered from near (-3) to far ($+8$). We observe that Deep3D is able to infer depth from multiple cues including size, occlusion, and geometric structure.

Table 1. Deep3D evaluation. We compare pixel-wise reconstruction error for each method using Mean Absolute Error (MAE) as metric.

Method	MAE
Global disparity	7.75
[10]	7.75
DNN trained on estimated depth	7.29
Deep3D (ours)	6.87
[10] + Oracle	6.31
Deep3D + Oracle	5.47

Table 2. Human subject evaluation. In each entry, the first number represents the frequency of the row method being preferred to the column method by human subjects, while the second number represents their certainty (i.e. the percentage of people who gave an answer instead of “not sure”). Note that 66 % of times subjects prefer Deep3D to [10] and 24 % of the times Deep3D is preferred over the ground truth.

	Global disparity	[10] + Oracle	Deep3D (ours)	Ground truth
Trivial	N/A	27.21 %/56 %	25.28 %/74 %	7.80 %/77 %
Eigen et al.	72.79 %/56 %	N/A	33.53 %/71 %	10.19 %/77 %
Ours	74.72 %/74 %	66.47 %/71 %	N/A	24.43 %/81 %
Ground truth	92.20 %/77 %	89.81 %/77 %	75.57 %/81 %	N/A

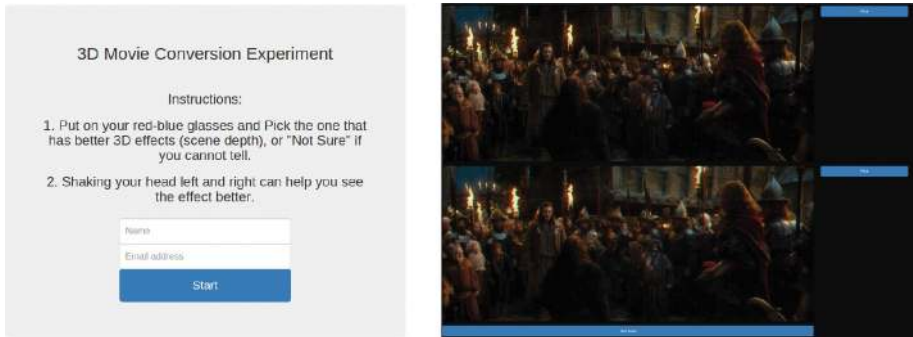


Fig. 4. Human subject study setup. Each subject is shown 50 pairs of 3D anaglyph images. Each pair consists of the same scene generated by 2 randomly selected methods. The subjects are instructed to wear red-blue 3D glasses and pick the one with better 3D effects or “not sure” if they cannot tell. The study result is shown in Table 2. (Color figure online)

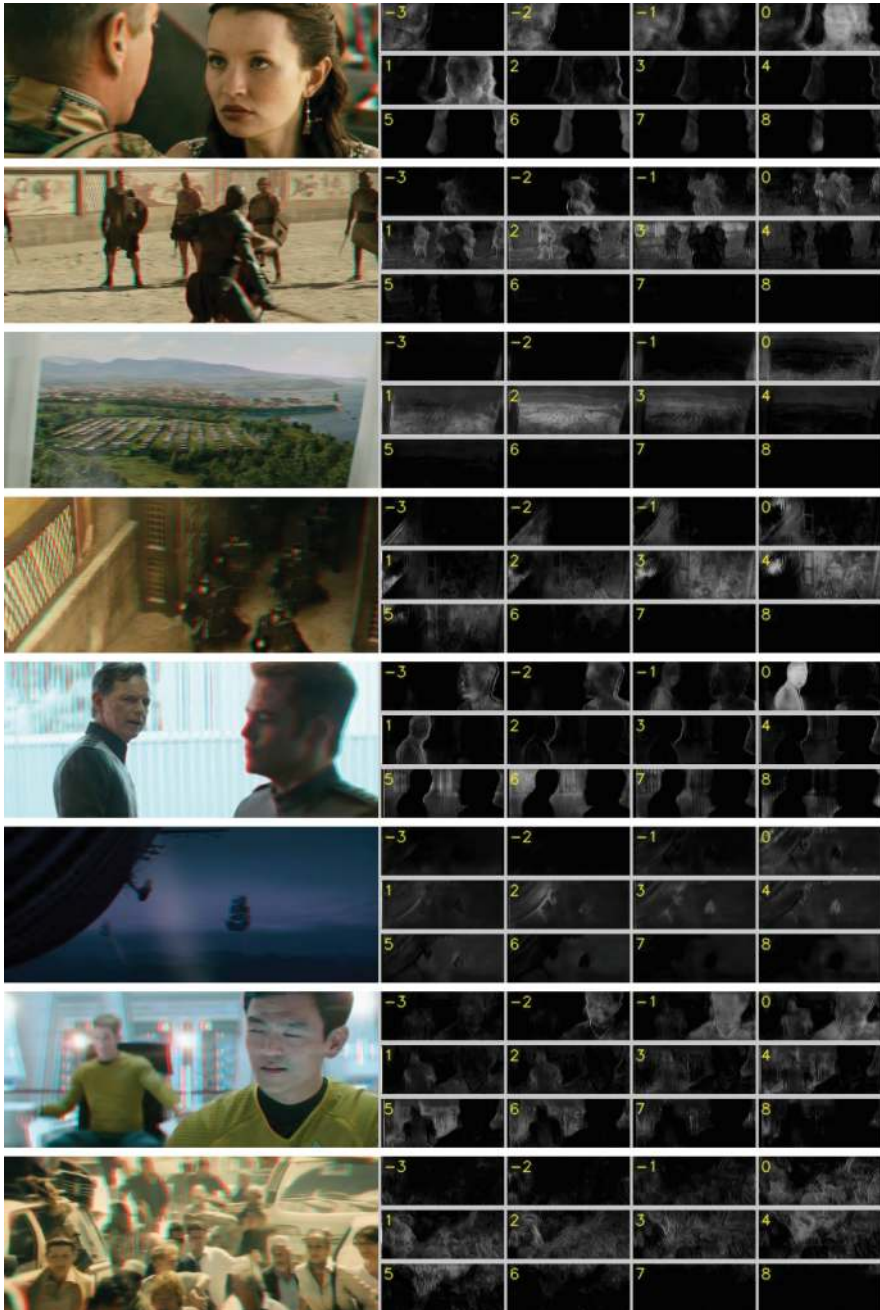


Fig. 5. Qualitative results. Column one shows an anaglyph of the predicted 3D image (best viewed in color with red-blue 3D glasses). Each anaglyph is followed by 12 heat maps of disparity channels -3 to 8 (closer to far). In the first example, the man is closer and appears in the first 3 channels while the woman is further away and appears in 4th-5th channels; the background appears in the last 4 channels. In the second example, the person seen from behind is closer than the other 4 people fighting him. In the third example, the window frame appears in the first 3 channels while the distant outside scene gradually appears in the following channels. (Color figure online)



Fig. 6. Comparison between [10] and Deep3D. The first column shows the input image. The second column shows the prediction of [10] and the third column shows Deep3D’s prediction. This figure shows that Deep3D is better at delineating people and figuring out their distance from the camera.

We also compare Deep3D’s internal disparity maps (column 3) to [10]’s depth predictions (column 2) in Fig. 6. This figure demonstrates that Deep3D is better at delineating people and figuring out their distance from the camera.

Note that the disparity maps generated by Deep3D tend to be noisy at image regions with low horizontal gradient, however this does not affect the quality of the final reconstruction because if a row of pixels have the same value, any disparity assignment would give the same reconstruction. Disparity prediction only needs to be accurate around vertical edges and we indeed observe that Deep3D tends to focus on such regions.

Human Subject Evaluation We also conducted a human subject study to evaluate the visual quality of the predictions of different algorithms. We used four algorithms for this experiment: Global Disparity, [10] + Oracle, Deep3D without Oracle, and the ground-truth.²

For the human subject study, we randomly selected 500 frames from the test set. Each annotator is shown a sequence of trials. In each trial, the annotator sees two anaglyph 3D images, which are reconstructed from the same 2D frame by two algorithms, and is instructed to wear red-blue 3D glasses and pick the one with better 3D effects or select “not sure” if they are similar. The interface for this study is shown in Fig. 4. Each annotator is given 50 such pairs and we collected decisions on all $C_4^2 500$ pairs from 60 annotators.

Table 2 shows that Deep3D outperforms the naive Global Disparity baseline by a 49 % margin and outperforms [10] + Oracle by a 32 % margin. When facing

² [10] without Oracle and Deep3D + Oracle are left out due to annotator budget Note that a change in average scene depth only pushes a scene further away or pull it closer and usually doesn’t affect the perception of depth variation in the scene.

against the ground truth, Deep3D’s prediction is preferred 24.48 % of the time while [10] + Oracle is only preferred 10.27 % of the time and Global Disparity baseline is preferred 7.88 % of the time.

5.4 Algorithm Analysis

Ablation Study. To understand the contribution of each component of the proposed algorithm, we show the performance of Deep3D with parts removed in Table 3. In Deep3D w/o lower level feature we show the performance of Deep3D without branching off from lower convolution layers. The resulting network only has one feed-forward path that consists of 5 convolution and pooling module and 2 fully connected layers. We observe that the performance significantly decreases compared to the full method.

In Deep3D w/o direct training on stereo pairs we show the performance of training on disparity maps generated from stereo pairs by block matching algorithm [23] instead of directly training on stereo pairs. The predicted disparity maps are then fed to DIBR method to render the right view. This approach results in decreased performance and demonstrates the effectiveness of Deep3D’s end-to-end training scheme.

We also show the result from directly regressing on the novel view without internal disparity representation and selection layer. Empirically this also leads to decreased performance, demonstrating the effectiveness of modeling the DIBR process.

Temporal Information. In our main experiment and evaluation we only used one still frame of RGB image as input. We made this choice for fair comparisons and more general application domains. Incorporating temporal information into Deep3D can be handled in two ways: use multiple consecutive RGB frames as input to the network, or provide temporal information through optical flow frames similar to [19].

We briefly explored both directions and found moderate performance improvements in terms of pixel-wise metrics. We believe more effort along this direction, such as model structure adjustment, hyper-parameter tuning, and

Table 3. Ablation studies. We evaluate different components of Deep3D by removing them from the model to further understand the contribution of each component. Note that removing lower level features and selection layer both result in performance drop.

Method	MAE
Deep3D w/o lower level feature	8.24
Deep3D w/o direct training on stereo pairs	7.29
Deep3D w/o selection layer	7.01
Deep3D	6.87

Table 4. Temporal information. We incorporate temporal information by extending the input to include multiple consecutive RGB frames or optical flow frames. We observe that additional temporal information leads to performance gains.

Method	MAE
Deep3D with 5 RGB frames	6.81
Deep3D with 1 RGB frames and 5 optical flow frames	6.86
Deep3D	6.87

explicit modeling of time will lead to larger performance gains at the cost of restricting application domain to videos only (Table 4).

6 Conclusions

In this paper we proposed a fully automatic 2D-to-3D conversion algorithm based on deep convolutional neural networks. Our method is trained end-to-end on stereo image pairs directly, thus able to exploit orders of magnitude more data than traditional learning based 2D-to-3D conversion methods. Quantitatively, our method outperforms baseline algorithms. In human subject study stereo images generated by our method are consistently preferred by subjects over results from baseline algorithms. When facing against the ground truth, our results have a higher chance of confusing subjects than baseline results.

In our experiment and evaluations we only used still images as input while ignoring temporal information from video. The benefit of this design is that the trained model can be applied to not only videos but also photos. However, in the context of video conversion, it is likely that taking advantage of temporal information can improve performance. We briefly experimented with this idea but found little quantitative performance gain. We conjecture this may be due to the complexity of effectively incorporating temporal information. We believe this is an interesting direction for future research.

Acknowledgements. This work is in part supported by ONR N00014-13-1-0720, NSF IIS-1338054, Allen Distinguished Investigator Award and contracts from the Allen Institute for Artificial Intelligence.

References

1. Motion Picture Association of America: Theatrical market statistics (2014)
2. Fehn, C.: Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-tv. In: Electronic Imaging 2004, International Society for Optics and Photonics, pp. 93–104 (2004)
3. Zhuo, S., Sim, T.: On the recovery of depth from a single defocused image. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 889–897. Springer, Heidelberg (2009)

4. Cozman, F., Krotkov, E.: Depth from scattering. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings, pp. 801–806. IEEE (1997)
5. Zhang, L., Vázquez, C., Knorr, S.: 3D-tv content creation: automatic 2D-to-3D video conversion. *IEEE Trans. Broadcast.* **57**(2), 372–383 (2011)
6. Konrad, J., Wang, M., Ishwar, P., Wu, C., Mukherjee, D.: Learning-based, automatic 2D-to-3D image and video conversion. *IEEE Trans. Image Process.* **22**(9), 3485–3496 (2013)
7. Appia, V., Batur, U.: Fully automatic 2D to 3D conversion with aid of high-level image features. In: IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, p. 90110W (2014)
8. Saxena, A., Sun, M., Ng, A.Y.: Make3D: learning 3D scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(5), 824–840 (2009)
9. Baig, M.H., Jagadeesh, V., Piramuthu, R., Bhardwaj, A., Di, W., Sundaresan, N.: Im2depth: scalable exemplar based depth transfer. In: 2014 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 145–152. IEEE (2014)
10. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2650–2658 (2015)
11. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5162–5170 (2015)
12. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012)
13. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the kitti dataset. *Int. J. Rob. Res. (IJRR)* **32**, 1231–1237 (2013)
14. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1529–1537 (2015)
15. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. arXiv preprint [arXiv:1504.00702](https://arxiv.org/abs/1504.00702) (2015)
16. Flynn, J., Neulander, I., Philbin, J., Snavely, N.: Deepstereo: learning to predict new views from the world’s imagery. arXiv preprint [arXiv:1506.06825](https://arxiv.org/abs/1506.06825) (2015)
17. Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: learning optical flow with convolutional networks. arXiv preprint [arXiv:1504.06852](https://arxiv.org/abs/1504.06852) (2015)
18. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. arXiv preprint [arXiv:1511.05440](https://arxiv.org/abs/1511.05440) (2015)
19. Wang, L., Xiong, Y., Wang, Z., Qiao, Y.: Towards good practices for very deep two-stream convnets. arXiv preprint [arXiv:1507.02159](https://arxiv.org/abs/1507.02159) (2015)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
21. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint [arXiv:1502.03167](https://arxiv.org/abs/1502.03167) (2015)

22. Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., Zhang, Z.: Mxnet: a flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint [arXiv:1512.01274](https://arxiv.org/abs/1512.01274) (2015)
23. Hirschmüller, H.: Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(2), 328–341 (2008)