



# Open-World Stereo Video Matching with Deep RNN

Yiran Zhong<sup>1,3,4</sup>(✉), Hongdong Li<sup>1,3</sup>, and Yuchao Dai<sup>2</sup>

<sup>1</sup> Australian National University, Canberra, Australia

<sup>2</sup> Northwestern Polytechnical University, Xi'an, China

{yiran.zhong,hongdong.li}@anu.edu.au, daiyuchao@nwpu.edu.cn

<sup>3</sup> Australian Centre for Robotic Vision, Canberra, Australia

<sup>4</sup> Data61 CSIRO, Canberra, Australia

**Abstract.** Deep Learning based stereo matching methods have shown great successes and achieved top scores across different benchmarks. However, like most data-driven methods, existing deep stereo matching networks suffer from some well-known drawbacks such as requiring large amount of labeled training data, and that their performances are fundamentally limited by the generalization ability. In this paper, we propose a novel Recurrent Neural Network (RNN) that takes a continuous (possibly previously unseen) stereo video as input, and directly predicts a depth-map at each frame without a pre-training process, and without the need of ground-truth depth-maps as supervision. Thanks to the recurrent nature (provided by two convolutional-LSTM blocks), our network is able to memorize and learn from its past experiences, and modify its inner parameters (network weights) to adapt to previously unseen or unfamiliar environments. This suggests a remarkable generalization ability of the net, making it applicable in an *open world* setting. Our method works robustly with changes in scene content, image statistics, and lighting and season conditions *etc.* By extensive experiments, we demonstrate that the proposed method seamlessly adapts between different scenarios. Equally important, in terms of the stereo matching accuracy, it outperforms state-of-the-art deep stereo approaches on standard benchmark datasets such as KITTI and Middlebury stereo.

**Keywords:** Stereo video matching · Open world  
Recurrent neural network · Convolutional LSTM

## 1 Introduction

Stereo matching is a classic problem in computer vision, and it has been extensively studied in the literature for decades. Recently, deep learning based

---

**Electronic supplementary material** The online version of this chapter ([https://doi.org/10.1007/978-3-030-01216-8\\_7](https://doi.org/10.1007/978-3-030-01216-8_7)) contains supplementary material, which is available to authorized users.

stereo matching methods are taking over, becoming one of the best performing approaches. As an evidence, they occupy the leader-boards for almost all the standard stereo matching benchmarks (*e.g.*, KITTI [1], Middlebury stereo [2]).

However, there exists a considerable gap between the success of these “deep stereo matching methods” on somewhat artificially created benchmark datasets and their real-world performances when being employed “in the wild” (open world), probably for the following reasons:

- (1) Most of the existing deep stereo matching methods are supervised learning based methods, for which the training process demands massive annotated training samples. In the context of stereo matching, getting large amount of training data (*i.e.* ground-truth disparity/depth maps) is an extremely expensive task.
- (2) The performance of existing deep stereo matching methods and their applicability in real-world scenarios are fundamentally limited by their generalization ability: like most data-driven methods, they only work well on testing data that are sufficiently similar to the training data. Take autonomous driving for example, a deep stereo matching network trained in one city, under one traffic condition, might not work well in another city, under different lighting conditions.
- (3) So far, most deep stereo matching methods exclusively focus on processing single pair of stereo images in a frame-by-frame manner, while in real world stereo camera captures continuous video. The rich temporal information contained in the stereo video has not been exploited to improve the stereo matching performance or robustness.

In this paper, we tackle all the above drawbacks with current deep stereo matching methods. We propose a novel deep Recurrent Neural Network (RNN) that computes a depth/disparity map continuously from stereo video, without any pre-training process. Contrary to conventional stereo matching methods (*e.g.*, [3, 4]) which focus on processing a single pair of stereo images individually, this work is capitalized on explicitly exploiting the temporally dynamic nature of stereo video input.

Our deep stereo video matching network, termed as “OpenStereoNet” is not fixed, but changes its inner parameters continuously as long as new stereo frames being fed into the network. This enables our network to adapt to changing situations (*e.g.* changing lighting condition, changing image contents, *etc.*), allowing it to work in unconstrained *open world* environments. OpenStereoNet is made of a convolutional Feature-Net for feature extraction, a Match-Net for depth prediction, and two recurrent Long Short-Term Memory (LSTM) blocks to encode and to exploit temporal dynamics in the video. Importantly and in contrast to existing deep stereo matching methods, our network does not need any ground-truth disparity map as supervision, yet it naturally generalizes well to unseen datasets. As new videos are processed, the network is able to memorize, and to learn from, its past experiences. Without needing ground-truth disparity maps, our network is able to tune its parameters after seeing more images from

stereo videos, simply by minimizing image-domain warping errors. Also, to better leverage the sequential information in the stereo video, we apply the Long Short-Term Memory (LSTM) module to the bottleneck of feature extraction and feature matching part of our network. In the later part of this paper, we demonstrate that our method can be applied to vary open-world scenarios such as indoor/outdoor scenes, different weather/light conditions and different camera settings with superior performance. Also ablation study concerning the effect of the LSTM modules is conducted. Another novelty of this work is that: we adopt convolutional-LSTM [5] (cLSTM) as the recurrent feedback module, and use it directly on a continuous video sequence harnessing the temporal dynamics of the video. To our knowledge, while RNN-LSTM has been applied to other video processing tasks (such as sequence captions, or human action recognition), it has not been used for stereo matching for video sequences.

## 2 Related Work

Stereo matching is a classic problem in computer vision, and has been researched for several decades. There have been significant number of papers published on this topic (The reader is referred to some survey papers *e.g.*, [2,6]). Below we only cite a few most recent deep-learning based stereo methods that we consider most closely related to the method to be described.

**Supervised Deep Stereo Matching.** In this category, a deep network (often based on CNN, or Convolutional Neural Networks) is often trained to benefit the task of stereo matching in one of the following aspects: (i) to learn better image features and a tailored stereo matching metrics (*e.g.*, [7,8]); (ii) to learn better regularization terms in a loss function [9]; and (iii) to predict dense disparity map in an end-to-end fashion (*e.g.*, [4,10]). The learned deep features replace handcrafted features, resulting in more distinctive features for matching. End-to-end deep stereo methods often formulate the task as either depth values regression, or multiple (discrete) class classification. DispNetC [10] is a new development, which directly computes the correspondence field between stereo images by minimizing a regression loss. Another example is the GC-Net [4], which explicitly learns feature cost volume, and regularization function in a network structure. Cascade residual learning (CRL) [11] adopted a multi-stage cascade CNN architecture, following a coarse-to-fine or residual learning principle [12].

**Unsupervised Deep Stereo Matching.** Recently, there have been proposed deep net based single-image depth recovery methods which do not require ground-truth depth maps. Instead, they rely on minimizing photometric warping error to drive the network in an unsupervised way (see *e.g.*, [13–17]). Zhou *et.al.* [15] proposed an unsupervised method which is iteratively trained via warping error propagating matches. The authors adopted TV (total variation) constraint to select training data and discard uninformative patches. Inspired by recent advances in direct visual odometry (DVO), Wang *et.al.* [18] argued that the depth CNN predictor can be learned without a pose CNN predictor.

Luo *et al.* [19] reformulated the problem of monocular depth estimation as two sub-problems, namely a view synthesis procedure followed by standard stereo matching. However, extending these monocular methods to stereo matching is non-trivial. When feeding the networks with stereo pairs, their performances are even not comparable with traditional stereo matching methods [14].

**Recurrent Neural Net and LSTM.** Our method is based on RNN (with cLSTM as the feedback module), and directly applied to sequence input of stereo video harnessing the temporal dynamic nature of a continuous video. To the best of our knowledge, where RNN-LSTM has been applied to other video based tasks (such as a sequence captions, action recognition), it has not been directly used for stereo video matching, especially to exploit the temporal smoothness feature for improving stereo matching performance.

### 3 Network Architecture

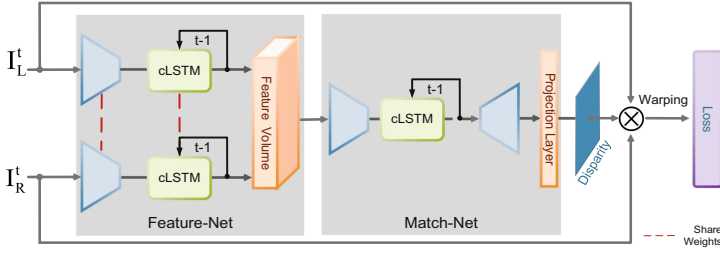
In this section, we describe our new “open-world” stereo video matching deep neural network (for ease of reference, we call it *OpenStereoNet*). The input to the network is a live continuous stereo video sequence of left and right image frames of  $I_L^t, I_R^t$ , for  $t = 1, 2, \dots$ . The output is the predicted depth-map (disparity map) at each time step  $t$ . We assume the input stereo images are already rectified.

Our network does not require ground-truth depth-maps as supervision. Instead, the stereo matching task is implemented by searching a better depth map which results in minimal photometric warping error between the stereo image pair. By continuously feeding in new stereo image frames, our network is able to automatically adapt itself to new inputs (could be new visual scenes never seen before) and produce accurate depth map estimations. More technical details will be explained in the sequel of the paper.

#### 3.1 Overall Network Architecture

The overall structure of our OpenStereoNet is illustrated in Fig. 1, which consists of the following major parts (or sub-Nets): (1) Feature-Net, (2) Match-Net, (3) LSTM blocks, and (4) a loss function block.

*Information Flow.* Starting from inputted left and right images at time  $t$ , the information processing flow in our network is clear: (1) The Feature-Net acts as a convolutional feature extractor which extracts features from the left and right images individually. Note, Feature-Net for the left image and Feature-Net for the right image share the weights. (2) The obtained feature maps are concatenated (with certain interleave pattern) into a 4D feature-volume. (3) The Match-Net takes the 4D feature volume as input, and learns an encoder-decoder representation of the features. A projection layer (based on soft-argmin [4]) within the Match-Net is applied to produce the 2D disparity map prediction. Finally, the loss function block employs the current estimated disparity map to warp the right image to the left view and compare the photometric warping



**Fig. 1.** Overall network structure of our OpenStereoNet. It consists of a convolutional Feature-Net, an encoder-decoder type Match-Net and two recurrent (convolutional) LSTM units to learn temporal dynamics of the video input. Given a stereo pair  $I_L^t, I_R^t$  at time  $t$ , the Feature-Net produces feature maps which are subsequently aggregated to form a feature-volume. The Match-Net first learns a representation of the feature volume then projects it to obtain disparity estimation. Our loss function is based on image warping error evaluated on raw image inputs by the current disparity map.

loss as well as other regularization term, which is used to refine the network via *backprop*.

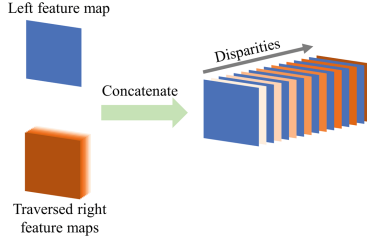
### 3.2 Feature-Net

Conventional stereo matching methods often directly compare the raw pixel values in the left image with that in the right image. Recent advance in deep learning show that using learned convolutional features can be more robust for various vision tasks. For stereo matching, to learn a feature map that is more robust to photometric variations (such as occlusion, non-lambertian, lighting effects and perspective effects) will be highly desirable.

In this paper, we design a very simple convolutional feature-net with 18 convolutional layers (including RELU) using  $3 \times 3$  kernels and skip connections in between. The output feature has a dimensionality of 32. We run feature extraction on both images in a symmetric weight-sharing manner.

### 3.3 Feature-Volume Construction

We use the learned features to construct a feature volume. Instead of constructing a cost volume by concatenating all costs with their corresponding disparities, we concatenate the learned features from the left and right images. Specifically, we concatenate each learned feature with their corresponding feature from the opposite stereo image across each disparity level in a preset disparity range  $D$  as illustrated in Fig. 2. All the features are packed to form a 4D feature volume with dimensionality  $H \times W \times (D + 1) \times 2F$  for the left-to-right and right-to-left feature volume correspondingly, where  $H, W, D, F$  represent the height, width, disparity range, and feature dimensionality respectively.



**Fig. 2.** Feature Volume Construction. We collect the two feature maps computed by the Feature-Net, and assemble them together to a Feature-Volume in the way as illustrated here: the blue rectangle represents a feature map from the left image, the stacked orange rectangles represent traversed the right feature at different disparities in the range  $[0, D]$ . Note that the left feature map is duplicated  $D + 1$  times to match the traversed right feature maps. (Color figure online)

### 3.4 Match-Net

Taking the assembled Feature-Volume as input, our Match-Net is constituted of an encoder-decoder as the front-end, followed by a single last layer which projects the output of the encoder-decoder to a 2D disparity-map.

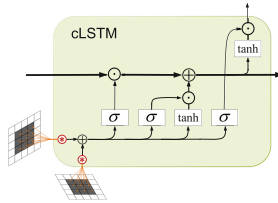
**Encoder-Decoder Front-End.** The denoising Encoder-Decoder is an hour-glass-shaped deep-net. Between the encoder and decoder there is a bottleneck, as shown in Fig. 1. Since the input feature-volume is of 4 dimensions,  $H$  (height)  $\times W$  (width)  $\times (D + 1)$ (disparity range)  $\times 2F$ (feature dim.), we use 3D-convolutional kernels and the underlying CNNs in the Encoder-Decoder are in fact 3D-CNNs.

**Projection Layer.** The output of the preceding encoder-decoder is still a 4D feature-volume. The last layer of our Match-Net first projects the 4D volume to a 3D *cost-volume*—*i.e.* an operation commonly used in conventional stereo matching methods, then applies the soft-argmin operation (*c.f.* [4]) to predict a disparity  $\delta = \sum_{d=0}^D [d \times \sigma(-c_d)]$ , where  $c_d$  is the matching cost at disparity  $d$  and  $\sigma(\cdot)$  denotes the softmax operator.

### 3.5 Convolutional-LSTM

Since our goal is to develop a deep-net focusing on stereo video processing (as opposed to individual images), in order to capture the inherent temporal dynamics (*e.g.*, temporal smoothness) existed in a video, we leverage the internal representations obtained by our two sub-networks (*i.e.*, Feature-Net and Match-Net), and model these internal representation’s dynamic transitions as an *implicit* model for the video sequence. Specifically, given a continuous video, we consider the image content (as well as the disparity) in each frame changes smoothly to

the next frame. To capture such dynamic changes, we adopt the structure of LSTM-based Recurrent Neural Networks (RNN). The LSTMs act as memory of the net, by which the network memorizes its past experiences. This gives our network the ability to learn the stereo video sequence’s temporal dynamics encoded in the inner states of the LSTMs. As shown in Fig. 3, the output of our Feature-Net, and the encoder-decoder in our Match-Net, are each passed to an LSTM unit. Briefly, LSTM units are a particular type of hidden unit that improve the training of RNNs [20]. An LSTM unit contains a cell, which can be thought of as a memory state. Access to the cell is controlled through an input gate and a forget gate. The final output of the LSTM unit is a function of the cell state and an output gate (*c.f.* [21]).



**Fig. 3.** A convolutional-LSTM.

We realize that for stereo matching it is desirable to use spatially-invariant operator such as convolutional kernels, in the same spirit of the CNN. In light of this, we propose to use the convolutional LSTM architecture (or cLSTM in short, *c.f.* [5]) as the recurrent unit for our stereo video matching task. This way, the relative spatial layout information of the feature representation—which is essential for the task of depth map computation—is preserved. In our experiments we used very small kernels for the cLSTM (*e.g.*  $3 \times 3$ , or  $5 \times 5$ ). This leads to compact LSTM units, and greatly simplifies the computation cost and GPU-RAM consumption. The encoder–LSTM–decoder architecture is also similar to the Encoder–Recurrent–Decoder architecture proposed in [22]. Our entire network works end-to-end to combine feature representation learning with the learning of temporal dynamics via the two LSTM blocks.

## 4 Self-adapting Learning and Loss Function

### 4.1 Self-adapting Training and Testing

Recall that the ultimate goal of this work is to develop a deep network that can automatically adapt itself to new (previously unseen) stereo video inputs. In this sense, the network is not fixed static, but is able to dynamically evolve in time. This is achieved by two mechanisms:

- The network has memory units (i.e. the two LSTM blocks), which enable the network to adjust its current behavior (partly) based its past experiences;

- We always run an online back-propagation (*backprop*) updating procedure after any feed-forward process.

The latter actually eliminates the separation between a network’s training stage and testing stage. In other words, our OpenStereoNet is constantly performing both operations all the time. This gives the network self-adaption ability, allows it to self-adapt by continuously fine-tuning its parameters based on new stereo image inputs (possibly seen in a new environment). Thus, our OpenStereoNet can “automatically” generalize to unseen images.

Since we do not require ground-truth depth-maps as supervision, input stereo pairs themselves serve as self-supervision signals, and the network is able to update automatically, by self-adapting learning.

## 4.2 Overall Loss Function

The overall loss function for our OpenStereoNet is a weighted summation of a data term and a regularization term, as in  $Loss = L_{data} + \mu L_{reg}$ .

**Data Term: Image Warping Error.** We directly measure the warping error evaluated on the input stereo images, based on the estimated disparity map. Specifically, given the left image  $I_L^t$  and the disparity map for the right image  $d_R^t = g(I_R^t, I_L^t, h_R^{t-1}, h_L^{t-1})$ , the right image  $I_R^t$  can be reconstructed by warping the left image with  $d_R^t$ ,  $I_R^t(u, v) = I_L^t(u + d, v)$ , where  $I_R^t$  is the warped right image. We use the discrepancy between  $I_R^t$  and the observed right image  $I_R^t$  as the supervision signal. Our data loss is derived as:  $L_{data} = \sum(\lambda_1(1 - \mathcal{S}(I_L, I_L'))/2 + \lambda_2(|I_L - I_L'| + |\nabla I_L - \nabla I_L'|)/N$ . The data term consists of  $S$ , which is the structural similarity SSIM as defined in [23], pixel value difference and image gradient difference. The trade-off parameters were chosen empirically in our experiments at  $\mu = 0.05$ ,  $\lambda_1 = 0.8$ ,  $\lambda_2 = 0.1$ .

**Regularization Term: Priors on Depth-Map.** We enforce a common prior that depth-maps are piecewise smooth or piecewise linear. This is implemented by penalizing the second-order derivative of the estimated disparity map. To exploit correlations between depth map values and pixel colors, we weight this term by image color gradient, *i.e.*:  $L_{reg} = \sum(e^{-|\nabla_u^2 I_L|} |\nabla_u^2 d_L| + e^{-|\nabla_v^2 I_L|} |\nabla_v^2 d_L|)/N$ , where  $\nabla$  is gradient operator.

## 5 Experiments

We implement our OpenStereoNet in TensorFlow. Since the network runs in an online fashion (with batch-size one), *i.e.*, there is no clear distinction between training and testing, we start from randomly initialized weights for both the Feature-Net and the Match-Net, and allow the network to evolve as new stereo images being fed in. All images have been rescaled to  $256 \times 512$  for easy comparison. Typical processing time of our net is about 0.8–1.6 s per frame tested on a



regular PC of 2017 equipped with a GTX 1080Ti GPU. We use the RMSProp optimizer with a constant learning rate of 0.001. We have evaluated our network on several standard benchmark datasets for stereo matching, including KITTI [1], Middlebury [24,25], Synthia [26], and Frieberg SceneFlow [10] (e.g. FlyingThings3D). These experiments are reported below.

### 5.1 KITTI Visual Odometry (VO) Stereo Sequences

In this set of experiments on KITTI dataset [1], we simply feed a KITTI VO stereo video sequence to our network, and start to produce a depth map prediction, as well as update the network weights frame by frame by backproping the error signal of the loss function. In all our experiments we observe that: soon after about a few hundreds of input frames have been processed (usually about 10 s video at 30 fps) the network already starts to produce sensible depth maps, and the loss function appears to converge. We call this process of training on the first a few hundred frames the *network prime* process, and we believe its purpose is to teach the network to learn useful convolutional features for typical visual scenes. Once the network has been “primed”, it can be applied to new previously unseen stereo videos.

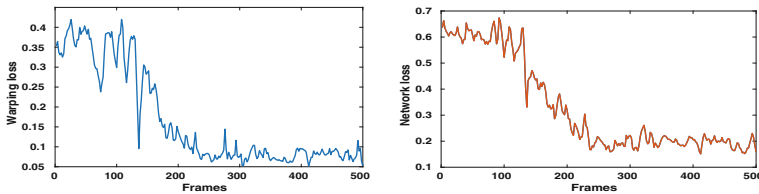


Fig. 4. Typical network convergence curves from random initialization.

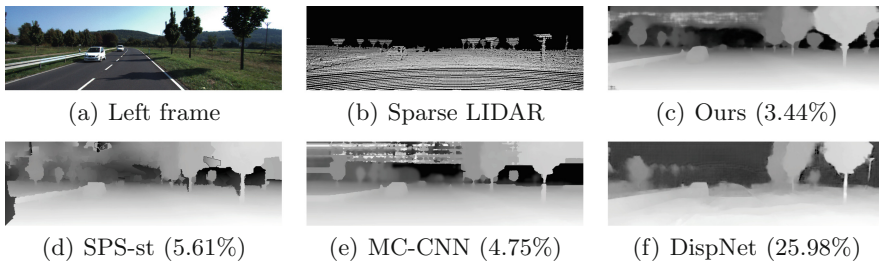


Fig. 5. Our qualitative results on KITTI VO dataset: results are reported on the  $D1_{all}$  error metric. Disparities are transformed to log space for better visualization.

Figure 4 shows typical converge curves for a network during the prime stage. After the prime stage, we randomly select 5 KITTI VO sequences, test our network on them, and compare its performance with three state-of-the-art stereo

methods, including the DispNet [10], MC-CNN [7], and SPS-ST [27]. The first two are deep stereo matching methods, and the last one a traditional (non-deep) stereo method. Quantitative comparison of their performances are reported in Table 1, from which one can clearly see that our OpenStereoNet achieves the best performance throughout all the metrics evaluated. For deep MC-CNN we use a model which was firstly trained on Middlebury dataset for the sake of fair comparison. For SPS-ST, its meta-parameters was also tuned on KITTI dataset. Figure 5 gives some sample visual results for comparison. Note that our method obtains sharp and clean depth discontinuity for cars and trees, better than the other methods. Dispnet, on the other hand, is affected by shadows on road, with many artifacts on the road surface. A quantitative comparison is provided in Table 1. Our method outperforms all baseline methods with a large margin.

**Table 1.** Quantitative results on KITTI VO dataset.

Methods	Abs Rel	Sq Rel	RMSE	RMSE log	D1_all	$\delta < 1.25$
Dispnet [10]	0.122	1.938	8.844	0.189	32.045	0.877
MC-CNN [7]	0.069	1.229	6.002	0.264	8.018	0.932
SPS-st [27]	0.060	1.341	5.521	0.159	4.970	0.957
Ours	<b>0.053</b>	<b>0.540</b>	<b>4.451</b>	<b>0.137</b>	<b>4.403</b>	<b>0.959</b>

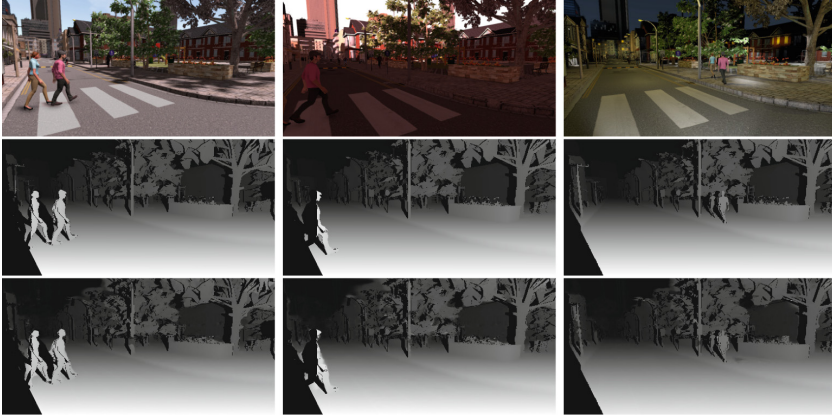
## 5.2 Synthia Dataset

The Synthia dataset [26] contains 7 sequences with different scenarios under different seasons and lighting conditions. Our primary aim for experimenting on Synthia is to analyze our network’s generalization (self-adaption) ability. We create a long video sequence by combining together three Synthia sequences of the same scene but under different seasons and lighting conditions. For example, Fig. 6 shows some sample frames of *Spring, Dawn and Night*. We simply run our network model on this video, and display the disparity error as a function of frames. We run our network on this long sequence. For each condition, we report our quantitative and qualitative results based on the first 250 frames of that sequence.

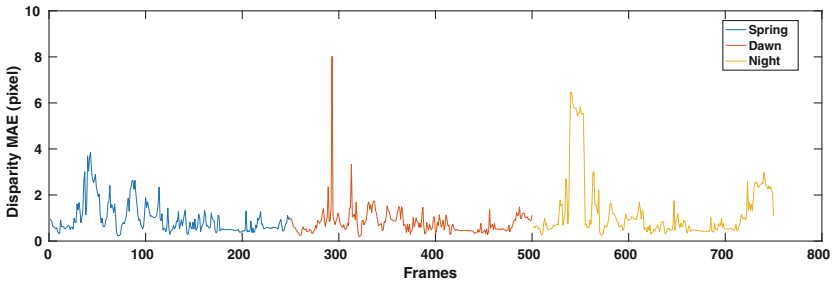
As shown in Fig. 6, our network recovers consistently high quality disparity maps regardless the lighting conditions. This claim is further proved by the qualitative results in Fig. 7. In term of disparity accuracy, our method achieves a Mean Absolute Error (MAE) of 0.958 pixels on the *Spring* scene while the *Dawn* sequence has reached an MAE of 0.7991 pixels and 1.2415 pixels for the *Night* sequence.

## 5.3 Ablation Studies: Effects of the LSTMs and Backprop

There are two mechanisms that contribute to the self-adaptive ability of our OpenStereoNet, *i.e.*, the cLSTMs recurrent blocks and the backprop refinement



**Fig. 6.** Our qualitative results on Synthia: top to bottom: input left image, ground truth disparity, our result. The first column is taken from the *Spring* subset, the middle column is from *Dawn*, and the last column is from *Night*. Our method performs uniformly on different sequences.



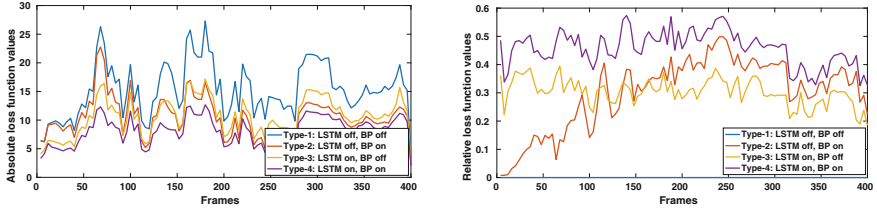
**Fig. 7.** We run our method on a continuous video sequence consisting of the same scene under three different season/lighting conditions (spring, dawn, night). The curve shows the final disparity error as a function of the stereo frame. From this curve it is clear that our network is able to adapt to new scenarios automatically.

process. To understand their respective effects on the final performance of our network, we conduct ablation studies by isolating their operations. To be precise, we have tested the following four types of variants of our full networks: (type-1) remove LSTMs and also disable the backprop process (*i.e.*, the *baseline* network); (type-2) remove LSTMs, but keep backprop on; (type-3) with LSTMs on, without backprop, and (type-4) with both LSTMs on and backprop on (*i.e.*, our full network).

Results by these four types of networks are given in the following curves in Fig. 8. One can clearly see the positive effects of the LSTM units and the backprop. In particular, adding LSTMs has reduced the loss function of the baseline network significantly. In another ablation test, we run the above type-3

**Table 2.** Ablation study on LSTM module on KITTI.

Methods	Abs Rel	Sq Rel	RMSE	RMSE log	D1_all	$\delta < 1.25$
Type-3 net (without LSTMs)	0.066	1.580	5.332	0.167	5.089	0.957
Type-4 net ( with LSTMs )	<b>0.053</b>	<b>0.540</b>	<b>4.451</b>	<b>0.137</b>	<b>4.403</b>	<b>0.959</b>



**Fig. 8.** We run our method on a continuous video sequence. The left figure shows the comparison of absolute losses for the 4 types (note: the lower, the better), while the right figure gives comparison of the relative loss against the the (type-1) baseline network (note: the higher, the better). The y-axis indicates the final disparity errors, and x-axis the input frame-Ids.

network on the previous selected KITTI VO sequence, and list their accuracy in Table 2. From this, one can see that by applying the LSTM module, we have achieved better performance across all error metrics.

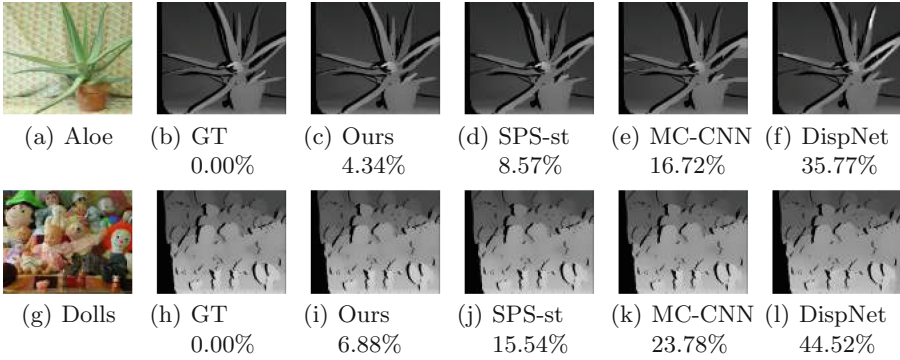
#### 5.4 Middlebury Stereo Dataset

The stereo pairs in the Middlebury stereo dataset [24, 25] are indoor scenes with multiple handcrafted layouts. The ground truth disparities are captured by structured light with higher density and precision than KITTI dataset. We report our results on selected stereo pairs from Middlebury 2005 [24] and 2006 [25] and compare with other baseline methods. In order to evaluate our method on these images, we augmented each stereo pair to a stereo video sequence by simply repeating the stereo pair.

We use *bad-pixel-ratio* as our error metrics used in this experiment, and all results are reported with 1-pixel thresholding. As shown in Fig. 9, our method achieves superior performance than all baseline methods. Other deep learning based methods have even worse performance than the conventional method SPS-st when there is no fine tuning.

#### 5.5 Other Open World Stereo Sequences

To further demonstrate the generalization ability of our OpenStereoNet, we test it on a number of other freely downloaded stereo video datasets from the Internet. Note that our network had never seen these test data before. Below we give some sample results, obtained by our method and by the DispNet, on the Freiburg Sceneflow Dataset [10] and on RDS-Random Dot Stereo.



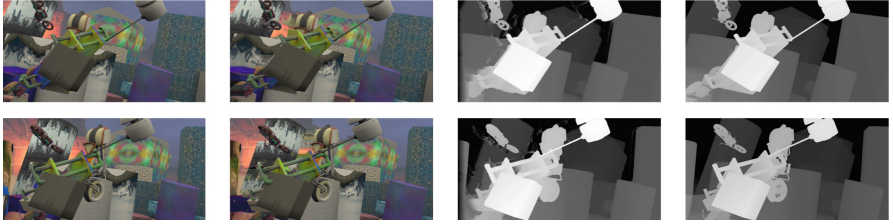
**Fig. 9.** Our results on Middlebury: left to right: left image, ground-truth disparity, our result, SPS-st result, MC-CNN result and DispNet result. We report the *bad-pixel-ratio* at 1-pixel threshold.

**Freiburg SceneFlow Dataset.** We select two stereo videos from the Monkaa and FlyingThings3D dataset [10] and directly feed them into our network. Qualitative results are shown in Figs. 10 and 11 correspondingly. Our network produces very accurate disparity maps when compared with the ground truth disparity maps. Furthermore, the reconstructed color images with the estimated disparity map further prove the effectiveness of our model.

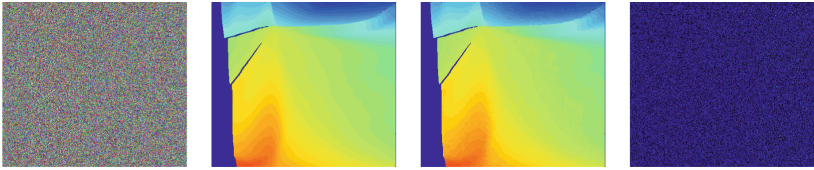
**Random Dot Stereo.** We test the behavior of our OpenStereoNet on random dot stereo images where there is no semantic content in the images. Our network works well, however the DispNet fails miserably as shown in Fig. 12.



**Fig. 10.** Our qualitative results on Monkaa: from top to bottom: left image, reconstructed left image, recovered disparity map. From left to right: frame 11, frame 15, frame 133, frame 143. (Color figure online)



**Fig. 11.** Our qualitative results on FlyingThings3D: from left to right: left image, reconstructed left image, estimated disparity map, and ground truth. (Color figure online)



**Fig. 12.** Test results on RDS (Random dot stereo) images: left to right: input stereo (left image), ground-truth disparity (color coded), our result, result by Dispnet. Our method successfully recovers the correct disparity map, demonstrating its superior generalibility on unseen images. (Color figure online)

## 6 Conclusions and Discussions

This paper addresses a practical demand of deploying stereo matching technique to unconstrained real-world environments with previously unseen or unfamiliar “open-world” scenarios. We envisage such a stereo matching method that is able to take a continuous live stereo video as input, and automatically predict the corresponding disparity maps. To this end, this paper has proposed a deep Recurrent Neural Network (RNN) based stereo video matching method—*OpenStereoNet*. It consists of a CNN Feature-Net, a Match-Net and two convolutional-LSTM recurrent blocks to learn temporal dynamics in the scene. We do notice that finding optical flow (or scene flow) between image frames is yet another feasible paradigm to encode and to exploit temporal dynamics existed in a video sequence. However, we argue optical flow itself is a significant research topic in itself, no less challenging than stereo matching, and a comparison between the two approaches deserves to be a valuable future work.

Our OpenStereoNet does not need ground-truth disparity maps for training. In fact, there is even no clear distinction between training and testing as the network is able to learn on-the-fly, and to adapt itself to never-seen-before imageries rapidly. We have conducted extensive experiments on various datasets in order to validate the effectiveness of our network. Importantly, we have found that our network generalizes well to new scenarios. Evaluated based on absolute per-

formance metrics for stereo, our method outperforms state-of-the-art competing methods by a clear margin.

**Acknowledgements.** Y. Zhong’s PhD scholarship is funded by CSIRO Data61. H. Li’s work is funded in part by Australia ARC Centre of Excellence for Robotic Vision (CE140100016). Y. Dai is supported in part by National 1000 Young Talents Plan of China, Natural Science Foundation of China (61420106007, 61671387), and ARC grant (DE140100180). The authors are very grateful to NVIDIA’s generous gift of GPUs to ANU used in this research.

## References

1. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The kitti vision benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2012)
2. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comp. Vis.* **47**(1–3), 742 (2002)
3. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(2), 328341 (2008)
4. Kendall, A., et al: End-to-end learning of geometry and context for deep stereo regression. In: Proceedings of the IEEE International Conference on Computer Vision, October 2017
5. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W., Woo, W.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Proceedings of the Advances in Neural Information Processing Systems, NIPS 2015, Cambridge, MA, USA, pp. 802–810. MIT Press (2015)
6. Janai, J., Gney, F., Behl, A., Geiger, A.: Computer vision for autonomous vehicles: problems, datasets and state-of-the-art. Arxiv (2017)
7. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.* **17**(1), 22872318 (2016)
8. Luo, W., Schwing, A.G., Urtasun, R.: Efficient deep learning for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5695–5703, June 2016
9. Seki, A., Pollefeys, M.: SGM-Nets: semi-global matching with neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
10. Mayer, N., et al.: A large dataset to train convolutional networks for disparity, optical OW, and scene OW estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2016
11. Pang, J., Sun, W., Ren, J.S., Yang, C., Yan, Q.: Cascade residual learning: a two-stage convolutional neural network for stereo matching. In: International Conference on Computer Vision - Workshop on Geometry Meets Deep Learning (2017)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, June 2016
13. Garg, R., Vijay Kumar, B.G., Carneiro, G., Reid, I.: Unsupervised CNN for single view depth estimation: geometry to the rescue. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 740–756. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46484-8\\_45](https://doi.org/10.1007/978-3-319-46484-8_45)

14. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
15. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
16. Xie, J., Girshick, R., Farhadi, A.: Deep3D: fully automatic 2D-to-3D video conversion with deep convolutional neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 842–857. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46493-0\\_51](https://doi.org/10.1007/978-3-319-46493-0_51)
17. Zhong, Y., Dai, Y., Li, H.: Self-supervised learning for stereo matching with self-improving ability. [arXiv:1709.00930](https://arxiv.org/abs/1709.00930) (2017)
18. Wang, C., Buenaposada, J.M., Zhu, R., Lucey, S.: Learning Depth from Monocular Videos using Direct Methods. ArXiv e-prints, November 2017
19. Luo, Y., et al.: Single View Stereo Matching. ArXiv e-prints, March 2018
20. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. *Trans. Neural Netw.* **5**(2), 157166 (1994)
21. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 17351780 (1997)
22. Fragkiadaki, K., Levine, S., Felsen, P., Malik, J.: Recurrent network models for human dynamics. In: Proceedings of the IEEE International Conference on Computer Vision, Washington, D.C., USA, pp. 4346–4354. IEEE Computer Society (2015)
23. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600612 (2004)
24. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, p. 18, June 2007
25. Hirschmüller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society (2007)
26. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.: The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
27. Yamaguchi, K., McAllester, D., Urtasun, R.: Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 756–771. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10602-1\\_49](https://doi.org/10.1007/978-3-319-10602-1_49)