Faster Self-adaptive Deep Stereo*

Haiyang Wang[1], Xinchao Wang[2], Jie Song[1], Jie Lei[1], and Mingli Song[1]

[1] Zhejiang University, Hangzhou, China
{haiyang.wang,sjie,ljaylei,brooksong}@zju.edu.cn
[2] Stevens Institute of Technology, New Jersey, USA
xinchao.wang@stevens.edu

Abstract. Fueled by the power of deep learning, stereo vision has made unprecedented advances in recent years. Existing deep stereo models, however, can be hardly deployed to real-world scenarios where the data comes on-the-fly without any ground-truth information, and the data distribution continuously changes over time. Recently, Tonioni et al. proposed the first real-time self-adaptive deep stereo system (MADNet) to address this problem, which, however, still runs at a relatively low speed with not so satisfactory performance. In this paper, we significantly upgrade their work in both speed and accuracy by incorporating two key components. First, instead of adopting only the image reconstruction loss as the proxy supervision, a second more powerful supervision is proposed, termed Knowledge Reverse Distillation (KRD), to guide the learning of deep stereo models. Second, we introduce a straightforward yet surprisingly effective Adapt-or-Hold (AoH) mechanism to automatically determine whether or not to fine-tune the stereo model in the online environment. Both components are lightweight and can be integrated into MADNet with only a few lines of code. Experiments demonstrate that the two proposed components improve the system by a large margin in both speed and accuracy. Our final system is twice as fast as MADNet, meanwhile attains considerable superior performance on the popular benchmark datasets KITTI.

Keywords: Stereo matching · Self-supervised · Real-time

1 Introduction

Environment depth information is key to many applications such as autonomous driving and mobile robotics. Compared with technologies such as LIDAR, Structured Light and Time-of-Flight, stereo is competitive in practical application scenarios due to its lower cost, higher resolution and better universality for almost any environment. Many traditional stereo algorithms have been proposed in recent decades. However, most of these algorithms are limited to specific conditions (e.g., occlusions, texture-less areas, photometric distortions). Since Mayer

* This work is supported by National Natural Science Foundation of China (61976186), the Major Scientfic Research Project of Zhejiang Lab (No.2019KD0AC01) and Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies.

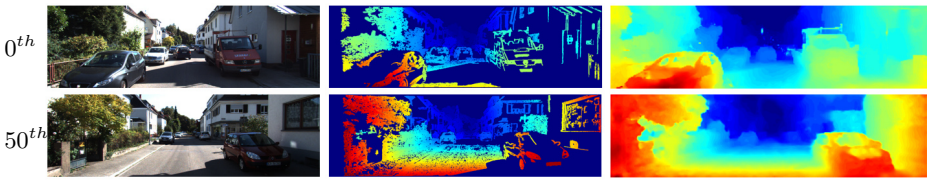


Fig. 1. Disparity maps predicted by AoHNet on KITTI sequence [9]. Given the source images (a), AoHNet exploits the traditional stereo method [10] as a second supervision (b) to quickly adapt to the new real scenario (c).

et al. [1] proposed the first end-to-end stereo network DispNet, using convolutional neural networks (CNNs) to regress depth maps from images directly has become a dominant paradigm for stereo matching and is followed by many state-of-the-art stereo methods [2–5].

However, existing deep learning-based stereo methods always suffer from the domain shift problem where accuracy drops dramatically when the domain of testing data is different from that of training data [6]. Fine-tuning on the target domain can solve the above problem, but obtaining aligned label data often requires a large cost. [6, 7] propose to generalize deep stereo to novel domains without the need of labels. However, they assume that the data of the target domain is available in advance while in the real world the data domain usually changes over time. Recently, Tonioni et al. [8] proposed the first real-time self-adaptive deep stereo system (MADNet) towards addressing the above problem. They cast the adaption as a continuous learning process and proposed a lightweight, fast and modular architecture (MADNet) together with a tailored training algorithm, termed Modular ADaptation (MAD) to improve the running speed. To bypass the unsupervised problem, they adopt image reconstruction loss as the proxy objective to train the model.

In this work, we significantly upgrade their method in both accuracy and speed based on two key insights. Firstly, MADNet only adopts image reconstruction loss as the proxy supervision. Albeit effective to some degree, this proxy objective requires a relatively long time (about 900 frames) to adapt the deep stereo model to a new scenario. To address this problem, a second supervision objective is proposed, termed Knowledge Reverse Distillation (KRD), to enhance the adaption process of the deep stereo model. In KRD, our goal is somewhat opposite to traditional KD: we leverage the noisy predictions from lightweight traditional models (teachers) as supervision to guide the learning of the deep stereo model (student), and make the student surpass the teacher. As the KRD loss is more calibrated to the goal than image reconstruction loss, the adaption process with the KRD loss takes less time (about 400 frames in our experiments) than with solely the image reconstruction loss. Fig. 1 shows that with the supervision of KRD, AoHNet can quickly adapt to the new real scenario from a synthetic scenario and solve most of the mistakes within 50 frames.

Secondly, it can be found that the ceaseless fine-tuning of the deep stereo model in the online environment has two huge weaknesses. One is dropping the network running speed to a third, which is unbearable for real-world applications. The other is hurt the accuracy of the system, as the model will become over-fitted if it continues to train in the adapted environment. Thus we devise an extremely lightweight yet surprisingly effective Adapt-or-Hold (AoH) mechanism. The AoH mechanism is implemented by a Deep Q-Network (DQN) based on reinforcement learning. In every frame, the DQN directly decide whether to adapt or hold according to the input state. As it is really micro so incurs nearly no additional overhead into the system. Experiments conducted on KITTI [9] demonstrate that with AoH mechanism, our method achieves superior accuracy than MADNet. Even the deep stereo model adapts itself on only 10% of the frames, which meanwhile speeds up our system to about 29 FPS, one time faster than MADNet.

In summary, we make the following contributions:

- We introduce the Knowledge Reverse Distillation, a more powerful supervision than image reconstruction loss, to transfer deep stereo models to new scenarios without any ground-truth information.
- We propose an Adapt-or-Hold mechanism that allows the deep stereo model to hold or adapt itself automatically in the online environment. This mechanism improves not only the speed but also the accuracy of the system.
- Experimental results demonstrate that the proposed online system works about one time faster than its predecessor MADNet, meanwhile attains significantly superior accuracy on the popular benchmark KITTI.

2 Related Work

Here we briefly review some of the most related topics, including traditional stereo algorithms, supervised stereo algorithms, self-supervised depth estimation and deep reinforcement learning.

Traditional stereo algorithms. Researches have recently proposed many methods for the stereo matching, which finds its application in a wide domain of computer vision tasks [11–14]. Such algorithms usually involve four steps: i) matching cost computation, ii) cost aggregation, iii) disparity optimization/computation, and iv) disparity refinement. Scharstein et al. [15] divided these algorithms into two parts: local algorithms and global algorithms. Local algorithms firstly define a support window and an evaluation function, and then aggregate matching costs over the window. Global algorithms usually establish a loss function that combines matching cost terms and smoothness terms on the whole image, and then solves it using graph-based methods [16–18]. Thus, they often perform better than the local algorithm in quality and stability. However, global algorithms often rely on multiple iterations, which are challenging to be done in real-time. An excellent trade-off between accuracy and execution time is represented by ELAS [10] which is a Bayesian approach proposing a generative probabilistic model for stereo matching. It can compute accurate disparity of images at frame rates close to real-time.

Supervised stereo methods. Zbontar and LeCun [19] firstly applied deep learning in stereo vision by replacing the matching cost computation with a Siamese CNN learning similarity on small image patches. However, it still needs a series of post-processes. DispNet [1] was the first end-to-end stereo network that broke the pipeline in [15] by regressing the depth map from two images directly. GCNet [20] leverages the knowledge of the problem's geometry to form a cost volume and learns to incorporate contextual information using 3-D convolutions over this volume. Other works [2–5] following end-to-end stereo architectures outperform previous methods by building more complex architectures. However, the methods above all focus on accuracy with little consideration of speed and require a lot of training data with ground truth, which is not suitable in practical application scenarios [21–26].

Self-supervised depth estimation. Depth estimation in a self-supervised way is popular recently as it overcomes the shortcoming of requiring a large number of annotations. Some methods [27–29] make use of image reconstruction loss to drive the network in an unsupervised way. This loss is calculated from warping different views, coming from stereo pairs or image sequences. [6] proposed to adopt the off-the-shelf stereo algorithms together with a confidence estimator network CCNN [30] to fine-tune the network offline. A Deep Recurrent Neural Network with LSTM blocks [31] was proposed, which was able to adapt between different scenarios seamlessly, but it doesn't take speed into account for requiring 0.8-1.6 seconds for inference. Tonioni et al. [8] proposed the first real-time self-adaptive deep stereo system which only used image reconstruction loss. However, as shown in [32], the photometric loss is not a good choice for stereo problems. Different from [8], we propose an additional supervision obtained by the traditional algorithm to enhance the adaption process. What's more, a straightforward and efficient way is proposed to extract the high confidence pixels of the traditional algorithm without the need for additional networks.

Deep reinforcement learning. Since the first deep reinforcement learning model [33], termed Deep Q-Network (DQN), successfully learn control policies in Atari game, deep reinforcement learning has attracted the attention of many researchers. [34] adds a Target Q network to compute the target values, which reduces the correlations between the action-values and the target values. Other methods such as prioritized experience replay [35], double DQN [36], dueling DQN [37] are proposed to improve the performance of deep reinforcement learning. Because of the excellent performance of deep reinforcement learning in control policy problems, we make use of it to decide whether the stereo model needs to be fine-tuned or not in the online environment.

3 Methodology

Starting with a pre-trained deep stereo model (e.g., pre-trained on the synthetic data [1]), our goal is to deploy this stereo model to real-world applications where 1) no ground-truth information is available along with the raw data; 2) the

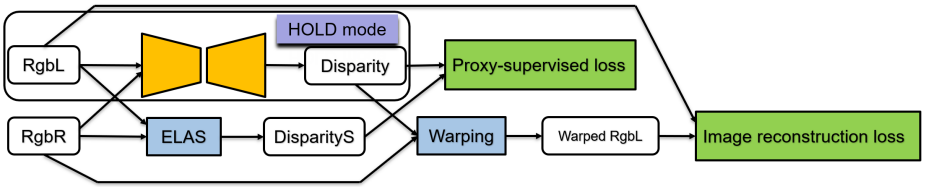


Fig. 2. An illustration of the self-supervised adaption framework. The white rectangles represent image data, the orange rectangle represents the network to be trained, the blue rectangles represent the traditional algorithm and the green rectangles represent the loss functions.

data distribution in the current scenario is different from that of the training data used for pre-training the deep stereo model; 3) the scenarios may change continuously over the time. Thus, a *real-time*, *self-supervised* and *self-adaptive* system is needed to tackle all the aforementioned problems.

3.1 The Overall Framework

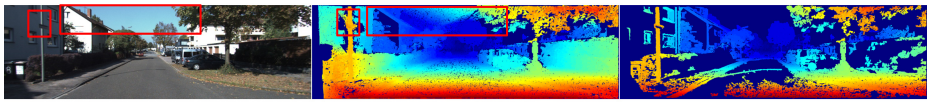
The overall framework of the proposed system is illustrated in Fig. 2. In order to avoid the loss of speed and accuracy caused by adapting the network all the time, we devised the AoH mechanism to enable the system to work in two modes: the ADAPT mode and the HOLD mode.

When the deep stereo model has been fully adapted to the current scenario, our system works in the HOLD mode. In this mode, the deep stereo model conducts only the inference process: fed with paired frames, outputting the stereo predictions. As no back-propagation is involved, the system works especially fast in this mode. If the scenario changes, the system will switch to the ADAPT mode. In this mode, the system will not only infer the stereo information of the current frames but also adapt itself to the new scenario by computing the loss and perform back-propagation. To speed up the adaption process of the deep stereo model for being rapidly switched to the more efficient HOLD mode, we propose the KRD objective to facilitate the adaption of the deep stereo model.

Now, the detailed description of the proposed KRD objective and the AoH mechanism is provided.

3.2 Knowledge Reverse Distillation

KRD is devised to remedy the incapability of the image reconstruction loss for adapting the deep stereo model to new scenarios. The goal of KRD is that by learning from the noisy predictions of the lightweight teacher, the student can overcome the shortcomings of the teacher so thus surpass the teacher after learning. We adopt the ELAS algorithm [10] as the teacher. ELAS is a lightweight, fast and general-purpose stereo algorithm that does not require any adaptation



(a) (b) (c)

Fig. 3. Examples of proxy labels computed by ELAS, left image (a), disparity map processed only by left-right consistency detection (b), and the sparse disparity map after low-texture areas removal (c).

to be deployed in different scenarios. Though ELAS generates accurate disparity at most pixels, there are still many unreliable predictions. A network trained on the raw output of the traditional algorithm would learn the inherent shortcomings of it. Therefore, two remedies are proposed to help the student stereo model overcome the weakness of ELAS and finally surpass it in performance. Firstly, a lightweight confidence measure is proposed to quantify the reliability of predictions of every pixel efficiently. With this confidence measure, predictions with low confidence will be masked so that they will not contribute to the total loss. Secondly, we don't abandon the image reconstruction loss as well. The deep stereo model is supervised by both the KRD loss and image reconstruction loss. The synergy between these two losses pulls the model out of their weaknesses. Now, a more detailed description of these two remedies will be given.

Confidence Measure Specifically, we mask two types of predictions (shown in Fig. 3): 1) those do not pass the left-right consistency detection, like the pixels removed in (b); and 2) those of pixels which lie in low-texture regions, e.g., the sky in (c). The texture is calculated as the sum of the Sobel filter values of the pixels within the defined window, which is defined in [10]. The predictions of low-texture regions are masked for two considerations. On the one hand, Elas has the same disadvantages as traditional algorithms, that is, poor performance in the low-texture areas. For example, the area near the street lamp on the left and the sky area connected to the tree on the right in Fig. 3, which can not be detected in left-right consistency detection. On the other hand, from the perspective of model learning, pixels in low-texture regions are usually large in amount and their stereo predictions provide redundant supervision. A large amount of redundant supervision will overwhelm the objective, making the student learning bias to low-texture regions.

Given an input stereo pair I_l and I_r, we obtain the left and right disparity maps D_l and D_r. For all pixels p in the original disparity map D_l, if the texture of the pixel $texture(p)$ is lower than the threshold β or the left-right consistency detection $|D_l(p) - D_r(p - D_l(p))|$ is larger than the threshold δ, the disparity values are masked:

$$M(p) = \begin{cases} 0, & texture(p) \leq \beta \\ 0, & |D_l(p) - D_r(p - D_l(p))| \geq \delta. \\ 1, & otherwise \end{cases} \tag{1}$$

Synergy between KRD and Image Reconstruction Loss We adopt both the KRD loss and the image reconstruction loss to adapt our stereo model to new scenarios. The overall objective is:

$$L = L_R + \lambda L_{KRD}, \tag{2}$$

where L_{KRD} represents the KRD loss and L_R represents the image reconstruction loss. λ is the hyper-parameter trading off these two loss. The KRD loss is defined to be the average ℓ_1 distance between disparity maps from our model D and ELAS algorithm D_{ELAS}:

$$L_{KRD} = \frac{1}{|M|}|M \odot (D - D_{ELAS})|. \tag{3}$$

M is the binary matrix introduced in Equation 1, and $|M|$ denotes the number of all valid pixels in D_{ELAS}. Image reconstruction loss is obtained by computing the discrepancy between the left image I_l and the reconstructed left image I'_l from the right image and the left disparity map. Following [28], we use a combination of ℓ_1 and single scale SSIM [38] as our image reconstruction loss L_R.

$$L_R = \frac{1}{N}\sum_p \alpha \frac{1 - SSIM(I_l(p), I'_l(p))}{2} + (1-\alpha)\cdot|I_l(p) - I'_l(p)|. \tag{4}$$

N denotes the number of all pixels in the image and p represents each pixel.

3.3 The Adapt-or-Hold Mechanism

Keeping the model always adapting in the online environment heavily reduces the real-time responsiveness of our system. Worse, it also decreases the accuracy of the system. Here we introduce the proposed AoH mechanism which enables our system to automatically switch between the ADAPT mode and the HOLD model, which significantly improves the system in both speed and accuracy.

Markov Decision Process We define AoH mechanism as a Markov Decision Process that contains state, action, and reward. According to the input state, the agent chooses one action from action space and gets the corresponding reward. Here are the definitions. The state should contain enough information to enable the agent to select a good action. Image reconstruction loss can reflect whether the model parameters are suitable in the online environment, and the computation is small compared with KRD. Therefore, we define the state as the image reconstruction loss of the last ten frames. The action space is clearly defined as two discrete options, ADAPT or HOLD. Reward is used to evaluate the results of the action taken by the agent. Here, image reconstruction loss L_R and time consumption T are both considered. If the agent chooses the action of ADAPT, T is set to -1, otherwise it is 1. By adjusting the weight κ between image reconstruction loss and time consumption, the percentage of adaption can be controlled. Here. The reward equation is as follows:

$$R = \frac{1}{e^{L_R}} + \kappa T. \tag{5}$$

Deep Q-Network We use the DQN proposed in [34] to train the agent and improve the performance by referring to [36]. The DQN is a combination of deep learning and Q-learning, using DNN to predict the Q value $Q(s, a)$ of each action a for the input state s. There are two networks in DQN, one called target_net, to get the value of q_{target}, and the other, called eval_net, to get the value of q_{eval}. The target_net parameters θ^- are only updated with the eval_net parameters θ every few steps and are keep fixed between individual updates. This reduces the correlations between the q_{eval} and the q_{target} and significantly improved the stability of learning. The q_{target} is defined as:

$$q_{target} = r + \gamma Q(s', argmax_a(s^{'}, a; \theta); \theta^-), \tag{6}$$

where r is the reward, γ is the discount factor, $s^{'}$ represents the state of next step. The loss function is defined as the difference between q_{target} and q_{eval}:

$$Loss(\theta) = E[(r + \gamma Q(s^{'}, argmax_a(s^{'}, a; \theta); \theta^-) - Q(s, a; \theta))^2]. \tag{7}$$

The training data is randomly extracted from the memory buffer, where each record $(s, a, r, s^{'})$ includes the current state, action, corresponding reward, and next state. The size of memory buffer is limited, so the records will be overwritten as the network updates. By randomly extracting records from memory for learning, the correlation between experiences is disrupted, making the neural network updating more efficient.

4 Experiment

In this section, we first describe the implementation details and then conduct benchmark comparisons with our teacher algorithm ELAS, a supervised algorithm DispNet and a online self-adaptive algorithm MADNet. After that, we performed some ablation experiments to prove the effectiveness of KRD and AOH. For the KRD, we compare the performance of different loss functions. These experiments are made on two different kinds of datasets, one is KITTI 2012 and KITTI 2015 which provides discrete images, but the label has a higher density. The other is a continuous video of [9], which is more suitable for online learning. In the meantime, we make an experiment to analyze the sparsity and accuracy of the preserved predicted labels under different texture threshold. As for AOH, we designed a detailed comparison experiment, including not only fine-tuning in advance and adapting all the time, but also three other typical strategies are designed. The details will be shown below.

4.1 Implementation Details

We adopt MADNet [8] as the backbone of the proposed AoHNet. In order to achieve the real-time performance of the ELAS algorithm, we reduce the input image to a quarter, calculate the disparity map, and finally linear interpolation

to the original resolution. In our experiments, the texture threshold β is set to 50, left-right consistency threshold δ is 2 and the hyper-parameter λ trading off two loss objectives is 0.1. Following [28], α in L_R is 0.85.

The micro DQN only contains two hidden layers with 20 and 10 units respectively, followed by ReLU activation. The output layer has 2 nodes with no activation. The action chosen is determined according to the Q-function, which has the maximum value. The DQN is trained on the KITTI raw [9] in advance. The weight coefficient κ between image reconstruction loss and time consumption is set to 0.02. As for training, We set 5,000 as memory buffer size and 32 as batch size. 1000 pre-training steps are preceded to gather experience replay buffer. ϵ-greedy and the discount factor γ are set to 0.9. ϵ-greedy means the action is selected according to the learned network by the probability of ϵ and is randomly selected by the probability of $1 - \epsilon$. The target_net updates every 1000 steps. An RMSProp optimizer was used with a learning rate of 0.001.

Unless otherwise specified, for all experiments involved in this paper, the weights pre-trained on synthetic data [1] are used as a common initialization and evaluate the proposed method on real datasets KITTI [9, 39, 40]. As there is the same number of labeled images and unlabeled images in the training set of KITTI 2012 and KITTI 2015. We use unlabeled images to train and labeled images to test. For all experiments, both average End Point Error (EPE) and the percentage of pixels with disparity error larger than 3 (D1-all) are analyzed. Since the image format of each sequence is different, a central crop with a size of 320×1216 is extracted from each frame as proposed in [8]. Finally, we use Adam as the optimizer with a constant learning rate equal to 0.0001.

4.2 Benchmark Comparison

In this section, we conduct benchmark comparisons to demonstrate the superiority of AoHNet. We compare AoHNet with the following competitors: (1) ELAS [10], our teacher, a fast and relatively accurate traditional method; (2) DispNet [1], a supervised algorithm that uses ground-truth labels for training directly; (3) MADNet [8], the first online stereo method. (4) Recent self-supervised stereo methods. The network parameters of DispNet is obtained from [8] which have been fine-tuned on KITTI. For both MADnet and AoHNet, the networks are pre-trained on the synthetic data [1] and then fine-tuned on the unlabeled data of KITTI in a self-supervised way. Finally, we evaluate the performance of all algorithms on the labeled data of KITTI 2012 and KITTI 2015.

Experimental results are provided in Table 4.2. It can be seen that: (1) Compared with deep stereo models, the traditional algorithm ELAS [10] produces a much larger error in both KITTI 2012 and KITTI 2015; (2) The proposed model, albeit trained online in a self-supervised way, outperforms DispNet trained with ground-truth information, which means that our self-supervised method can even be comparable to some supervised methods; (3) Compared with MADNet, our approach exhibits significantly superior performance in both precision and speed thanks to the KRD loss and the AoH mechanism. (4) The SOTA [43] achieves the smallest D1-all error. Despite the higher accuracy, it runs 20 times slower

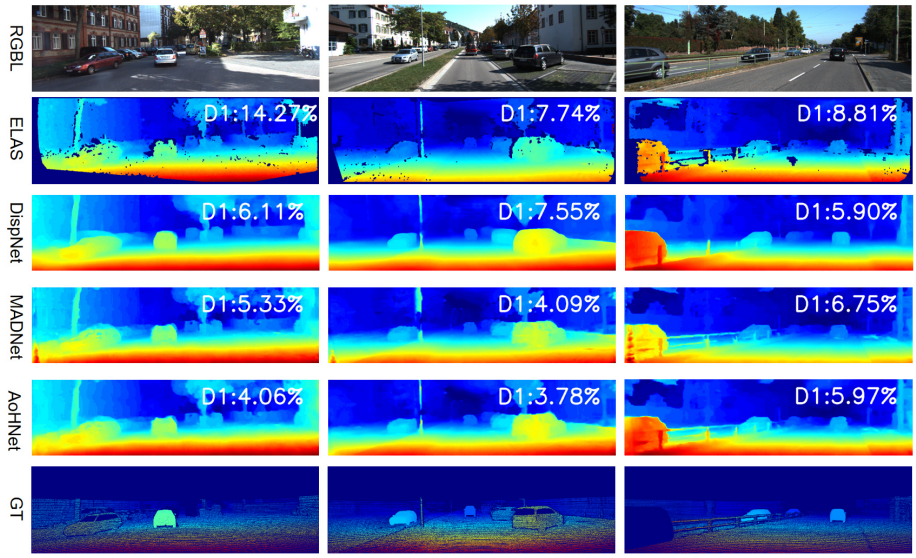


Fig. 4. Quantitative comparison of different methods on KITTI 2015.

Table 1. Comparison of different algorithms in KITTI 2012 and KITTI 2015.

KITTI	ELAS [10]	DispNet [1]	MADNet [8]	Zhou [41]	Li [42]	Aleotti [43]	AoHNet
2012 D1-all	17.12	9.53	9.25	9.91	8.60	-	8.64
2015 D1-all	14.78	7.87	8.53	-	8.98	4.06	7.76
FPS	3.34	16.67	14.26	2.56	1.37	2.44	28.95

than AoHNet. Besides, as it requires a monocular completion network to provide proxy labels in addition to the conventional algorithm, making it cumbersome to deploy in a real-time changing environment.

Fig. 4 visualizes some examples produced by the above algorithms in three different scenarios on the KITTI 2015 dataset, from left to right are "City", "Resident" and "Road". The D1-all error is shown in the right corner of the disparity maps. As is shown in Fig. 4, the disparity maps generated by DispNet has a precise shape and smooth edges, but the overall error is somewhat significant. AoHNet yields lower total error and preserves better results in detail. For example, the isolation barrier in the middle of the road on the right image.

4.3 Ablation Study

In this section, we conduct ablation studies to validate the effectiveness of the KRD loss and AoH mechanism proposed in the paper.

Table 2. Comparison of different loss objectives on KITTI.

Model	Color	KITTI 2012			KITTI 2015		
		Frames	D1-all(%)	EPE	Frames	D1-all(%)	EPE
MADNet	Color	330	9.47	1.51	250	9.63	1.71
+KRD	Color	290	9.19	1.37	210	9.41	1.75
AoHNet$^-$	Color	240	8.94	1.41	200	9.25	1.65
MADNet	Gray	150	7.49	1.31	190	7.98	1.39
+KRD	Gray	110	7.11	1.25	160	7.60	1.36
AoHNet$^-$	Gray	**90**	**7.05**	**1.19**	**70**	**7.58**	**1.32**

With versus without KRD In our method, we propose to adopt the KRD loss, together with the image reconstruction loss, to guide the learning of the deep stereo model. Table 2 reports the comparison of different loss objectives. MADNet [8] only uses image reconstruction loss as the self-supervised loss. "+KRD" means only using KRD loss and AoHNet$^-$ means to adopt KRD loss together with image reconstruction loss, but AoH mechanism is removed. Here "Frames" means the number of frames required by the network to be adapted to KITTI from synthetic data [1] (If the accuracy is not improved after ten consecutive evaluations, the network is considered to have been adapted to the new scenario).

In [8], all experiments were performed on color images. However, we find that the network performs better on gray images than color images. Gray images means three channels are the same. We provide experimental results on both color images and gray images. On KITTI 2012, MADNet requires 330 frames to be fully adapted to the new scenario on the color images. However, less than half of the frames (150 frames) are needed on the gray images, and the D1-all error is lower.

To find out whether image reconstruction loss is still needed as supervision, we also do experiments that only used KRD loss. Table 2 shows that the performance of only using KRD loss is better than that of only using image reconstruction loss but slightly worse than that of the combination of the two losses. It may be because KRD loss only provides sparse supervision, and image reconstruction loss can help to learn the missing part of them.

Due to the use of a more powerful supervision KRD, no matter in color images or gray images, AoHNet$^-$ is superior to MADNet in both the adaption speed and the accuracy. Finally, we only need less than 100 frames (90 frames in KITTI 2012 and 70 frames in KITTI 2015) to adjust the network from one domain to another. This implies that KRD loss not only improves the accuracy of the network but also makes the network adapting to new scenarios faster.

We also make experiments on a continuous video to analyze the effectiveness of KRD. Fig. 5 plots the D1-all error across frame for MADNet and AoHNet$^-$ on the 2011_09_30_drive_0028_sync sequence which is a 2500 frames residential video of KITTI [9]. The three color lines represent the three patterns. The red line presents the performance of MADNet fine-turned offline on KITTI, which is used as a benchmark for comparison. The green and blue lines represent MADnet

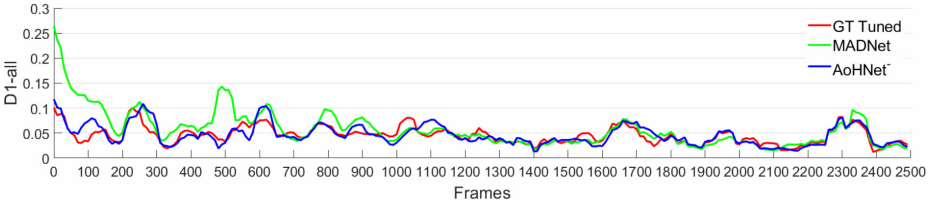


Fig. 5. D1-all error across frames on the 2011_09_30_drive_0028_sync sequence.

Table 3. The performance of different texture thresholds on KITTI 2015.

Sparsity	NO	LR	S20	S50	S80	S100
Density(%)	77.56	71.71	61.33	43.64	32.54	27.20
D1-all(%)	4.43	4.79	4.23	**3.42**	3.43	3.48
Deep-D1-all(%)	7.72	7.72	8.28	**7.68**	7.80	8.05

and AoHNet$^-$ respectively. The parameters of them are initialized on synthetic data [1]. Both lines improve their performances by back-propagation. After a period of adaption, they achieve comparable performance to the offline fine-tuned model (red). It shows that MADNet (green) needs about 900 frames to reach the similar performance to fine-tuning, while AoHNet$^-$ (blue) only needs less than 400 frames with the help of KRD loss.

Influence of Different Sparsity We make use of left-right consistency detection and low-texture area removal to filter the noisy pixels. Different texture thresholds are set to analyze the sparsity and accuracy of the preserved labels on 200 images from KITTI 2015 [39]. As shown in the Table 3, "Density" represents the percentage of valid pixels in disparity maps. "No" represents the original outputs of ELAS, which the density of valid pixel is 77.56%. "LR" indicates that only left-right consistency detection is performed, and the D1-all error is largest. This means that only by left-right consistency detection, there are still many outliers that can't be detected. "S" plus number represents different texture threshold. For example, "S20" represents that the texture threshold is 20. As the texture threshold increases, from 20 to 100, the density of the valid pixels decreases. However, even when the texture threshold is set to 100, the density is still higher than 19.73% of the ground truth provided by KITTI 2015.

As the texture threshold increases, the D1-all error of the sparse disparity maps first decreases and then increases. When the texture threshold is 50, the D1-all error is minimized. This is because traditional methods tend to perform poorly in low-texture areas, so at low texture thresholds, more percentage of the low-confidence pixels are removed, causing D1-all error to decrease. As the texture threshold increases, the percentage of high-confidence pixels is increasing, more pixels of high-confidence are removed, so the D1-all error increases.

Table 4. Performance of AoH and other comparisons on the KITTI2015 sequence.

Model	MADNet			AoHNet					
	NO FT	GT	FULL	FULL	R-10	F250	E1-10	HAND	AoH
D1-all(%)	47.82	4.65	4.90	4.70	4.83	5.94	4.70	4.64	**4.51**
EPE	12.46	1.22	1.23	1.24	1.24	1.29	1.23	1.23	**1.19**
FPS	39.48	39.48	14.26	13.93	29.85	31.34	29.90	24.19	28.95

We further experimented with the above data for network training. The networks are fine-tuned in the same way on the different sparsity disparity maps and then use the ground-truth labels of KITTI 2015 for evaluation. The results are shown in the "Deep-D1-all" row. Finally, the texture threshold is set to 50. Under this threshold, the valid pixel density of the disparity map is 43.64%, and the accuracy is 96.58%.

With versus without AoH We have demonstrated the effectiveness of KRD, which adapts the network to a new scenario in less than 100 frames. However, fine-tuning the network all the time in the online environment comes with the side effect that back-propagation slows the network speed down to a third, which is unbearable for real application scenarios. What's more, it leads to another shortcoming that the model may become over-fitted if it continues training when it has already been adapted to the environment. To overcome the above problems, we propose an Adopt-or-Hold Mechanism that can automatically decide when the network needs adaption and when to stop adaption.

Table 4 shows performance of MADNet and AoHNet on a 2500 frames residential video of KITTI [9]. "NO FT" means that the network parameters are trained from synthetic data and have not been fine-tuned on KITTI. The error is large, which indicates that deep learning-based methods produce poor performance when the domain of data changes. "GT" means the results attained by the model that is fine-tuned offline on the target domain. "FULL" means that the network is fine-tuned all the time during the video. When AoHNet and MADNet work in the mode of "FULL", their speeds both drop to nearly one-third of the inference speeds. The error of AoHNet is smaller than MADNet due to the help of KRD loss. The AoH Mechanism performs excellent. Since the adaption is only made on 10% of the video frames in AoH Mechanism, the speed is one time faster compared to the "FULL" adaption. What's more, the D1-all error is even smaller than that of fine-tuning the network all the time.

According to the analysis, the AoH Mechanism updates about 250 frames across the entire video, so we designed three other methods for comparison. "R-10" means randomly choose 10 percent of frames for adaption. "F250" means only updating the network on the first 250 frames, and the rest frames only make an inference. "E1-10" means updating the network 1 frames every 10 frames, a total of 250 frames are updated on the whole video. "HAND" is the method designed manually with image reconstruction loss. The error of "F250" is the largest, which is probably because of the environmental changes during the video.

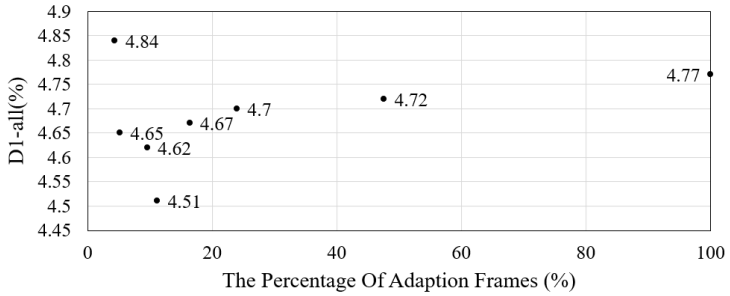


Fig. 6. The relationship between the percentage of adaption frames and the average D1-all error on the 2011_09_30_drive_0028_sync sequence.

"HAND" is a little better than "R-10" and "E1-10". AoH produces superior results to all other methods, verifying the effectiveness of the AoH mechanism.

Furthermore, we did an experiment to analyze the relationship between the percentage of frames for adaption and the D1-all error on the whole video. The results are shown in the Fig. 6. As the percentage of the adaption frames increases, D1-all error decreases first and then increases. When the percentage of the adaption frames is about 11%, the error becomes the smallest. It implies that once the network has been adapted to the new environment, the adaption process should stop in time to avoid the model becoming over-fitted.

5 Conclusion and Future Work

In this paper, we proposed AoHNet, a real-time, self-supervised and self-adaptive online framework that can automatically adapt to new environments without the need for ground-truth labels. Two key components are introduced to improve the precision and the speed of deep stereo models: the Knowledge Reverse Distillation and the Adapt-or-Hold mechanism. Knowledge Reverse Distillation leverages the noisy predictions from lightweight traditional models (teachers) as supervision to guide the learning of the deep stereo model (student) and makes the student surpass the teacher. Adapt-or-Hold (AoH) mechanism based on Deep Q-Netwok can automatically determines when the deep stereo model adapts or holds in online environment. Experiments demonstrate that the proposed approach outperforms existing methods significantly in both speed and precision.

We believe that the direction of deep stereo matching in the future is that without the need for aligned labels, the network can adjust itself online according to the changing environment, rather than training a specific model for a particular scene. Besides, more attention will be paid on the embedded side, to bring the newest technology to the real practical applications.

References

1. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: IEEE Conference on Computer Vision and Pattern Recognition. (2016)
2. Pang, J., Sun, W., Ren, J.S., Yang, C., Yan, Q.: Cascade residual learning: A two-stage convolutional neural network for stereo matching. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 887–895
3. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 5410–5418
4. Guo, X., Yang, K., Yang, W., Wang, X., Li, H.: Group-wise correlation stereo network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3273–3282
5. Zhang, F., Prisacariu, V., Yang, R., Torr, P.H.: Ga-net: Guided aggregation net for end-to-end stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 185–194
6. Tonioni, A., Poggi, M., Mattoccia, S., Di Stefano, L.: Unsupervised adaptation for deep stereo. In: IEEE International Conference on Computer Vision. (2017)
7. Pang, J., Sun, W., Yang, C., Ren, J., Xiao, R., Zeng, J., Lin, L.: Zoom and learn: Generalizing deep stereo matching to novel domains. In: IEEE Conference on Computer Vision and Pattern Recognition. (2018)
8. Tonioni, A., Tosi, F., Poggi, M., Mattoccia, S., Stefano, L.D.: Real-time self-adaptive deep stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 195–204
9. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32** (2013) 1231–1237
10. Geiger, A., Roser, M., Urtasun, R.: Efficient large-scale stereo matching. In: Asian conference on computer vision, Springer (2010) 25–38
11. Wang, X., Li, Z., Tao, D.: Subspaces indexing model on grassmann manifold for image search. IEEE Transactions on Image Processing **20** (2011) 2627–2635
12. Qiu, J., Wang, X., Maybank, S.J., Tao, D.: World from blur. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. (2019) 8493–8504
13. Wang, X., Türetken, E., Fleuret, F., Fua, P.: Tracking interacting objects using intertwined flows. IEEE Transactions on Pattern Analysis and Machine Intelligence **38** (2016) 2312–2326
14. Lan, L., Wang, X., Hua, G., Huang, T.S., Tao, D.: Semi-online multi-people tracking by re-identification. International Journal of Computer Vision **128** (2020) 1937–1955
15. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision **47** (2002) 7–42
16. Klaus, A., Sormann, M., Karner, K.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: 18th International Conference on Pattern Recognition (ICPR'06). Volume 3., IEEE (2006) 15–18
17. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions via graph cuts. Technical report, Cornell University (2001)
18. Yang, Y., Qiu, J., Song, M., Tao, D., Wang, X.: Distilling knowledge from graph convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2020)

19. Zbontar, J., et al.: Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research **17** (2016) 2
20. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: IEEE International Conference on Computer Vision. (2017) 66–75
21. Wang, X., Türetken, E., Fleuret, F., Fua, P.: Tracking interacting objects optimally using integer programming. In: European Conference on Computer Vision (ECCV). (2014) 17–32
22. Yu, X., Liu, T., Wang, X., Tao, D.: On compressing deep models by low rank and sparse decomposition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
23. Yang, E., Deng, C., Li, C., Liu, W., Li, J., Tao, D.: Shared predictive cross-modal deep quantization. IEEE transactions on neural networks and learning systems **29** (2018) 5292–5303
24. Yin, X., Wang, X., Yu, J., Zhang, M., Fua, P., Tao, D.: FishEyeRecNet: A multi-context collaborative deep network for fisheye image rectification. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018)
25. Deng, C., Yang, E., Liu, T., Li, J., Liu, W., Tao, D.: Unsupervised semantic-preserving adversarial hashing for image search. IEEE Transactions on Image Processing **28** (2019) 4032–4044
26. Wang, J., Huang, S., Wang, X., Tao, D.: Not all parts are created equal: 3d pose estimation by modeling bi-directional dependencies of body parts. In: IEEE International Conference on Computer Vision (ICCV). (2019)
27. Garg, R., BG, V.K., Carneiro, G., Reid, I.: Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: European Conference on Computer Vision, Springer (2016) 740–756
28. 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) 270–279
29. Ye, J., Ji, Y., Wang, X., Ou, K., Tao, D., Song, M.: Student becoming the master: Knowledge amalgamation for joint scene parsing, depth estimation, and more. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
30. Poggi, M., Mattoccia, S.: Learning from scratch a confidence measure. In: BMVC. (2016)
31. Zhong, Y., Li, H., Dai, Y.: Open-world stereo video matching with deep rnn. In: European Conference on Computer Vision (ECCV). (2018) 101–116
32. Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Loss functions for image restoration with neural networks. IEEE Transactions on Computational Imaging **3** (2016) 47–57
33. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
34. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518** (2015) 529
35. Schaul, T., Quan, J., Antonoglou, I., Silver, D.: Prioritized experience replay. arXiv preprint arXiv:1511.05952 (2015)
36. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: Thirtieth AAAI conference on artificial intelligence. (2016)
37. Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., De Freitas, N.: Dueling network architectures for deep reinforcement learning. arXiv preprint arXiv:1511.06581 (2015)

38. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13** (2004) 600–612
39. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3061–3070
40. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2012) 3354–3361
41. 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) 1851–1858
42. Li, A., Yuan, Z.: Occlusion aware stereo matching via cooperative unsupervised learning. In: Asian Conference on Computer Vision, Springer (2018) 197–213
43. Aleotti, F., Tosi, F., Zhang, L., Poggi, M., Mattoccia, S.: Reversing the cycle: self-supervised deep stereo through enhanced monocular distillation. arXiv preprint arXiv:2008.07130 (2020)