Scientific Research Publishing

Journal of Computer and Communications, 2023, 11, 83-98
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Adaptive Recurrent Iterative Updating Stereo Matching Network

Qun Kong, Liye Zhang*, Zhuang Wang, Mingkai Qi, Yegang Li

School of Computer Science and Technology, Shandong University of Technology, Zibo, China
Email: qkong1026@163.com, *zhangliye@sdut.edu.cn

How to cite this paper: Kong, Q., Zhang, L.Y., Wang, Z., Qi, M.K. and Li, Y.G. (2023) Adaptive Recurrent Iterative Updating Stereo Matching Network. *Journal of Computer and Communications*, **11**, 83-98.
https://doi.org/10.4236/jcc.2023.113007

Received: March 4, 2023
Accepted: March 28, 2023
Published: March 31, 2023

Abstract

When training a stereo matching network with a single training dataset, the network may overly rely on the learned features of the single training dataset due to differences in the training dataset scenes, resulting in poor performance on all datasets. Therefore, feature consistency between matched pixels is a key factor in solving the network's generalization ability. To address this issue, this paper proposed a more widely applicable stereo matching network that introduced whitening loss into the feature extraction module of stereo matching, and significantly improved the applicability of the network model by constraining the variation between salient feature pixels. In addition, this paper used a GRU iterative update module in the disparity update calculation stage, which expanded the model's receptive field at multiple resolutions, allowing for precise disparity estimation not only in rich texture areas but also in low texture areas. The model was trained only on the Scene Flow large-scale dataset, and the disparity estimation was conducted on mainstream datasets such as Middlebury, KITTI 2015, and ETH3D. Compared with earlier stereo matching algorithms, this method not only achieves more accurate disparity estimation but also has wider applicability and stronger robustness.

Keywords

Stereo Matching, Whitening Loss, Feature Consistency, Convolutional Neural Network, GRU

1. Introduction

Stereo matching technology is a fundamental problem in computer vision, which aims to obtain depth information of the 3D scene generated by left-right stereo image pairs, and has been widely used in fields such as robot navigation, autonomous driving, 3D reconstruction, and augmented reality [1]. The core

task is to find the matching relationship between corresponding pixels in the two images, *i.e.*, to find the corresponding point of each pixel in the right image from the left image, and calculate the depth of the scene through the disparity information of these matching points. Therefore, the main problems to be solved in stereo matching are the correctness and accuracy of the matching points.

In traditional methods, stereo matching is summarized into four steps [2]: cost calculation, cost aggregation, disparity calculation, and disparity optimization. In recent years, with the rapid development of deep learning, more and more scholars at home and abroad have gradually used deep learning methods to replace the four steps in traditional methods, ultimately forming the popular end-to-end stereo matching network. The disparity estimation obtained by these end-to-end stereo matching networks has greatly improved in underdetermined areas such as weak texture and discontinuous regions compared to traditional methods. However, the generalization performance of stereo matching networks is still the main challenge for applying network structures to real-world scenarios. The generalization ability of a model refers to its ability to adapt to new data after training. The model learns the underlying patterns behind the data, and the trained network can also give appropriate output for data with the same pattern. Currently, the common method to achieve generalization ability is domain generalization based on domain-invariant features. Existing domain generalization methods can be simply divided into three categories: data manipulation [3], representation learning [4], and policy learning [5]. Some stereo matching networks have already obtained domain-invariant features by performing feature matching. DSMNet [6] designs two trainable neural network layers that can perform domain generalization well, and by regulating the distribution of the learned representations, the network maintains feature invariance to differences. CFNet [7] integrates multiple low-resolution dense cost volumes to guide the network to learn invariant geometric scene information from different datasets, expanding the receptive field for capturing global representations. Reference [8] proposes the MS-Net network, which replaces deep learning-based feature extraction with traditional matching functions and confidence measures, shifting the learning process from the color space to the matching space to prevent overgeneralization of specific dataset features. The above works transform the input to the domain-invariant feature space, reducing dependence on specific features in the dataset and exhibiting stronger robustness.

Based on the research ideas of the above model, this paper proposes a more widely applicable stereo matching network. In response to the problem of decreasing cross-domain feature consistency, a whitening loss function is introduced during feature extraction. As the loss function decreases, the stereo matching network relies less on matching-unrelated information to form feature representations, thus extending the stereo matching network to real-world scenarios and improving the model's generalization ability.

This paper is organized as follows: Section 2 discusses related work; Section 3 introduces the TUNet architecture and the improved adaptive stereo matching

network ATUNet; Section 4 presents experimental results and analysis; and finally, Section 5 draws conclusions based on the findings.

2. Related Work

2.1. Stereo Matching Networks Based on Deep Learning

In recent years, with the rapid development of Convolutional Neural Network (CNN) [9], as well as the significant improvement in computing power of various hardware devices with technological advancement, more and more scholars at home and abroad have been using deep learning methods to reduce the phenomenon of mismatching in the ill-posed areas of stereo matching algorithms. Scholars have used CNN to replace individual steps in traditional binocular stereo matching algorithms, dividing deep learning-based binocular stereo matching algorithms into non-end-to-end stereo matching algorithms and end-to-end stereo matching algorithms [10].

Compared with traditional stereo matching algorithms, non-end-to-end stereo matching algorithms can obtain good disparity effects in complex scenes, greatly promoting the development of stereo matching algorithms. However, non-end-to-end stereo matching algorithms only use local information for cost computation, lacking global information, which makes them still challenging in occlusion, low texture, and repetitive texture areas. Meanwhile, non-end-to-end stereo matching algorithms use a series of cascaded post-processing steps to refine disparities, which makes the training process complicated and difficult to directly optimize the entire stereo matching process. Therefore, using end-to-end stereo matching algorithms has become a research hotspot in stereo matching algorithms in recent years.

The end-to-end stereo matching algorithm inputs a pair of left-right stereo images into a convolutional neural network and directly outputs accurate disparity after training. In 2016, Mayer *et al.* [11] proposed the first end-to-end stereo matching network which is called DispNet, which used a convolutional neural network to extract features, obtained the feature correlation mapping between the left and right feature maps, and output disparity of different resolutions in multiple transposed convolution layers. They also contributed a large dataset called Scene Flow, generated through synthetic techniques, for network training. Du *et al.* [12] input foreground segmentation information into the AMNNet network together, improving the generalization performance of the stereo matching network. PSMNet [13] proposed using spatial pyramid pooling and dilated convolution to expand the receptive field, which can combine global environmental information into image features. At the same time, they repeated the stacked 3D hourglass network from coarse to fine and fine to coarse to increase the utilization rate of global information. Cai *et al.* [8] pointed out that the poor generalization performance of the stereo network is caused by the network's strong dependence on image appearance and suggested using combinations of matching functions for feature extraction.

The stereo matching algorithm based on the end-to-end deep learning framework consists mainly of four modules: feature extraction, cost volume construction, cost aggregation, and disparity regression, which is consistent with the basic process of traditional stereo matching algorithms. In traditional stereo matching algorithms, manually designed feature descriptors such as SIFT [14] and SURF [15] are usually used for feature extraction. Although these feature descriptors cannot solve problems in specific scenes (such as textureless areas, overexposed areas, and repetitive problem areas), they rarely affect the disparity calculation effect due to dataset transformations. Therefore, the feature extraction layer in the deep learning framework can be considered as a key factor in improving the cross-dataset generalization ability of stereo matching networks. The feature extraction layer captures the style information of images by extracting the correlation between feature channels, which has been further explored in style transfer, image-to-image translation, and other fields. Recently, a selective whitening method was proposed in literature [16] to remove sensitive style information in the dataset, thereby reducing the learning of significant features in the dataset, where the style information selection depends on manually designed photometric transformations. Inspired by selective whitening, this paper chooses information that is sensitive to changes in stereo viewpoints, not just dependent on photometric transformations. This is because in the left and right views of stereo matching, the image transformation is not only photometric, but also involves changes in the scene, etc.

2.2. Factors Affecting the Generalization Ability of Stereo Matching

The key to enhancing the generalization ability of stereo matching networks is to improve their adaptation ability from one dataset to another. Generally speaking, there are significant differences in color, contrast, texture, and scene between stereo images before and after cross-dataset, which can cause the training dataset features learned by deep stereo matching networks to not be well adapted to other datasets, ultimately resulting in erroneous matching results when the network estimates disparities for other datasets.

In order to verify the phenomenon of erroneous disparity estimation due to large image differences before and after cross-dataset in the model, this paper uses the mainstream PSMNet network model for cross-domain feature visualization. First, PSMNet is trained to convergence on the Scene Flow dataset, and then the results of the feature extraction layer from different datasets are visualized and compared in testing. As shown in **Figure 1**, two sets of stereo image pairs from the Scene Flow and KITTI 2015 datasets were selected, and they were transmitted to the PSMNet network to obtain their feature visualization results.

The output of the feature extraction part of the PSMNet network is a feature tensor of size C × 1/4 H × 1/4 W, where C is the number of feature channels. By analyzing the feature differences before and after cross-domain comparison in the same channel dimension, the information difference of the features can be

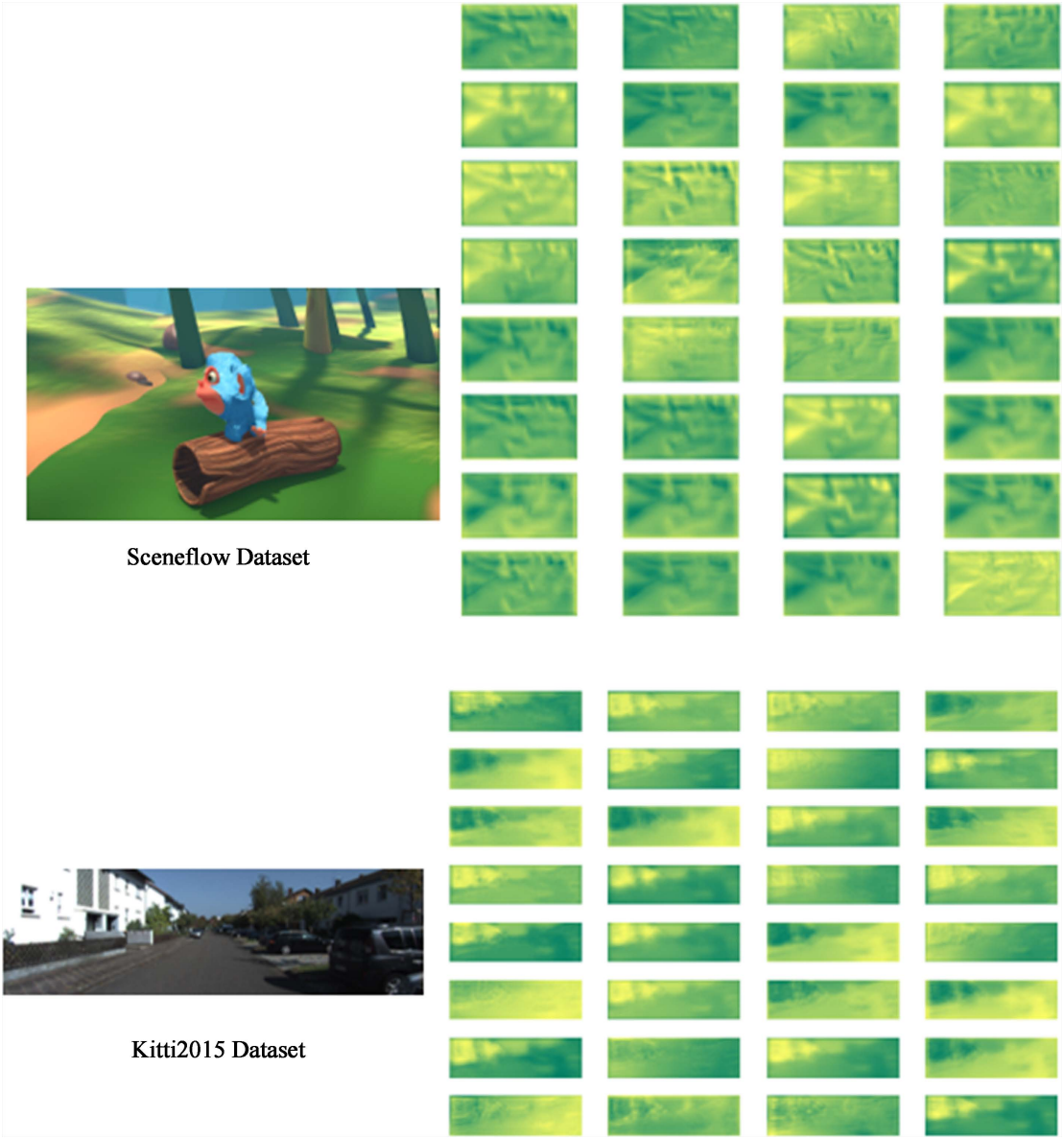

Figure 1. Feature visualization.

observed. In order to more intuitively observe the feature transformation, this paper uses the method of mean [17] to determine the feature differences of the network before and after cross-domain. The specific method of the mean method is as follows: first, calculate the mean of each feature of the network on the first dataset, which can be calculated by averaging the output of the network; then, use the trained network to perform forward propagation on the second dataset and record the output of each feature to further calculate the mean of each feature in the second dataset; finally, compare the mean of each feature in the first and second datasets, if the mean of a certain feature in the first dataset is significantly different from that in the second dataset, then it indicates that the feature has differences on different datasets. In each channel, the mean on the pixel dimension (H, W) is defined as the following formula:

$$u_p(x) = \frac{1}{HW}\sum_h^H \sum_w^W FV_{hw} \tag{1}$$

where, H and W represent the pixel-wise positions, H represents the height of the pixel dimension, W represents the width of the pixel dimension, and ε is a small constant added to avoid division by zero in the denominator.

In the generalization experiment, we randomly selected the left images from 105 pairs of stereo images in Scene Flow and KITTI 2015 datasets, and then calculated the mean values of the two datasets in the same channel. As shown in Figure 2, the black line represents the mean distribution of channel 1 in Scene Flow dataset, and the blue line represents the mean distribution of channel 1 in KITTI 2015 dataset. The mean value curves show that for a group of images with low sensitivity to color changes, the feature means are relatively close. Conversely, for a group of images with high sensitivity to color changes, the feature means vary greatly. To better explain this, we refer to larger changes as "sensitive changes" and smaller changes as "insensitive changes". Examples of sensitive and insensitive changes are shown in Figure 3.

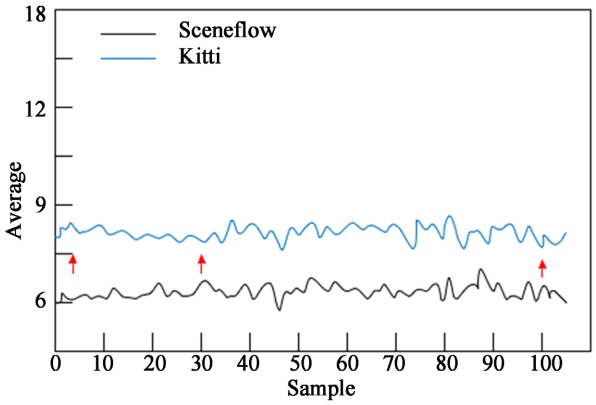

Figure 2. Characteristic channel average curve.

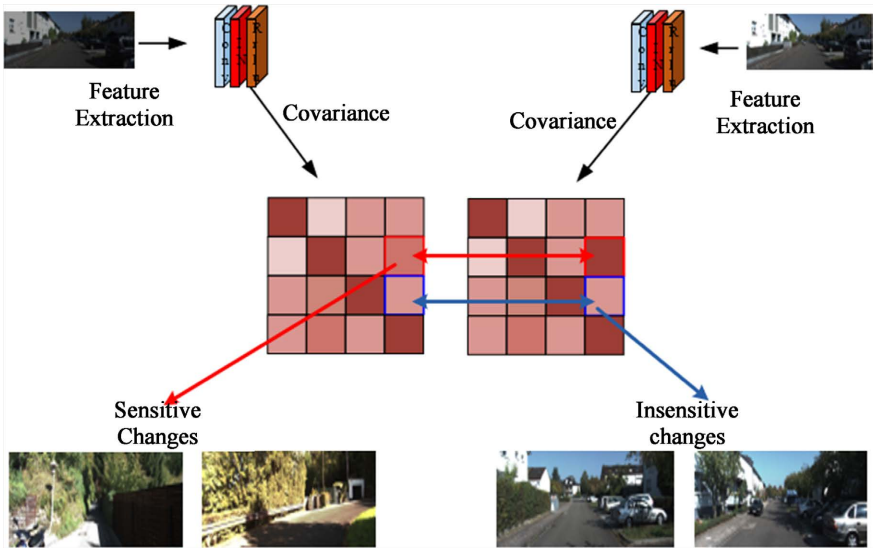

Figure 3. Feature changes.

3. Method

3.1. Transformer-Based Iterative Update Stereo Matching Network

Transformer-based Iterative Update Stereo Matching Network (TUNet) framework is shown in **Figure 4**. The extracted left and right feature maps are transformed into a more easily matched feature that is related to context and position through feature transformation. The cost volume is constructed through similarity calculation and then iteratively updated through GRU to obtain the disparity estimation result.

3.1.1. Feature Transform Module

During feature extraction, a pair of stereo images I_l and I_r are input to two feature extraction networks with weight sharing. The architecture of the feature extraction network consists of a series of residual layers and subsampled layers that extract left and right feature at different resolutions. Then, the attention mechanism from the Transformer algorithm [18] is added to aggregate global contextual information by using alternate self-attention and cross-attention layers, so that the feature maps processed by the Transformer can produce dense matching in low texture areas. Meanwhile, relative positional encoding is added to the feature vectors to greatly enhance the position dependency of the feature maps. Linear Transformers [19] are used to reduce computational complexity during the alternate calculation process of self-attention and cross-attention.

3.1.2. Disparity Iterative Update Module

The disparity update is performed using Gated Recurrent Unit (GRU) [20], which is a type of recurrent neural network (RNN) [21] unit used for modeling sequential data. The specific steps are as follows: starting from the initial disparity of $d_0 = 0$, the disparity estimation is performed by producing an update direction Δd in each iteration, which is fed into the next iteration to compute the current disparity estimation: $d_{k+1} = \Delta d + d_{k+1}$. The disparity estimation is calculated by inputting the left feature maps, correlations, and the updated hidden state into the GRU, which updates the hidden state and further predicts the new disparity based on the updated hidden state.

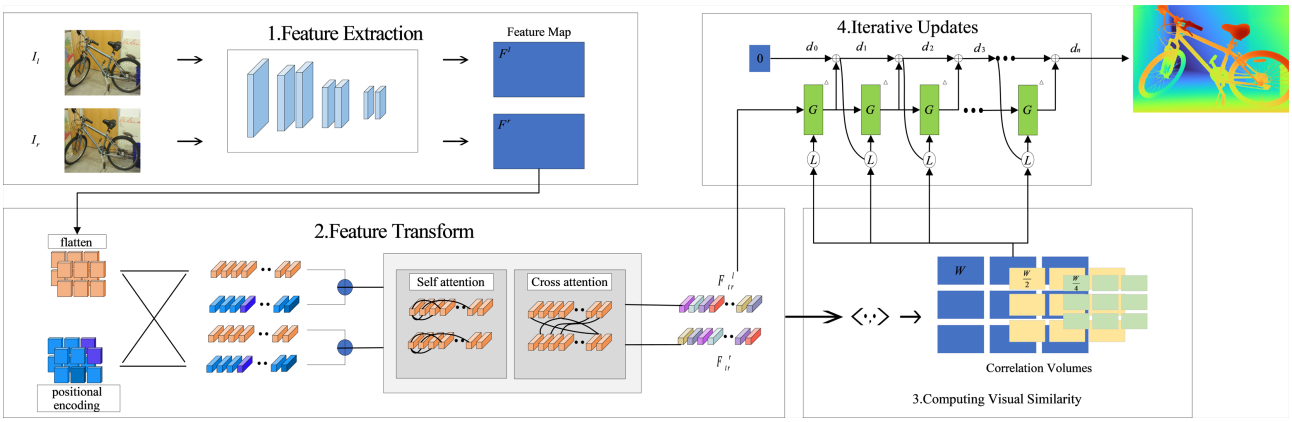

Figure 4. Network architecture of TUNet algorithm.

3.2. Improved Adaptive Recurrent Iterative Update Stereo Matching Network

Building on the TUNet stereo matching network in Section 3.1, this paper introduces an adaptive recurrent iterative updating stereo matching network—ATUNet, through incorporating a whitening loss module in the feature extraction module. By suppressing feature consistency, the model's generalization performance is improved.

3.2.1. Whitening Loss Module

Stereo matching networks typically use Batch Normalization (BN) [22] to normalize features. During training, BN uses batch-wise statistics to normalize features, while during inference, it uses the statistics of the entire training dataset. This leads to the over-reliance of stereo matching networks on the training dataset, making them more sensitive to dataset shifts. To extend feature consistency across different datasets, Instance Normalization (IN) [23] layers are used to replace some BN layers. Unlike BN layers, the IN layer normalizes each sample across its channel dimension, thus avoiding any dependence on the data. For each sample $X \in \mathbb{R}^{C\times H\times W}$, the IN layer normalization process is as follows:

$$\hat{X}_i = \frac{1}{\sigma_i}\left(X_i - \mu_i\right) \tag{2}$$

In the equation above, u_i and σ_i represent the mean and variance, respectively, and C represents the index of the feature channel. Although the IN layer normalizes features within the local neighborhood, it does not consider the correlation between different channels. To further improve the consistency of feature representation, the whitening loss module can remove the redundancy between features by suppressing the feature covariance components that are sensitive to changes in color and other factors in the dataset, as shown in **Figure 5**.

Firstly, feature extraction is performed, and then the extracted features are subjected to the following computation:

Setp 1: compute the feature vector covariance matrix $\Sigma\left(\hat{X}\right)$:

$$\Sigma\left(\hat{X}\right) = \frac{1}{HW}\left(\hat{X}\right)\left(\hat{X}\right)^{\mathrm{T}} \tag{3}$$

Setp 2: calculate the feature covariance matrix $\Sigma_n\left(\hat{X}^l\right)$ between the left image feature vector variance $\Sigma_n\left(\hat{X}^r\right)$ and its corresponding right image feature vector variance $V_{i,j}$:

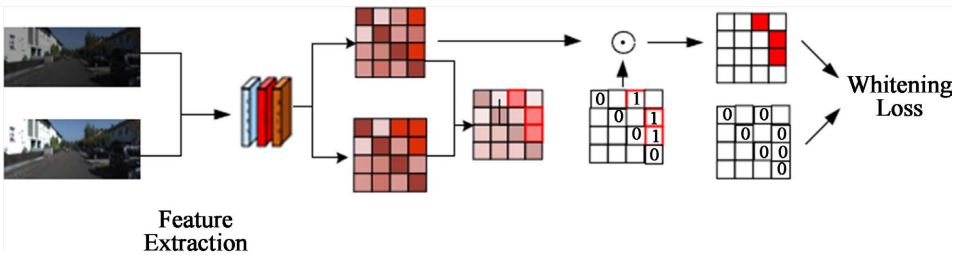

Figure 5. Whitening loss.

$$\begin{aligned}\mu_{\Sigma_n} &= \frac{1}{2}\left(\Sigma_n\left(\hat{X}^l\right)+\Sigma_n\left(\hat{X}^r\right)\right)\\ V_{i,j} &= \frac{1}{2N}\sum_{n=1}^{N}\left(\left(\Sigma_n\left(\hat{X}^l\right)-\mu_{\Sigma_n}\right)^2+\left(\Sigma_n\left(\hat{X}^r\right)-\mu_{\Sigma_n}\right)^2\right)\end{aligned} \tag{4}$$

where covariance matrix $V_{i,j}$ between the *i*-th and *j*-th channels represents the sensitivity to viewpoint changes. If the covariance elements between the left and right features have high variances, these elements are considered to be components that are sensitive to viewpoint changes, that is, the correlation between the two features is high. Therefore, these covariance elements should be considered in the whitening loss. To obtain these values, the k-means [24] method can be used to cluster the covariance matrix $V_{i,j}$ and calculate the selective mask.

Setp 3: compute the selective mask $\tilde{M}_{i,j} \in \mathbb{R}^{C\times C}$:

$$\tilde{M}_{i,j} = \begin{cases} 1, & V_{i,j} \in \delta_p \\ 0, & \text{other} \end{cases} \tag{5}$$

3.2.2. Whitening Loss Module

Compute whitening loss on the left image feature vector variance:

$$L^W = \frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}\left\|\Sigma_\gamma\left(\hat{X}^l\right)\odot\tilde{M}\odot\hat{M}\right\|_1 \tag{6}$$

where $\hat{M}$ is an upper triangular matrix, Γ represents the number of layers for loss calculation, and γ represents the corresponding intermediate layer.

Finally, the loss function of the stereo matching network with the introduced whitening loss is calculated as follows:

$$L = L^{disp} + L^W \tag{7}$$

where L^{disp} is the disparity loss function, which is calculated using the smooth L_1 loss, as shown in Equation (8):

$$L^{disp} = \sum_{i=1}^{N}\gamma^{N-i}\left\|d_{gt}-d_i\right\|_1 \tag{8}$$

By introducing whitening loss, the stereo matching network can not only reduce its dependence on irrelevant information but also further improve the consistency and generalization of its feature representation. Since the differences between left and right stereo images are usually limited to specific physical features, such as diffuse reflection of light, the network model can learn these generalized physical features from limited training data. This enables the network to better adapt and perform when facing new datasets and scenes, thereby improving its reliability and stability in practical applications. In addition, the introduction of whitening loss can also help the network learn more discriminative features, further enhancing its matching performance and accuracy.

4. Experiments

In this experiment, the proposed stereo matching network model was trained only on the Scene Flow dataset, and then tested on the KITTI 2015, Middlebury,

and ETH3D datasets to evaluate its cross-dataset generalization ability. The network was built using the PyTorch framework on an NVIDIA RTX A6000 48 G, and the stereo matching network model was trained using a batch size of 8 and the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$). Prior to training, the input images were randomly cropped to 512 × 256. Finally, the network was trained for 15 epochs on the Scene Flow dataset with a learning rate of 0.001.

4.1. Datasets

4.1.1. Scene Flow

The Scene Flow dataset contains high-resolution images and the optical flow and depth information between adjacent frames of multiple indoor and outdoor scenes. Each scene includes approximately 40 adjacent frames with a resolution of 1024 × 436 pixels. These frames were captured at a frame rate of 15 frames per second. Each scene in this dataset contains various types of objects such as vehicles, pedestrians, buildings, etc., with diverse directions and speeds of movement. Therefore, this dataset is very useful for testing the motion and depth estimation capabilities of various types of objects in different scenes.

4.1.2. KITTI 2015

The KITTI 2015 dataset contains image sequences of multiple real-world scenes, each captured by a stereo camera setup comprising of left and right cameras. The dataset includes approximately 200 sequences, each of which contains high-resolution images and accurate depth and optical flow information collected by a system of sensors such as laser scanners and cameras. The images in the dataset cover various scenes, including city streets, highways, rural roads, etc., and exhibit diverse movements and shape changes of objects such as vehicles, pedestrians, buildings, etc.

4.1.3. Middlebury

The Middlebury dataset provides image sequences of various resolutions, including Full, Half, and Quarter resolutions, which can be used to test and evaluate algorithms of different accuracy. The images in the dataset cover various scenes, including indoor, outdoor, natural, and artificial scenes, where objects exhibit diverse features such as shape, size, motion, and color. Additionally, this datas*et al*so provides multiple evaluation metrics, such as flow and disparity error, flow and disparity visualization, which can be used to assess the accuracy and performance of algorithms.

4.1.4. ETH3D

The ETH3D dataset includes multiple sets of image sequences captured by multiple cameras, including 27 stereo image pairs for training and 20 stereo image pairs for testing. Each sequence contains complete camera intrinsic and extrinsic parameters and highly accurate 3D point cloud information. Additionally, the dataset provides depth maps, surface normal maps, and surface texture maps in various formats, which can be used to test and compare different 3D reconstruc-

tion algorithms.

4.2. Feature Generalization Analysis

In order to verify the generalization ability of the model, this paper defines the mean of the same feature channels extracted by the model in different datasets as the feature similarity. The formula for this is:

$$D_m = \mu_S - \mu_R \tag{8}$$

where μ_S represents the response difference of the mean, and μ_R and D_m represent the feature means of two different datasets, respectively.

This paper randomly selected 100 images from the Scene Flow and KITTI 2015 datasets and used different methods to visualize response differences. The results are shown in Figure 6, where the horizontal axis represents the number of 32 feature channels, and the vertical axis represents the response difference amplitude. The smaller the amplitude of the response difference, the closer the mean of the information extracted by the feature extraction module in the two datasets.

From Figure 6, it can be seen that different stereo matching models have significant fluctuations in response differences across different datasets. Among them, the stereo matching model with the addition of the whitening loss module has response differences that fluctuate up and down by no more than 0.5 across datasets, and its fluctuation curve is relatively smoother compared to the currently popular PSMNet and the Iterative Stereo Matching Network.

4.3. Contrast Experiment

To evaluate the effectiveness of the whitening loss module, three methods for improving generalization ability, including instance normalization, domain normalization, and the whitening loss module, were added to the model in Section 3.1 for experimental comparison. The threshold error matching rate was used as the evaluation method, where the threshold was 3PX for the KITTI 2015

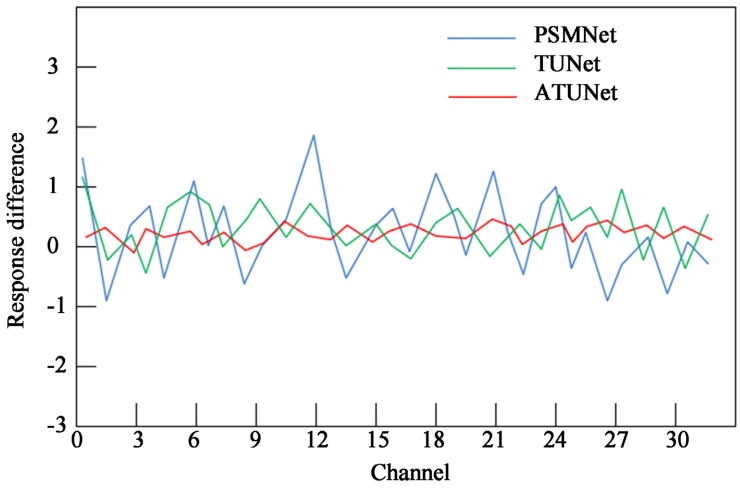

Figure 6. Response difference.

dataset and 2PX for the Middlebury dataset. As shown in Table 1, ATUNet with the added whitening loss module achieved a 35.05% improvement in accuracy at 3PX on the KITTI 2015 dataset and a 14.6% improvement in accuracy at 2PX on the Middlebury dataset compared to the original TUNet model. Compared with the other two methods for improving generalization ability, introducing the whitening loss module into the original model helps the model to better generalize to other datasets.

To further validate the superiority of the proposed approach, this paper compared the adaptive cyclic iterative updating stereo matching network ATUNet with cross-dataset stereo matching networks and other state-of-the-art end-to-end stereo matching networks on three real datasets. It can be seen that among all stereo matching network models, ATUNet achieved a leading performance compared with other Scene Flow pre-trained stereo matching networks and traditional stereo matching algorithms. As shown in Table 2, the 2 px pixel error rate reached 18.1 on the Middlebury dataset, which was 30.06% higher than PSMNet and 15.02% higher than DSMNet, the most advanced cross-dataset invariant stereo matching network. On the KITTI 2015 dataset, the 3 px pixel error rate reached 6.3, which was 68.18% higher than PSMNet and 3.07% higher than DSMNet. Figure 7 and Figure 8 show the disparity visualization results of ATUNet on the Middlebury and KITTI 2015 datasets.

4.4. Experimental Test on ETH3D

In this section, the effectiveness of the proposed method was evaluated on the ETH3D stereo matching dataset, and ATUNet was compared with various traditional and deep learning stereo matching methods. The model was trained only on the Scene Flow dataset and then tested on the test set provided by the ETH3D

Table 1. Whitening loss module.

Methods	KITTI 2015	Middlebury
TUNet model	9.7	21.2
+Instance normalization layer	8.3	19.1
+Domain normalization layer	8.1	18.3
+Whitening loss (ATUNet)	6.3	18.1

Table 2. KITTI 2015 and Middlebury generalization ability.

Models	Middlebury	KITTI 2015
GWCNet	32.5	22.7
PSMNet	25.9	19.8
GANet	24.3	11.7
DSMNet	21.3	6.5
Our	18.1	6.3

Figure 7. Visualization effect on middlebury dataset.

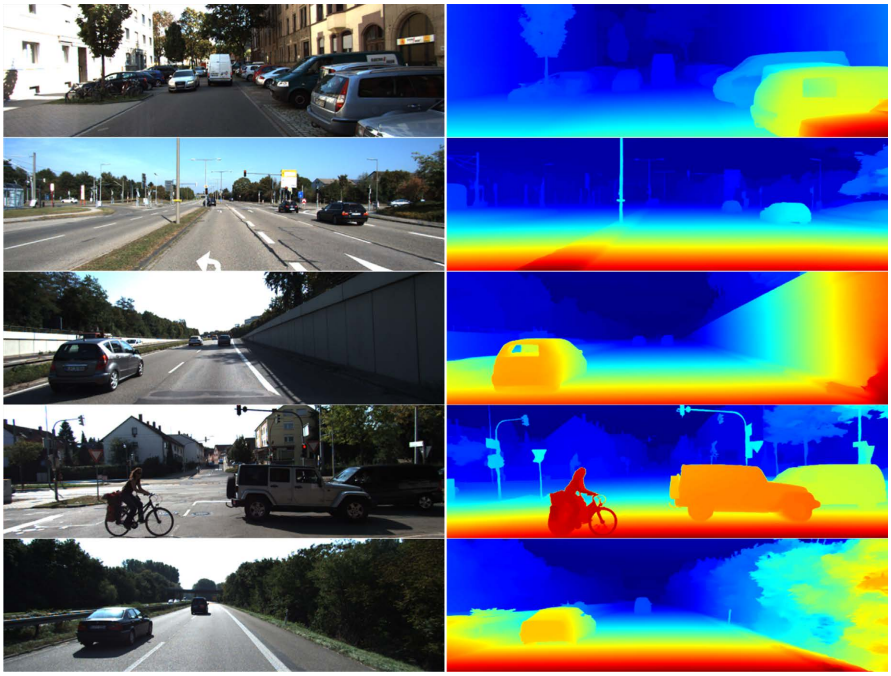

Figure 8. Visualization Effect on KITTI 2015 Dataset.

dataset. The ETH3D visualization results are shown in **Figure 9**, and the evaluation results are shown in **Table 3**. Among them, ATUNet performed best on the ETH3D dataset, with a pixel ratio greater than 0.5 between the estimated and true values reaching 6.23%, a pixel ratio greater than 1.0 reaching 2.32%, and an average absolute error of 0.16. Compared with the popular GWCNet [25] model currently on the market, the pixel ratio greater than 0.5 between the estimated and true values was increased by 47.5%, the pixel ratio greater than 1.0 was increased by 36.6%, and the average absolute error was increased by 44.8%. At the same time, **Table 3** also shows that the proposed ATUNet model with the addition of the whitening loss module has better generalization performance than the

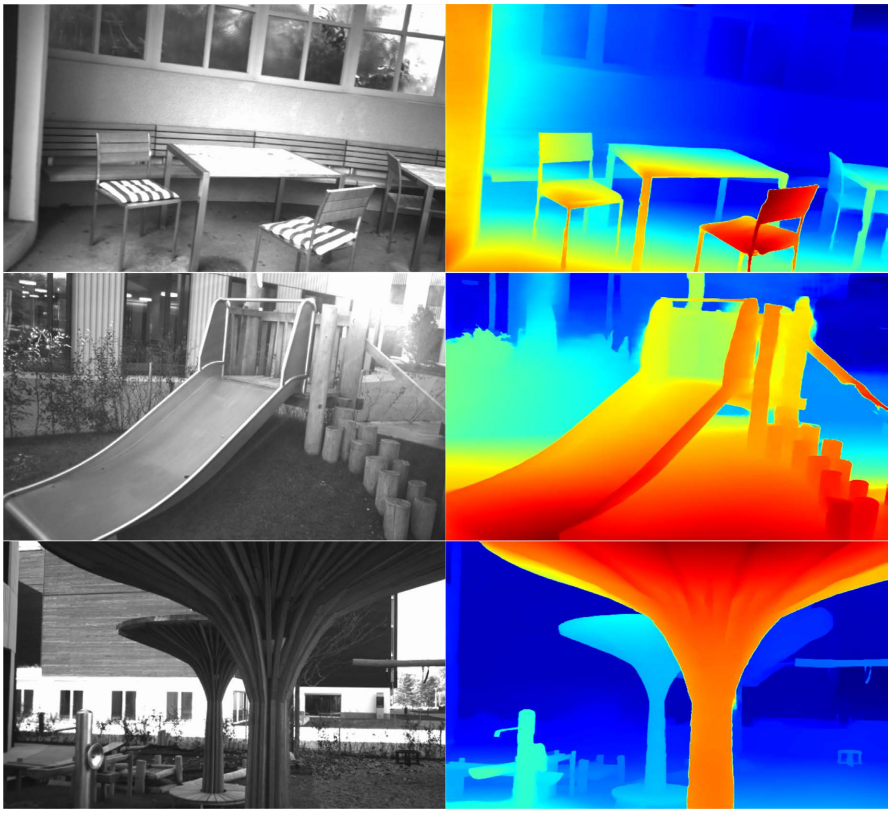

Figure 9. ETH3D training dataset effect diagram.

Table 3. ETH3D evaluation results.

Models	GANet	HSMNet	GWCNet	TUNet	ATUNet(ours)
Training datasets	Synthetic datasets	Synthetic datasets	Scene Flow	Scene Flow	Scene Flow
Bad 0.5	26.54	12.13	12.04	7.33	6.23
Bad 1.0	7.32	4.00	3.66	2.51	2.32
AvgErr	0.43	0.29	0.29	0.18	0.16

TUNet model. The pixel ratio greater than 0.5 between the estimated and true values was increased by 15.0%, the pixel ratio greater than 1.0 was increased by 7.6%, and the average absolute error was increased by 11.1%.

5. Conclusion

This paper proposes a stereo matching model with wider adaptability, which incorporates a whitening loss module during feature extraction to improve the model's generalization ability by constraining the variation of sensitive pixels in the feature domain. Experimental results show that the improved network model has good cross-dataset adaptability and can better transfer the training results to other datasets through transfer learning. The proposed method is compared with several existing stereo matching algorithms on multiple datasets and effectively

reduces the error matching rate while exhibiting a certain level of robustness.

Acknowledgements

This work was supported by Natural Science Foundation of China Youth Fund (No. 62001272), Shandong Provincial Natural Science Fund (No. ZR2019BF022).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Geiger, A., Roser, M. and Urtasun, R. (2010) Efficient Large-Scale Stereo Matching. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-19315-6_3

[2] Scharstein, D. and Szeliski, R. (2002) A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, **47**, 7-42. https://doi.org/10.1023/A:1014573219977

[3] LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep Learning. *Nature*, **521**, 436-444. https://doi.org/10.1038/nature14539

[4] Bengio, Y., Courville, A. and Vincent, P. (2013) Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**, 1798-828. https://doi.org/10.1109/TPAMI.2013.50

[5] Xu, J., Wang, H., Niu, Z.-Y., *et al.* (2020) Conversational Graph Grounded Policy Learning for Open-Domain Conversation Generation. *Proceedings of the* 58*th Annual Meeting of the Association for Computational Linguistics*, Online, 5-10 July 2020. https://doi.org/10.18653/v1/2020.acl-main.166

[6] Zhang, F., Qi, X., Yang, R., *et al.* (2020) Domain-Invariant Stereo Matching Networks. In: Vedaldi, A., Bischof, H., Brox, T. and Frahm, J.-M., Eds., *Computer Vision—ECCV* 2020. *Lecture Notes in Computer Science*, Vol. 12347, Springer, Cham. https://doi.org/10.1007/978-3-030-58536-5_25

[7] Shen, Z., Dai, Y. and Rao, Z. (2021) CFNet: Cascade and Fused Cost Volume for Robust Stereo Matching. *Proceedings of the* 2021 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Nashville, 20-25 June 2021. https://doi.org/10.1109/CVPR46437.2021.01369

[8] Cai, C., Poggi, M., Mattoccia, S. and Mordohai, P. (2020) Matching-Space Stereo Networks for Cross-Domain Generalization. *Proceedings of the* 2020 *International Conference on* 3*D Vision* (3*DV*), Fukuoka, 25-28 November 2020. https://doi.org/10.1109/3DV50981.2020.00046

[9] Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998) Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, **86**, 2278-2324. https://doi.org/10.1109/5.726791

[10] Li, J., Wang, P., Xiong, P., Cai, T., Yan, Z., *et al.* (2022). Practical Stereo Matching via Cascaded Recurrent Network with Adaptive Correlation. *Proceedings of the* 2022 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, 18-24 June 2022, 16263-16272. https://doi.org/10.1109/CVPR52688.2022.01578

[11] Mayer, N., Ilg, E., Hausser, P., *et al.* (2016) A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. *Proceedings of the* 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*),

Las Vegas, 27-30 June 2016. https://doi.org/10.1109/CVPR.2016.438

[12] Du, X., El-Khamy, M. and Lee, J. (2019) Amnet: Deep Atrous Multiscale Stereo Disparity Estimation Networks. ArXiv Preprint ArXiv: 190409099.

[13] Chang, J.-R. and Chen, Y.-S. (2018) Pyramid Stereo Matching Network. *Proceedings of the* 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018. https://doi.org/10.1109/CVPR.2018.00567

[14] Lowe, D.G. (1999) Object Recognition from Local Scale-Invariant Features. *Proceedings of the 7th IEEE International Conference on Computer Vision*, Kerkyra, 20-27 September 1999. https://doi.org/10.1109/ICCV.1999.790410

[15] Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L. (2008) Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, **110**, 346-359. https://doi.org/10.1016/j.cviu.2007.09.014

[16] Choi, S., Jung, S., Yun, H., *et al.* (2021) RobustNet: Improving Domain Generalization in Urban-Scene Segmentation via Instance Selective Whitening. *Proceedings of the* 2021 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Nashville, 20-25 June 2021. https://doi.org/10.1109/CVPR46437.2021.01141

[17] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. *Proceedings of the* 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90

[18] Vaswani, A., Shazeer, N., Parmar, N., *et al.* (2017) Attention Is All You Need. *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*, Long Beach, 4-9 December 2017.

[19] Schlag, I., Irie, K. and Schmidhuber, J. (2021) Linear Transformers Are Secretly Fast Weight Programmers. *Proceedings of the* 38*th International Conference on Machine Learning*, Online, 18-24 July 2021.

[20] Cho, K., Van Merriënboer, B., Gulcehre, C., *et al.* (2014) Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Doha, 25-29 October 2014. https://doi.org/10.3115/v1/D14-1179

[21] Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) Learning Representations by Back-Propagating Errors. *Nature*, **323**, 533-536. https://doi.org/10.1038/323533a0

[22] Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the* 32*nd International Conference on International Conference on Machine Learning*, Lille, 6-11 July 2015, 448-456.

[23] Ulyanov, D., Vedaldi, A. and Lempitsky, V. (2016) Instance Normalization: The Missing Ingredient for Fast Stylization. ArXiv Preprint ArXiv: 1607.08022.

[24] Bahmani, B., Moseley, B., Vattani, A., Kumar, R. and Vassilvitskii, S. (2012) Scalable k-Means++. *Proceedings of the VLDB Endowment*, **5**, 622-633. https://doi.org/10.14778/2180912.2180915

[25] Guo, X., Yang, K., Yang, W., Wang, X. and Li, H. (2019) Group-Wise Correlation Stereo Network. *Proceedings of the* 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019. https://doi.org/10.1109/CVPR.2019.00339