A NOVEL STEREO MATCHING PIPELINE WITH ROBUSTNESS AND UNFIXED DISPARITY SEARCH RANGE

Jiazhi Liu[*†], *Feng Liu*[*†]

[*]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences; [†]School of Cyber Security, University of Chinese Academy of Sciences
Email: {liujiazhi, liufeng}@iie.ac.cn

ABSTRACT

Stereo matching is an essential basis for various applications, but most stereo matching methods have poor generalization performance and require a fixed disparity search range. Moreover, current stereo matching methods focus on the scenes that only have positive disparities, but ignore the scenes that contain both positive and negative disparities, such as 3D movies. In this paper, we present a new stereo matching pipeline that first computes semi-dense disparity maps based on binocular disparity, and then completes the rest depending on monocular cues. The new stereo matching pipeline have the following advantages: It 1) has better generalization performance than most of the current stereo matching methods; 2) relaxes the limitation of a fixed disparity search range; 3) can handle the scenes that involve both positive and negative disparities, which has more potential applications, such as view synthesis in 3D multimedia and VR/AR. Experimental results demonstrate the effectiveness of our new stereo matching pipeline.

Index Terms— Stereo matching, Cross-domain generalization, View synthesis

1. INTRODUCTION

Recently, "metaverse" has attracted many people's attention for its unlimited possibilities in the future. Compared with current 2D display techniques, "metaverse" additionally utilizes binocular disparity [1] that refers to the slight difference between the left and right retina images, making people enjoy more realistic and immersive experience. Therefore, disparity estimation plays a basic role in the techniques of "metaverse". Since we have plenty of available stereoscopic content (such as 3D movies), there is a clear need for methods that can estimate disparity from such stereo pairs.

Given a position $\mathbf{p}$ in the 3D world space, if its projections into the rectified left and right views are $\mathbf{p}^{\mathrm{L}}$ and $\mathbf{p}^{\mathrm{R}}$, then their

disparity is defined as $\boldsymbol{D}^{\mathrm{L}}(\mathbf{p}^{\mathrm{L}}) = \boldsymbol{D}^{\mathrm{R}}(\mathbf{p}^{\mathrm{R}}) = \mathbf{p}_x^{\mathrm{L}} - \mathbf{p}_x^{\mathrm{R}}$ where $\mathbf{p}_x^{\mathrm{L}}$ and $\mathbf{p}_x^{\mathrm{R}}$ denote the horizontal components of $\mathbf{p}^{\mathrm{L}}$ and $\mathbf{p}^{\mathrm{R}}$. Current state-of-the-art stereo matching methods are designed and trained to match only over positive disparity ranges because the prior methods 1) mainly focused on the applications of autonomous driving and robot navigation that only require positive disparities; 2) need plenty of data to supervise the network, but there is not enough labeled data for the scenes containing both positive and negative disparities; 3) usually have poor generalization performance, so these methods have difficulties in utilizing large amounts of labeled data from known datasets. In this paper, we explore the above challenges and present a new stereo matching pipeline: first computing semi-dense disparity maps based on binocular disparity, and then estimating the rest based on monocular cues.

In contrast, the traditional stereo matching pipeline [2] usually contains four steps: matching cost computation, cost aggregation, disparity computation, and refinement. Recent learning-based stereo matching methods also follow the traditional pipeline and design various network architectures. Here we review matching cost computation and cost aggregation that prior methods mainly involve.

Matching costs are calculated based on hand-crafted or learning-based feature maps extracted from the left and right images, which expresses how well a pixel in the reference image matches its candidates in the other. Feature descriptors in traditional stereo matching methods are hand-crafted, usually based on gradients [3,4] and mutual information [5]. The early deep-learning-based methods focus on designing feature extraction networks to characterize the left and right images [6,7], and then utilize non-learned cost aggregation and refinement to determine the final disparity map. The recent end-to-end methods also design various Siamese-like networks to extract features, such as GCNet [8], StereoNet [9], PSMNet [10], and GANet [11], and their feature extractors are integrated into the whole network and trained in an end-to-end manner. After obtaining matching costs, we intuitively want to take the candidate with the smallest matching cost (or the largest similarity) as the best-matched pixel, but researchers generally do not adopt the intuitive operation for

the following reasons: 1) the left and right features have poor ability to characterize the texture-less or repetitive regions; 2) the single-visible pixels do not have corresponding pixels in the other image. Therefore, for these ticklish pixels, the prior methods usually estimate their disparities from their neighboring tractable pixels, which is the job of cost aggregation.

Cost aggregation denotes we aggregate space context information in the cost volume, so that we can compute more reliable disparities. Most traditional and some learning-based methods aggregate matching costs in a 3D cost volume (*disparity×height×width*). Cost aggregation in a 3D cost colume, including SGM [5], CostFilter [4], DispNet [12], and AANet [13], resembles the votes of neighboring pixels. Recent learning-based methods, such as PSMNet [10], GANet [11], and CFNet [14] execute cost aggregation in a 4D cost volume (*feature×disparity×height×width*), because researchers believe that 4D cost volumes contain more information than the 3D form.

Motivated by the neuroscience fact that binocular disparity is a low-level, pre-attentive cue of human depth cues, we propose a new stereo matching pipeline that highlights the role of feature extractors. Different from the traditional stereo matching pipeline [2], our proposed pipeline no longer needs cost aggregation: It first computes accurate semi-dense disparity maps directly from feature maps, and then generates the final disparity maps by disparity completion networks.

In summary, our proposed stereo matching pipeline has the following advantages: 1) It has better generalization performance than most state-of-the-art stereo matching methods; 2) It relaxes the limitation of a fixed disparity search range; 3) It can handle the scenes that involves both positive and negative disparities, whereas prior learning-based methods can only deal with the scenes with positive disparities; 4 It does not store and process cost volumes at the inference time, so it needs less GPU memory.

2. METHOD

Our goal is to compute the disparity map from a known rectified stereo pair. As mentioned above and shown in Figure 1, here we introduce the two steps of our stereo matching pipeline in detail.

2.1. Computing Semi-dense Disparity Maps directly from Feature Maps

Like prior methods, we adopt a shared feature extractor to extract the feature maps from the left and right images. Each feature map is a tensor with the size of $\alpha h \times \alpha w \times n$ where h, w, n, α represent image height, width, feature channels and the down-scaled scale. Here we introduce the symbol α to emphasize that current feature extractors generally extract feature maps at smaller resolution than the original image. Like AANet [13], we utilize a full correlation to generate a

3D cost volume $\boldsymbol{C} = (c_{i,j}^{(d)})$ for pre-defined disparity candidates $\mathcal{D} = \{d \in \mathbb{Z} \mid \alpha d_{\min} \leqslant d \leqslant \alpha d_{\max}\}$. The 3D cost volume $\boldsymbol{C}$ is a tensor with the size of $\alpha h \times \alpha w \times m$ where $m = \alpha(d_{\max} - d_{\min}) + 1$, and each element $c_{i,j}^{(d)}$ records how well the position $(i/\alpha, j/\alpha)$ in the left image matches its disparity candidate $(i/\alpha, j/\alpha - d/\alpha)$ in the right image.

After building a 3D cost volume, prior methods usually execute cost aggregation on the cost volume, so they need to store the volume at the whole inference process. In contrast, we generate an initial disparity map $D^{\mathrm{init}} = (d_{i,j}^{\mathrm{init}}) \in [\alpha d_{\min}, \alpha d_{\max}]^{\alpha h \times \alpha w}$ and the corresponding reliability map $R = (r_{i,j}) \in [0,1]^{\alpha h \times \alpha w}$ directly from the cost volume, and thus there is no need to store the volume at the subsequent steps. The reliability map records how reliable a predicted disparity is, so we can obtain semi-dense disparity maps from initial disparity maps and reliability maps.

We take the computation of disparity at the position (i, j) as an example. We first convert the 3D cost volume $\boldsymbol{C}$ into the 3D probability volume $\boldsymbol{P} = (p_{i,j}^{(d)})$ by executing the softmax operation along the disparity dimension. Like the sub-pixel maximum a posteriori (MAP) [15,16], we compute the initial disparity map according to

$$d_{i,j}^{\mathrm{init}} = \hat{d}_{i,j} + o_{i,j} \tag{1}$$

where $\hat{d}_{i,j} = \mathrm{argmax}_{d \in \mathcal{D}} \, p_{i,j}^{(d)}$ denotes we chose the disparity candidate with the largest probability and $o_{i,j}$ is a refinement term to achieve sub-pixel precision. The refinement term $o_{i,j}$ is computed according to the possibilities at $\hat{d}_{i,j} - 1$ and $\hat{d}_{i,j} + 1$:

$$o_{i,j} = \sum_{\delta=-1}^{1} \delta \cdot p_{i,j}^{(\hat{d}_{i,j}+\delta)} \left/ \sum_{\delta=-1}^{1} p_{i,j}^{(\hat{d}_{i,j}+\delta)} \right. \tag{2}$$

where we regard $\sum_{\delta=-1}^{1} p_{i,j}^{(\hat{d}_{i,j}+\delta)}$ as the reliability $r_{i,j}$ of the initial disparity $d_{i,j}^{\mathrm{init}}$.

Finally, we filter the initial disparity map D^{init} to obtain a semi-dense disparity map by left-right consistency checking [4] and reliability checking (see Figure 1). The latter checking refers to that the reliability of a predicted disparity has to be larger than the reliability threshold τ.

2.1.1. Loss Function

Unlike prior stereo matching methods that train the feature extractor and rest modules in an end-to-end way, our method first trains the feature extractor since we have been highlighting its importance. The loss function contains three components: $\ell^{\mathrm{dvr}} + \ell^{\mathrm{ur}} + \ell^{\mathrm{trr}}$.

Double-visible-region loss ℓ^{dvr}. For the position (i, j) in the probability volume $\boldsymbol{P}$, we hope its maximal probability along the disparity dimension occurs at the ground truth disparity. Moreover, if the position (i, j) is located in the occluded region, we cannot compute its disparity directly from feature maps since it has no corresponding pixel in the other.
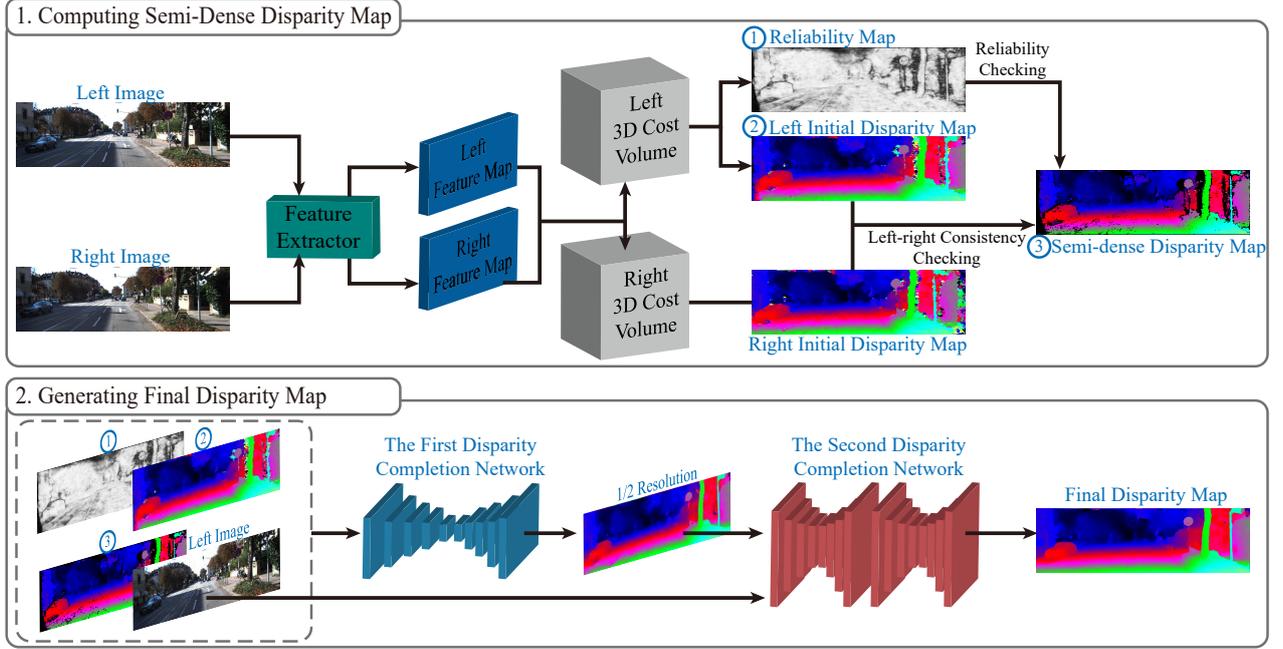
Fig. 1. Overview of our proposed stereo matching pipeline at the inference time. The pipeline contains two stages: first computing semi-dense disparity maps directly from feature maps; then generating final disparity maps by two disparity completion networks. At the training time, we first train the first stage using the loss functions in Sec. 2.1.1, and then the second stage.
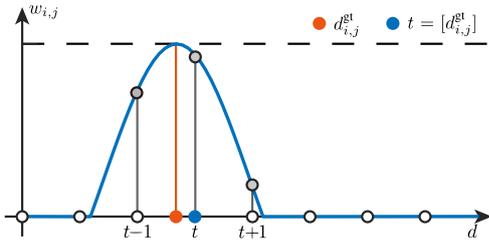

Fig. 2. An example of weights of the double-visible-region loss.

Therefore, we only execute this loss function in the double-visible region that can be derived from the ground-truth disparity map $D^{\text{gt}} = (d_{i,j}^{\text{gt}})$. Formally, the double-visible-region loss is defined as

$$\ell^{\text{dvr}} = -\frac{1}{|\mathcal{V}|} \sum_{(i,j) \in \mathcal{V}} \left(\sum_{\delta=-1,0,1} w_{i,j}^{(t+\delta)} \cdot \log\left(p_{i,j}^{(t+\delta)} \right) \right) \quad (3)$$

where $t = [d_{i,j}^{\text{gt}}]$ indicates the integral disparity closest to the ground truth and the set $\mathcal{V}$ denotes all the pixels in the double-visible region. The coefficient $w_{i,j}^{(t+\delta)}$ is a weight function, and the closer to the ground truth a disparity is, the larger its weight will be (see Figure 2). We give its detailed definition in the supplementary material.

Unreliable-region loss ℓ^{ur}. We hope that a predicted disparity has as low reliability as possible if it is located in the occluded region or its value diverges from the ground truth ($\geqslant 1$). Such the region is called the unreliable region $\mathcal{U}$. The unreliable-region loss is defined as follows:

$$\ell^{\text{ur}} = -\frac{1}{|\mathcal{U}|} \sum_{(i,j) \in \mathcal{U}} \log(1 - r_{i,j}). \quad (4)$$

To-refine-region loss ℓ^{trr}. If a predicted disparity $d_{i,j}^{\text{init}}$ is close to the ground truth, then we use this loss function to make the predicted disparity achieve more accurate sub-pixel precision. Formally, the loss function is

$$\ell^{\text{trr}} = \frac{1}{|\mathcal{T}|} \sum_{(i,j) \in \mathcal{T}} |d_{i,j}^{\text{init}} - d_{i,j}^{\text{gt}}| \quad (5)$$

where $\mathcal{T} = \{(i,j) \in \mathcal{V} \big| |d_{i,j}^{\text{init}} - d_{i,j}^{\text{gt}}| \leqslant 1\}$.

2.2. Generating Final Disparity Maps by Disparity Completion Networks

In the last subsection, we obtain a semi-dense disparity map, an initial disparity map, and a reliability map. As shown in Figure 1, we utilize two disparity completion networks to complete hierarchically the semi-dense disparity and then acquire the final disparity map. The first disparity completion network is similar to the monocular depth estimation network [17]. It takes the initial disparity map, the semi-dense

disparity map, the reliability map, and the reference color image as inputs, and outputs a full disparity map with the $1/2$ resolution. The second disparity completion network is similar to the refinement module in AANet [13]. It takes the full disparity map from the first network and the left image as inputs and outputs the final disparity map.

At the training time, the first disparity completion network also outputs the disparity maps with the resolution $1/4, 1/8, 1/16$. We train the two networks by supervising the five disparity maps with different resolutions. The corresponding loss function is the smooth L1 loss [10] that measures the difference between output disparity maps and the ground truths. More details are provided in the supplementary material.

3. EXPERIMENTS

3.1. Datasets

We train the networks only in SceneFlow [12], and evaluate them qualitatively and quantitatively on the training sets of KITTI2012 [18], KITTI2015 [19], Middlebury [20], which only contain positive disparities. We also use the dataset 3D Movie that includes negative disparities besides positive ones.
SceneFlow. SceneFlow is a large synthetic dataset containing 35,454 training and 4,370 test images with a resolution of 960×540, which have dense ground truth disparity maps.
KITTI2012&2015. KITTI2012 and KITTI2015 are real-world datasets with street views captured from a driving car, providing 394 stereo pairs of outdoor driving scenes with sparse ground-truth disparities for training, and 395 pairs for testing.
Middlebury. Middlebury is a small indoor dataset containing less than 50 stereo pairs with three different resolutions. Here we use its half version.
3D Movie. We chose stereo images that contain both negative and positive disparities from the 3D movie "Big Buck Bunny" ©by Blender Foundation (https://peach.blender.org/). Their resolution is 1080×1920. The dataset does not have available ground truth disparities, so we just give a qualitative evaluation and visualized results.

3.2. Implementation Details

We implemented the feature extractors in PyTorch and using Adam ($\beta_1 = 0.9, \beta_2 = 0.999$) as optimizer. At training time we set the minimum disparity $d_{\min} = 0$ and the maximum disparity $d_{\max} = 320$, whereas at test time we adaptively adjusted the minimum and maximum disparity for different datasets. Specifically, we set $d_{\min} = 0, d_{\max} = 192$ for KITTI2012 and KITTI2015, $d_{\min} = 0, d_{\max} = 320$ for Middlebury, and set $d_{\min} = -100, d_{\max} = 100$ for 3D Movie. We performed color normalization with ImageNet mean and standard deviation. During training, images were randomly
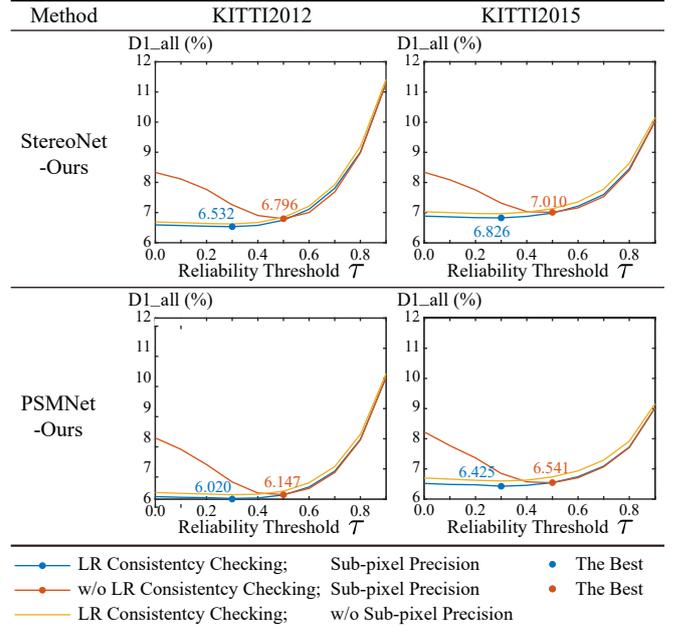

Fig. 3. Ablation study results for left-right consistency checking, sub-pixel precision, and different reliability threshold values for reliability checking. "StereoNet-Ours" denotes we utilized the feature extractor of StereoNet [9] in our proposed stereo matching pipeline. "PSMNet-Ours" denotes we utilized the feature extractor of PSMNet [10] in our proposed stereo matching pipeline. "w/o sub-pixel precision" means we just use $\hat{d}_{i,j}$ in Eq.(2) as the initial disparity map.

cropped to the size 288×576, and their color was randomly augmented. We also used asymmetric data augmentation [21]. We first trained the feature extractor for 18 epochs with a batch size 4, and the learning rate started at 0.0005 and was decreased by half every 3 epochs after 6 epochs. After training the feature extractor, we then trained the rest module networks for 24 epochs, and the learning rate started at 0.001 and was decreased by half every 4 epochs after 8 epochs. All the models are trained and tested on one NVIDIA Tesla-P100 GPU.

3.3. Ablation Study

We conducted experiments with several settings to evaluate our proposed stereo matching pipeline, including the usage of left-right consistency checking, sub-pixel precision, different reliability threshold values for reliability checking, and different feature extractors (StereoNet [9] and PSMNet [10]). Left-right consistency checking and reliability checking work together to filter possibly-wrong disparities in the initial disparity map. Higher reliability threshold τ means more strict reliability checking. As shown in Figure 3, if τ is lower than about 0.4, left-right consistency checking has a signifi-

Method	KITTI2012 D1_all(%)↓	KITTI2015 D1_all(%)↓	Middlebury bad2.0(%)↓	Time (s)↓
Traditional Methods				
CostFilter [4]	21.7	18.9	40.5	240*
SGM [5]	7.1	7.6	25.2	3.7*
Domain-specific Methods				
PSMNet [10]	15.1	16.3	25.1	0.337
GWCNet [22]	20.2	22.7	34.2	0.32*
AANet [13]	18.3	12.2	31.0	0.123
Domain-invariant Methods				
DSMNet [23]	6.2	6.5	**13.8**	1.6*
STTR [21]	8.4	7.7	oom	1.416
Our new stereo matching pipeline				
StereoNet-Ours	6.5	6.8	22.8	**0.095**
PSMNet-Ours	**6.0**	**6.4**	21.2	0.198

Table 1. Generalization performance evaluation. "oom" denotes "out of memory". We tested the running time on a NVIDIA P100 GPU for the KITTI resolution 384×1344. The symbol "*" in Time denotes that the running time value is taken from the stereo evaluation system of KITTI.

cant effect on the final result; otherwise, left-right consistency checking no longer work. Removing the sub-pixel precision module leads to a slight performance drop.

For the methods "StereoNet-Ours" and "PSMNet-Ours", we get the best result when using left-right consistency checking, reliability checking with $\tau = 0.3$, and sub-pixel precision module. If we do not adopt left-right consistency checking, we can also obtain competitive results just depending on sub-pixel precision and reliability checking with $\tau = 0.5$ (the blue dot v.s. the orange dot in Figure 3).

3.4. Evaluation

3.4.1. Cross-domain generalization performance

Current state-of-the-art methods can achieve impressive performance in some special datasets, but usually have poor generalization performance. Here we evaluate the generalization performance of our stereo matching pipeline. Specifically, all the methods are trained in the synthesized dataset Scene-Flow [12], and tested in three real-world datasets KITTI2012 [18], KITTI2015 [19], and Middlebury [20]. As shown in Table 1, our method outperforms traditional methods [4, 5], and the learning-based domain-specific methods [10, 13, 22] on three datasets. STTR [21] adopts the prevailing *transformer* architecture and is designed for cross-domain generalization. Our method outperforms STTR on two datasets. Note that STTR failed in the half-resolution version of Middlebury, because it requires too much GPU memory. DSM-Net [23] is also designed for cross-domain generalization, and our method can surpass it in two datasets. Besides, our method is far faster than the two domain-invariant methods STTR [21] and DSMNet [23].

Table 2 lists the qualitative evaluation results of our method and prior methods. For traditional methods, they can-

Method	Unfixed Range	Sub-pixel Precision	Negative Disparities	Robust
CostFilter [4]	✓	✗	✓	✗
SGM [5]	✓	✗	✓	✓
PSMNet [10]	✓✗	✓	✗	✗
AANet [13]	✗	✓	✗	✗
DSMNet [23]	✓✗	✓	✗	✓
STTR [21]	✓	✓	✗	✓
Ours	✓	✓	✓	✓

Table 2. Qualitative evaluation of our method and prior methods. The symbol ✓✗ denotes in the column "Unfixed Range" denotes that the performance will be unstable if we change their disparity search range [15].
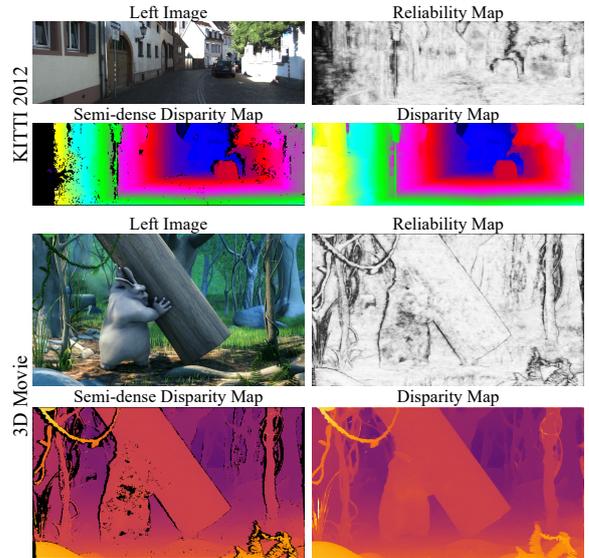

Fig. 4. Visualized results. In reliability maps, bright colors means high reliability, whereas dark colors denotes low reliability.

not generate sub-pixel disparity maps because they use the scheme "winner takes all". For learning-based methods, 1) if a method (such as AANet [13]) adopt cost aggregation on a 3D cost volume, its disparity search range is fixed and cannot be changed; 2) if a method (such as PSMNet [10] and DSM-Net [23]) executes cost aggregation on a 4D cost volume, its disparity range can be changed, but the performance is unstable if the range is changed. These learning-based methods are limited to supporting the scenes with only positive disparities, whereas our stereo matching pipeline can also support the scenes with both negative and positive disparities.

3.4.2. Visualized Results

We give an illustration of our method on KITTI2012 and 3D Movie. As Figure 4 shows, although our method is only

trained in a synthesized dataset and over positive disparities, it still has a satisfying visualized result. Note that the stereo images of 3D Movie contain both positive and negative disparity, but it does not matter for our method. Figure 4 also shows reliability maps and semi-dense disparity maps. Surprisingly, the disparities of more than 80% pixels can be computed accurately just depending on feature extractors, and most unknown disparities occur in the occluded region. For more visualized results, please see the supplementary material.

4. CONCLUSION

In this paper, we have presented a new stereo matching pipeline that highlights binocular disparity of human depth cues, and no longer needs cost aggregation: It first computes accurate semi-dense disparity maps directly from feature maps, and then generates the final disparity maps by disparity completion networks. The new stereo matching pipeline 1) has superior generalization performance, 2) avoids the need to pre-specify a fixed disparity search range, and 3) can handle the scenes with both positive and negative disparities. We experimentally demonstrate that our stereo matching pipeline generalizes to different domains without fine-tuning and give visualized results on the scenes with both negative and positive disparities.

5. REFERENCES

[1] Piotr Didyk, Tobias Ritschel, Elmar Eisemann, Karol Myszkowski, and Hans-Peter Seidel, "A perceptual model for disparity," in *SIGGRAPH*, 2011.

[2] Daniel Scharstein and Richard Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1, 2002.

[3] Andreas Klaus, Mario Sormann, and Konrad Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *ICPR*, 2006, vol. 3.

[4] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE TPAMI*, vol. 35, no. 2, 2012.

[5] Heiko Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE TPAMI*, vol. 30, no. 2, 2007.

[6] Jure Zbontar, Yann LeCun, et al., "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, no. 1, 2016.

[7] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun, "Efficient deep learning for stereo matching," in *CVPR*, 2016.

[8] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry, "End-to-end learning of geometry and context for deep stereo regression," in *ICCV*, 2017.

[9] Sameh Khamis, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi, "Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction," in *ECCV*, 2018.

[10] Jia-Ren Chang and Yong-Sheng Chen, "Pyramid stereo matching network," in *CVPR*, 2018.

[11] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr, "Ga-net: Guided aggregation net for end-to-end stereo matching," in *CVPR*, 2019.

[12] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *CVPR*, 2016.

[13] Haofei Xu and Juyong Zhang, "Aanet: Adaptive aggregation network for efficient stereo matching," in *CVPR*, 2020.

[14] Zhelun Shen, Yuchao Dai, and Zhibo Rao, "CFNet: Cascade and fused cost volume for robust stereo matching," in *CVPR*, 2021.

[15] Stepan Tulyakov, Anton Ivanov, and Francois Fleuret, "Practical deep stereo (pds): Toward applications-friendly deep stereo matching," *Advances in neural information processing systems*, vol. 31, 2018.

[16] Tyler Nuanes, Matt Elsey, Aswin Sankaranarayanan, and John Shen, "Soft cross entropy loss and bottleneck tri-cost volume for efficient stereo depth prediction," in *CVPR*, 2021.

[17] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.

[19] Moritz Menze and Andreas Geiger, "Object scene flow for autonomous vehicles," in *CVPR*, 2015.

[20] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *German conference on pattern recognition*, 2014.

[21] Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X Creighton, Russell H Taylor, and Mathias Unberath, "Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers," in *ICCV*, 2021.

[22] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li, "Group-wise correlation stereo network," in *CVPR*, 2019.

[23] Feihu Zhang, Xiaojuan Qi, Ruigang Yang, Victor Prisacariu, Benjamin Wah, and Philip Torr, "Domain-invariant stereo matching networks," in *ECCV*, 2020.