UDFNET: UNSUPERVISED DISPARITY FUSION WITH ADVERSARIAL NETWORKS

Can Pu, Robert B. Fisher

School of Informatics, University of Edinburgh, UK

ABSTRACT

Fusing disparity maps from different methods is an useful technique to get a refined disparity map by leveraging the complimentary advantage. We present a model for disparity fusion that uses an adversarial network, which can be trained without using ground truth disparity data. We input two initial disparity maps (from the left view) along with auxiliary information (gradient, left & right intensity image) into the generator and train the generator to output a refined disparity map registered on the left view. The refined left disparity map and left intensity image are used to reconstruct a fake right intensity image. Finally, the fake and real right intensity images (from the right stereo vision camera) are fed into a discriminator. The trained network's architecture is effective for the fusion task (90 fps on Kitti2015). The accuracy is on par or even better than the state-of-art supervised methods. A demo video is available https://youtu.be/XTHOF3kZGsU.

***Index Terms*—** Disparity Fusion, Adversarial network, Unsupervised, Stereo-stereo fusion, Stereo-lidar fusion.

1. INTRODUCTION

With the popularity of 3D vision, how to get more accurate disparity (equivalent to depth)[1] information is important. Currently, there are many methods to obtain depth information, such as active illumination devices (eg: structured light cameras, Time of Flight (ToF) sensors), passive methods (monocular vision [1], stereo vision [2, 3, 4, 5]) etc. However, none of these methods are perfect in all scenes. Thus, disparity fusion from multiple sources is urgently needed, where different data sources can compensate for the weaknesses of each other.

Recently, different kinds of disparity fusion methods have emerged in different sub-tasks, such as stereo-ToF fusion ([6, 7, 8]), stereo-stereo fusion ([9]), Lidar-stereo fusion ([10, 11]) and general depth fusion ([12]). For this task, deep-learning based methods perform much better. However, all of the previous algorithms are supervised. As far as we know, we are the first to develop an unsupervised depth fusion method.

Unsupervised disparity fusion is hard because it requires computing an accurate disparity map without any ground truth disparity data. Existing unsupervised strategies based on left and right intensity consistency cannot guarantee a highly accurate disparity map. For example, Monodepth [1] treated the left-right intensity consistency error as a global metric in their cost function and slight intensity changes in the images influence the global estimation greatly. Here, left-right intensity consistency is just one of our local refinement metrics, which increases both global robustness and accuracy. Previous work, such as Sdf-GAN [12], achieves top disparity fusion performance but it needs ground truth disparity data to train. By combining the global disparity initialization with local disparity refinement, we can achieve unsupervised fusion. Thus, the proposed work is different from previous work.

In this paper, a fully unsupervised disparity fusion framework (Figure 1) is proposed based on Generative Adversarial Network (GAN [13]). The generator is trained to output a refined disparity value close to the weighted sum of the disparity inputs from global initialization (Equation 1). Then, three refinement principles are adopted to refine the depth. (1) The reconstructed intensity error between the reconstructed and real right intensity image is minimized (Equation 2). (2) The similarities between the reconstructed and real right image in different receptive fields are maximized (Equation 3). (3) The refined disparity map is smoothed based on the corresponding intensity image space (Equation 4). An efficient network structure has been designed (See supplementary material).

Section 2 presents the methodology. Section 3 presents the experimental results. Section 4 presents the conclusion.

Contributions: We have:

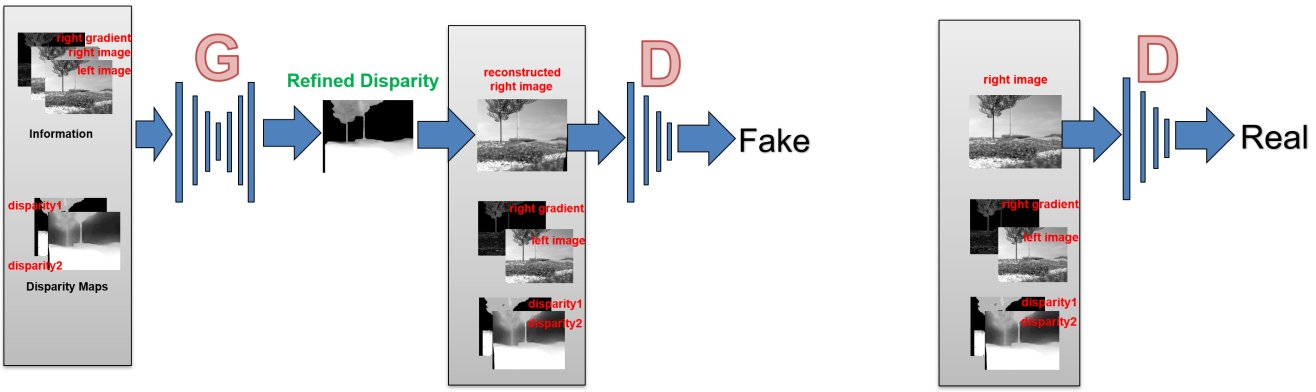
1. An efficient unsupervised disparity fusion strategy by combining global disparity initialization and local refinement
2. An indirect method using a GAN to force the disparity Markov Random Field in the refined disparity map to be close to that in the real disparity map
3. An unsupervised end-to-end uncertainty-based pipeline that can fuse registered disparity maps from different sources

2. METHOD

First the pipeline is proposed and then the cost functions for the networks are presented.

Thanks to TrimBot2020 [EC Grant Agreement No. 688007] for funding.

[1] depth = focal_length*baseline/disparity.



(a) Negative examples

(b) Positive examples

Fig. 1: Our target output is refined disparity map. The inputs to the generator (G) are the initial disparity maps in the left view and auxiliary information (left intensity image, right intensity image, right gradient). The gradient of the left view is calculated from the left image directly. The input to discriminator (D) is the combination of the auxiliary information, the initial disparity inputs and the reconstructed/real right image. The generator produces a refined disparity map in the left view. The refined left disparity map and left image are used to reconstruct the right image. The discriminator discriminates whether the input is fake (reconstructed right image) or true (right image). The images in each block come from a synthetic garden dataset.

2.1. Fusion Pipeline

The whole process based on GAN [13] is shown in Figure 1.

2.2. Objective Function

The goal is to get a refined disparity map from initial disparity maps and auxiliary image information. The main ideas are:

• The disparity fusion has initial disparity inputs (unlike stereo vision etc.). The initial disparity maps should be used to provide the global initial value of the refined disparity map first. That is, the refined disparity map from the generator is encouraged to be similar to the input disparities (Equation 1).

• The initialization based on Equation 1 provides a coarse disparity map. Refinement will be realized by three local decision strategies. We reconstruct the right intensity image from the left intensity image and disparity map. Thus, the accuracy of the refined disparity map can be assessed indirectly by comparing the reconstructed right image and real right image. We design the L_1 intensity error based on the gradient in Equation 2 and describe the distance between the Markov Random Field of the refined disparity map and real disparity distribution in Equation 3 indirectly. We also design a disparity smoothness term to reduce the outliers and noise in Equation 4 using the gradient.

More specifically, our cost functions are:

(1) A constraint that the output should be close to the weighted sum of the initial disparity inputs:

$$\mathcal{L}_c(G) = \mathop{\mathbb{E}}_{u\in\tilde{x},u_s\in\bar{x}_s,\tilde{x}\sim P_G,s=1..Z}[w_{u_s}||\tilde{x}_u - \bar{x}_{u_s}||_1] \quad (1)$$

where $\bar{x}_{u_s}$ is the disparity value of a pixel u_s in the s^{th} initial disparity map $\bar{x}_s$ (In Fig. 1, it is 'disparity1' or 'disparity2') corresponding to pixel u in the refined disparity map $\tilde{x}$ (In Fig. 1, it is 'Refined Disparity'). P_G represents the distribution of the samples $\tilde{x}$ from the generator and $\tilde{x}_u$ is the disparity value of pixel u. $||\bullet||_1$ is L_1 distance. w_{u_s} is the confidence of the pixel u_s. If no prior knowledge is available, $w_{u_s} = 1/Z$ for all pixels where Z is the number of initial disparity inputs.

(2) To encourage disparity estimates at edges to be more accurate, we incorporate gradient information as a weight into the L_1 distance to make the disparity edges shaper:

$$\mathcal{L}_{L_1}(G) = \mathop{\mathbb{E}}_{I_r\sim P_R,\tilde{I}_r\sim P'_G}[\exp(\alpha|\nabla(I_r)|)||I_r - \tilde{I}_r||_1] \quad (2)$$

where I_r is the real right intensity image from the right camera (In Fig. 1, it is 'right image') and $\tilde{I}_r$ is the reconstructed right intensity image from the generator (In Fig. 1, it is 'reconstructed right image'). $\nabla(I_r)$ is the gradient of the gray-scale image in the right view (In Fig. 1, it is 'right gradient' using the Sobel operator). $\alpha \geq 0$ weights the gradient value. P'_G represents the distribution of the samples $\tilde{I}_r$ reconstructed from the left intensity image and corresponding refined disparity map. P_R represents the distribution of the samples I_r from the right camera in the stereo vision setting. The goal is to encourage disparity estimates at intensity edges (larger gradients) to be more accurate with less reconstructed intensity error.

(3) Unlike [12], we input the reconstructed right image and real right image into the discriminator, which gives indirect feedback about whether the refined disparity distribution is close to the ground truth. By making the discriminator output the probabilities at different receptive fields or scales [please refer to D_i in the discriminator network architecture

in the supplementary material. $i = 1..M$ and $M = 5$ is the number of the scales], the generator will be forced to make the disparity distribution in the refined disparity map be close to the real distribution. To alleviate training difficulties, we adopt the Improved WGAN loss function [14].

$$\mathcal{L}_{wgan}(G, D_i) = \mathop{\mathbb{E}}_{\tilde{I}_r \sim P'_G}[D_i(\tilde{I}_r)] - \mathop{\mathbb{E}}_{I_r \sim P_R}[D_i(I_r)] + \lambda \mathop{\mathbb{E}}_{\hat{I}_r \sim P_{\hat{I}_r}}[(||\nabla_{\hat{I}_r} D_i(\hat{I}_r)||_2 - 1)^2] \quad (3)$$

where D_i is the probability at the i^{th} scale that the input image patch to the discriminator is from the real distribution. λ is the penalty coefficient ($\lambda = 0.0001$ is set). $\hat{I}_r$ is the random sample and $P_{\hat{I}_r}$ is its corresponding distribution. (For details, see [14]).

(4) To suppress outliers and noise in the refined disparity map, a gradient-based smoothness term is used to propagate more accurate disparity values to the areas with similar color by the assumption that the disparity in the neighborhood should be similar if the intensity is similar:

$$\mathcal{L}_{sm}(G) = \mathop{\mathbb{E}}_{u \in \tilde{x}, v \in N(u), \tilde{x} \sim P_G}[\exp(\gamma - \beta |\nabla(I_l)_{uv}|) ||\tilde{x}_u - \tilde{x}_v||_1] \quad (4)$$

where $\tilde{x}_u$ is the disparity value of a pixel u in the refined disparity map $\tilde{x}$ from the generator. $\tilde{x}_v$ is the disparity value of a pixel v in the neighborhood $N(u)$ of pixel u. $\nabla(I_l)_{uv}$ is the gradient in the left intensity image from pixel u to pixel v. It is calculated from the left intensity image considering the diagonal, left and right directions. $\beta \geq 0$ and $\gamma \geq 0$ are responsible for how close the disparities are if the intensities in the neighborhood are similar.

(5) Finally, our final object function is:

$$G^* = \arg \min_G \max_{D_i} [\theta_1 \mathcal{L}_{L_1}(G) + \theta_2 \mathcal{L}_{sm}(G) + \theta_3 \mathcal{L}_c(G) - \theta_4 \sum_{i=1}^{M} \mathcal{L}_{wgan}(G, D_i)] \quad (5)$$

where θ_1, θ_2, θ_3, θ_4 are the weights for the different loss terms. There are no ground truth terms in Equation 1-5. Thus, the training is unsupervised.

3. EXPERIMENTAL RESULTS

The network is implemented using TensorFlow [16] and trained & tested using an Intel Core i7-7820HK processor (quad-core, 8MB cache, up to 4.4GHZ) and Nvidia Geforce GTX 1080Ti. In the following experiments, the inputs to the neural network were first normalized to [-1, 1]. After that, the input was flipped vertically with a 50% chance to double the number of training samples. Weights of all the neurons were initialized from a Gaussian distribution (standard deviation 0.02, mean 0). We trained each model for 500 epochs using disparity values calculated by different stereo algorithms on Kitti2015, with a batch size 4 using Adam [17] with a momentum of 0.5. The learning rate is changed from 0.005 to 0.0001 gradually. The method in [13] is used to optimize the generator and discriminator by alternating between one step on the discriminator and then one step on the generator. We set the parameters θ_1, θ_2, θ_3, θ_4 in Equation 5 to make those four terms contribute differently to the energy function in the training process. If the difference of two initial disparity values on the same pixel is small (<0.3 pixels), we assign a large value (0.99) to their confidence weight in Equation 1. If not, we set them uniformly (1/Z). Besides the confidence estimation above, we also adopted some empirical confidence estimation for the disparity inputs in the following experiments (For more details, see the corresponding experiments). We used the L_1 distance between the estimated value and ground truth as the error. The unit is pixels.

Table 1: Test Time and Training Parameter Setting

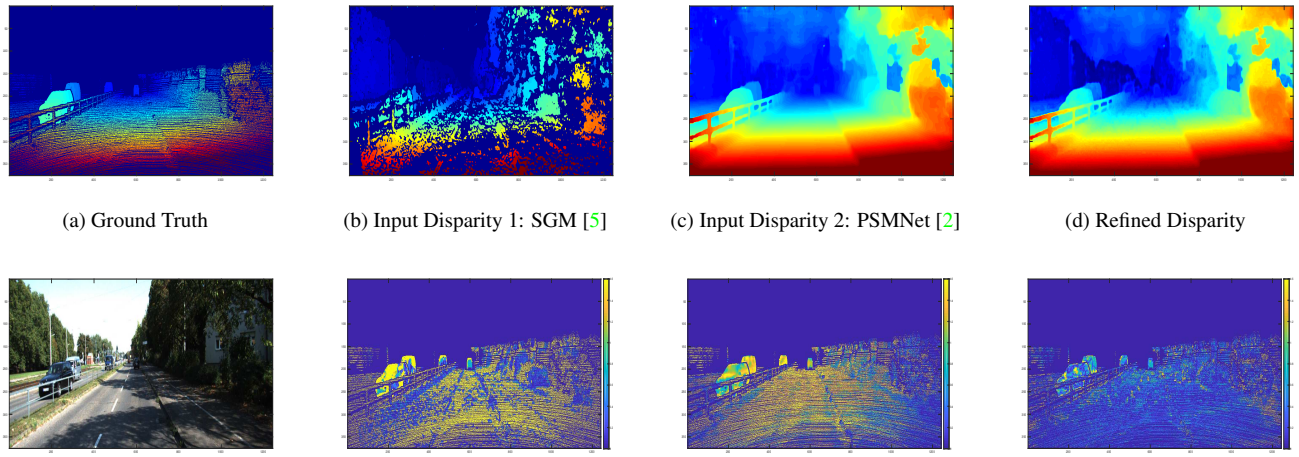
	Experiment with Real Dataset (Kitti2015)							
Para.	Test time	θ_1	θ_2	θ_3	θ_4	α	β	γ
Value	0.011 (s/frame)	1	20 to 1	0.0001 to 1K	1	3	1 to 3000	5

3.1. Stereo-stereo Fusion

We tested our network on the real Kitti2015 dataset, which used a Velodyne HDL-64E Lidar scanner to get the sparse ground truth and a 1242*375 resolution stereo camera to get stereo image pairs. The training dataset contains 400 unlabelled and labelled samples. There are another 400 samples in the test dataset. 50 samples from '000000_10.png' to '000049_10.png' in the Kitti2015 training dataset were used as our test dataset. 50 samples from '000050_50.png' to '000099_10.png' in the Kitti2015 training dataset were used as our validation dataset. The rest 700 samples were used as our training set. By flipping the training samples vertically, we doubled the number of training samples. We used the state-of-art stereo vision algorithm PSMNet [2] as one of our inputs. We used their released pre-trained model[2] on the Kitti2015 dataset to get the disparity maps. A traditional stereo vision algorithm SGM [5] is used as the second input to the network. Because we do not care about the sparsity of SGM, we set their parameters to produce more reliable disparity maps (0.78 pixels[3]). Thus, we assign big confidence values (0.8) to its valid pixels and 0 to its invalid pixels' confidences. More specifically, we used the implementation ('disparity' function) from Matlab2016b. The relevant

[2]PSMNet [2]: https://github.com/JiaRenChang/PSMNet

[3]This is a more accurate disparity but is calculated only using more reliable pixels. On average only 40% of the ground truth pixels are used. If we use all the valid ground truth to calculate its error, it is 22.13 pixels.



(a) Ground Truth (b) Input Disparity 1: SGM [5] (c) Input Disparity 2: PSMNet [2] (d) Refined Disparity

(e) Scene (f) Input Disparity 1 Error: SGM [5] (g) Input Disparity 2 Error: PSMNet [2] (h) Refined Disparity Error

Fig. 2: We trained the proposed network to fuse the initial disparity maps (b), (c) into a refined disparity map (d) for the same scene (e) from the Kitti2015 dataset [15]. (a) is the corresponding ground truth. (f), (g), (h) are the errors of (b), (c), (d). The colorbars (from blue to white) corresponds to 0 - 1.6 pixels and the lighter pixel have bigger error in (f), (g), (h).

parameters are: 'DisparityRange' [0, 160], 'BlockSize' 5, 'ContrastThreshold' 0.99, 'UniquenessThreshold' 70, 'DistanceThreshold' 2. The settings of the neural network are shown in Table 1 (For more details, see Table 1 in the supplementary material). We compared the algorithm with the state-of-art technique [12, 9] in stereo-stereo fusion and also stereo vision inputs [2, 5]. We compared our method with the supervised method in Sdf-GAN [12]. We trained Sdf-GAN on a synthetic garden dataset first and then fine-tuned the pre-trained model on the Kitti2015. 150 labeled samples from '00050_10.png' to '000199_10.png' in the initial training dataset were used for Sdf-GAN fine-tuning. The performance of the proposed algorithm (0.83 pixels) (See Table 2) is better than Sdf-GAN (1.17 pixels). The reason is because Sdf-GAN does not generalize well in the real environment. However, the proposed algorithm is less affected by such problems because the unsupervised method can use the unlabelled data directly.

For qualitative results, see Figure 2. Compared with SGM and PSMNet, the fused results are more dense, accurate and preserve the details better (eg: tree). But it fails on the sky because we treated the pixels (disparity = 0) as invalid (confidence = 0) in SGM. However, the disparity values in the sky area from PSMNet are all larger than 0 (confidence >0). So, the PSMNet misleads the network to adopt their disparity value as the initialization. Thus, the wrong confidence measurement can bring big error to the refined disparity map. It can be solved by adding more cues, such as semantic meaning, to make the confidence measurements more accurate.

Table 2: Average error (pixel) on Kitti2015

Source Error		**Fused Algorithm Error**		
SGM [5]	PSMNet [2]	DSF [9]	Sdf-GAN [12]	Ours
0.78	1.22	1.20	1.17	0.83

3.2. Additional Experiments

We have also done ablation study experiments and Stereo-Lidar fusion. The experimental results show our superiority again. For more details, see the supplementary material (https://arxiv.org/abs/1904.10044).

4. CONCLUSION

We proposed an unsupervised method to fuse the disparity estimates of multiple state-of-art disparity/depth algorithms. The experiments have shown the effectiveness of the energy function design based on multiple cues and the efficiency of the network structure. The proposed network can be generalized to other fusion tasks based on left-right image consistency (In this paper, we only did stereo-stereo and stereo-lidar fusion). The method proposed in this paper reduces the cost of acquiring labelled data necessary for use in a supervised method. Given the algorithm's low computation cost, the combination of the proposed method and existing depth-acquisition algorithms is a good solution to obtaining higher accuracy depth maps. Future work will investigate improved methods for setting the confidence values based on the initial disparity values and type of sensor.

5. REFERENCES

[1] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017, vol. 2, p. 7.

[2] Jia-Ren Chang and Yong-Sheng Chen, "Pyramid stereo matching network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5410–5418.

[3] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4040–4048.

[4] Dominik Honegger, Torsten Sattler, and Marc Pollefeys, "Embedded real-time multi-baseline stereo," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5245–5250.

[5] Heiko Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 2, pp. 807–814.

[6] Carlo Dal Mutto, Pietro Zanuttigh, and Guido Maria Cortelazzo, "Probabilistic tof and stereo data fusion based on mixed pixels measurement models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 11, pp. 2260–2272, 2015.

[7] Giulio Marin, Pietro Zanuttigh, and Stefano Mattoccia, "Reliable fusion of tof and stereo depth driven by confidence measures," in *European Conference on Computer Vision*. Springer, 2016, pp. 386–401.

[8] Gianluca Agresti, Ludovico Minto, Giulio Marin, and Pietro Zanuttigh, "Deep learning for confidence information in stereo and tof data fusion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 697–705.

[9] Matteo Poggi and Stefano Mattoccia, "Deep stereo fusion: combining multiple disparity hypotheses with deep-learning," in *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016, pp. 138–147.

[10] Will Maddern and Paul Newman, "Real-time probabilistic fusion of sparse 3d lidar and dense stereo," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 2181–2188.

[11] Kihong Park, Seungryong Kim, and Kwanghoon Sohn, "High-precision depth estimation with the 3d lidar and stereo fusion," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2156–2163.

[12] Can Pu, Runzi Song, Radim Tylecek, Nanbo Li, and Robert B Fisher, "Sdf-gan: Semi-supervised depth fusion with multi-scale adversarial networks," *arXiv preprint arXiv:1803.06657*, 2018.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5769–5779.

[15] Moritz Menze, Christian Heipke, and Andreas Geiger, "Joint 3d estimation of vehicles and scene flow," in *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: A system for large-scale machine learning.," in *OSDI*, 2016, vol. 16, pp. 265–283.

[17] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.