Sdf-GAN: Semi-supervised Depth Fusion with Multi-scale Adversarial Networks

Can Pu, Runzi Song, Nanbo Li, Robert B Fisher

University of Edinburgh

Abstract. Fusing disparity maps from different algorithms to exploit their complementary advantages is still challenging. Uncertainty estimation and complex disparity relationships between neighboring pixels limit the accuracy and robustness of the existing methods and there is no common method for depth fusion of different kind of data. In this paper, we introduce a method to incorporate supplementary information (intensity, gradient constraints etc.) into a Generative Adversarial Network to better refine each input disparity value. By adopting a multi-scale strategy, the disparity relationship in the fused disparity map is a better estimate of the real distribution. The approach includes a more robust object function to avoid blurry edges, impaints invalid disparity values and requires much fewer ground data to train. The algorithm can be generalized to different kinds of depth fusion. The experiments were conducted through simulation and real data, exploring different fusion opportunities: stereo-monocular fusion from coarse input, stereo-stereo fusion from moderately accurate input, fusion from accurate binocular input and Time of Flight sensor. The experiments show the superiority of the proposed algorithm compared with the most recent algorithms on the public Scene Flow and SYNTH3 datasets. The code is available from `https://github.com/Canpu999`

Keywords: Depth fusion, Disparity fusion, GAN, Stereo Vision, Monocular Vision, Time of Flight

1 Introduction

With recent improvements in depth sensing devices and algorithms, depth information is now easily accessible to researchers. However, each algorithm has its own advantages and disadvantages, with the result that no algorithm can perform accurately and robustly in all general scenes. For example, active illumination devices such as ToF (Time of Flight) sensors and structured light cameras [1] estimate the depth information accurately regardless of the scene content but struggle on low reflective surfaces or outdoors. Stereo vision algorithms ([2,3,4]) work better outdoors and perform accurately on high texture areas but behave poorly in repetitive or textureless regions. Monocular vision algorithms ([5]) work robustly in the textureless areas but tend to produce blurry depth edges. Thus, fusing multiple depth maps from different kind of algorithms or devices and utilizing their complementary strengths to get more accurate depth information is becoming a core technique for various applications.

The traditional pipeline for the majority of the fusion algorithms ([6,7,8,9,10]) is: 1) estimate depths from the different sensors, 2) estimate associated confidence maps and

3) apply a specific fusion algorithm based on the confidence maps to get a refined disparity map. This approach has three potential problems. Estimating the confidence map for different sensors is very hard, which limits the robustness and accuracy. Second, estimating the disparity relationships between neighboring pixels in a general scene is hard with human prior knowledge. Finally, there is no common methodology for different kinds of depth fusion, such as stereo-stereo fusion, monocular-stereo fusion, stereo-ToF fusion. Thus, researchers have designed different methods for different fusion tasks. However, the recent fusion method [11] based on end to end deep learning has provided a general solution to different kind of fusion but limited accuracy and robustness, in part due to not exploiting other associated information to help the network make judgement. It also did not exploit the disparity relationship between neighboring pixels.

In this paper, a model based on conditional generative adversarial network [12] is proposed to solve the three problems listed above, by designing an efficient network structure and an elegant robust object function. In addition to the initial disparity maps the network input also includes some other information, e.g. intensity and gradients (see Figure 1), in order to facilitate the selection of a more accurate disparity value from several algorithms. This avoids having to design a manual confidence measure for different sensors and allows a common methodology for different kinds of sensors. Discriminating disparity patches of different sizes from the GAN generator and ground truth by a multi-scale strategy in GAN discriminator (See Figure 3) also improves performance. To preserve and exploit the local information better, some successful ideas about local structure from Unet [13] and Densenet [14] have been used (See Figure 2 and Figure 3). To help the network refine the disparity maps accurately and robustly a novel objective function was designed. Gradient information is incorporated as a weight into the L1 distance to ensure that the disparity values at the edges are accurate and reduces edge blur. A smoothness term helps the network propagate the accurate disparity values at edges to adjacent areas, which impaints regions with invalid disparity values. The Wasserstein distance [15,16] replaced the Jensen-Shannon divergence [17] for GAN loss to reduce training difficulties and avoid mode collapse. Disparity patches of different sizes are discriminated by the discriminator to improve the disparity relationship between neighboring pixels.

Our alternative semi-supervised approach provides discriminator with larger amount of samples produced by generator than ground truth, which results in less ground truth requirement but can achieve the accuracy similar to our fully-supervised method.

Section 2 reviews key previous disparity fusion algorithms and also recent advances in GAN networks. Section 3 presents the new fusion model including the objective function and network structure. Section 4 presents the results of experiments conducted with synthetic and real data. Synthetic data include noise at different levels to demonstrate the effectiveness of the objective function. Real data also demonstrates the improved performance of the proposed approach and includes inputs from : monocular-stereo fusion from low quality input, stereo-stereo fusion from medium quality input, stereo-ToF fusion from high quality input and stereo-synthetic fusion from extremely high quality input.

Contributions: We have:

1. Improved fusion accuracy by making the network learn the disparity relationship between pixels without any prior knowledge.
2. Reduced the ground truth requirements drastically by using the proposed semi-supervised strategy.
3. Increased robustness by fusing intensity and gradient information as well as the depth data.
4. Proposed a common network methodology allowing different kinds of sensor fusion without requiring detailed knowledge of the performance of each sensor type.

2 Related Work

Recently, the approach of fusing depth maps from different sensors (eg: stereo-ToF depth fusion) has become popular. The majority of the fusion work (eg: [7,8,9,10]) share the same pipeline architecture, which estimates the uncertainty of each pixel first and then refines the depth map based on those confidence maps. A recent survey is in [6]. More recently, Dal Mutto et al. [7] used the IR frequency etc. of a ToF sensor to estimate the depth map uncertainty and used the similarity of image patches in the stereo images to estimate the confidence of the stereo depth map. Then a MAP-MRF framework refined the depth map. Later, Marin et al. [8] also utilized sensor physical properties to estimate the confidence for the ToF depth map and used an empirical model based on the global and local cost of stereo matching to calculate the confidence map for the stereo vision sensor. The extended LC (Locally Consistent) technique was used to fuse the depth maps based on each confidence map. To get a more accurate confidence map for fusion, Agresti et al. [9] used a simple convolution neural network for uncertainty estimation and then used the LC technique in [8] for the fusion. In addition to the work in stereo-ToF fusion above, Facil et al. [10] used a weighted interpolation of depths from a monocular vision sensor and a multi-view vision sensor based on the likelihood of each pixel to do the depth fusion.

The methods of the above approaches have two issues. 1) Estimating the confidence map for each type of sensor accurately is hard, which limits the accuracy of the refined depth map and makes the system unstable. 2) Accurately modeling the complex disparity relationships between neighboring pixels in random scenes is challenging, which limits the accuracy of the refined disparity map.

The other class of depth fusion methods is based on deep learning. The method proposed here belongs to this class and is the first to solve the two critical problems above simultaneously to our knowledge. Some researchers [9,18] have estimated the confidence maps for different sensors with deep learning methods and then incorporated the confidence as weights into the classical pipeline to refine the disparity map. However, these methods treat the confidence maps as an intermediate result and no one has trained the neural network to do the fusion from end to end directly and taken both the depth and confidence information into account simultaneously. For example, Poggi and Mattoccia [11] selected the best disparity value for each pixel from the several algorithms. Thus, they formulated depth fusion as a multi-labelling deep network classification problem. However, the method only used the disparity maps from the sensors and neglected other associated image information (*e.g.* intensities, gradients). The approach did not exploit the real disparity relationship between neighboring pixels.

The recent development of the GAN led to the foundation of the approach proposed here. The GAN [17] was firstly proposed by Goodfellow, who trained two neural networks (generator and discriminator) simultaneously to make the distribution of the output from the generator approximate the real data distribution by a minimax two-player strategy. To control the data generation process, Mirza and Osindero [12] conditioned the model on additional information. There are many variants based on the initial GAN as seen in this survey [19]. More recently, some researchers [15,16] used the Wasserstein distance to measure the distance between the model distribution and the real distribution, which reduced the difficulty of training the GAN drastically. It also reduced mode collapse to some extent. GANs have been applied to problems other than disparity fusion. For example, Isola et al. [20] trained a GAN to translate between image domains which can be also used to transfer the initial disparity maps from several sensors into a refined disparity map. However, the design proposed in [20] neglects information useful for disparity fusion area, which limits the accuracy of the refined disparity map.

In summary, there are previously developed methods for depth fusion based on both the algorithmic pipeline and emerging deep network techniques. In this paper, we combine image evidence as well as raw depth to give a more robust objective function. This is implemented in an end-to-end GAN. We are the first to our knowledge to use the basic structure of the GAN to learn the complex disparity relationship between neighboring pixels to improve fusion accuracy.

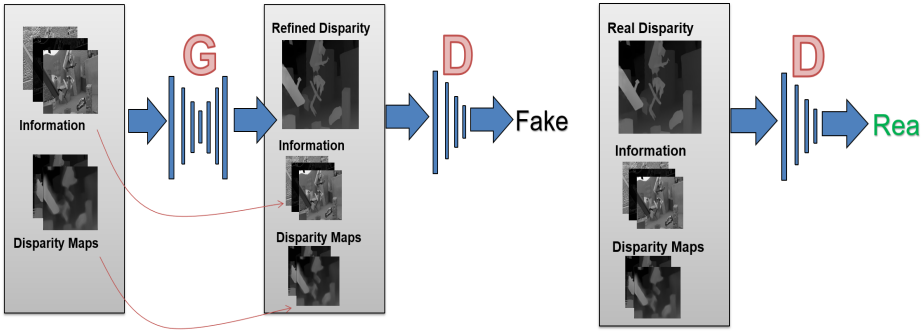
3 Methodology

First we introduce the proposed general model for depth fusion and then the new energy functions for the supervised and semi-supervised methods. These functions will make GAN training simple and the results more accurate and robust. Finally, the end-to-end GAN network structure is presented.

3.1 Generative Model

The Conditional GAN [12] is used to map a random noise distribution to the real data distribution constrained by observed conditions. The generator G is trained to produce output which cannot be classified as "fake" by the discriminator D. Simultaneously, the discriminator D is trained to become better at distinguishing that the input from generator G is fake and the input from the real data distribution is true. By adopting a minimax two-player game strategy, the two neural networks $\{G, D\}$ make the output distribution from the generator approximate the real data distribution. The full system diagram is shown in Figure 1.

Why neglect the random noise input? Given the multiple disparity inputs for the same scene are based on the underlying ground truth disparity map but have uncorrelated errors already, we neglect the additional random noise distribution input and only input the conditional data (disparity maps from the input algorithms and supplementary intensity and gradient information) into our model to produce a deterministically



(a) Negative examples (b) Positive examples

Fig. 1: We train a conditional GAN without random noise input to map disparity maps from several input algorithms to the ground truth based on associated image information. The generator, G, tries to predict a refined disparity map close to the ground truth. The discriminator, D tries to discriminate whether the input is fake (from G) or true (from the ground truth). The generator and discriminator can see both the supplementary information and disparity inputs simultaneously as the conditional information.

refined disparity map. This is obviously different from the previous work based on random noise inputs into GANs (*e.g.* [17,12]). By doing this, our model learns a complex and specific function which maps the different disparity maps from the same scene into the ground truth disparity map.

Why input the supplementary information? The work in [11] input only the disparity maps into the neural network without any additional information and estimated the best disparity value for each pixel from 8 algorithms, but still with errors. The proposed algorithm inputs supplementary information (*e.g.* the stereo reconstructed error estimates, the minimum global cost from some stereo vision algorithms, the gradient direction and magnitude) to the proposed neural network to help it make a more informed decision.

Why not one GAN for each input stream? From a mathematical perspective, fusing multiple disparity maps from the same scene could be framed as estimating the joint probability that the disparity maps from the different algorithms can be transformed into the same accurate refined disparity map. The work in [21] adopted parallel GANs (each for one input stream) and calculated the joint probability of all the domains by sharing the local weights between the different GANs. We concatenate all the disparity maps and supplementary information together and feed them into a single GAN to calculate the joint probability by sharing the same weights end to end, which reduces space and time complexity drastically.

3.2 Objective Function

To let the generator produce a more accurate refined disparity map, the objective function is designed as follows:

(1) To encourage the disparity value of each pixel to approximate the ground truth and to avoid blur at scene edges (such as occur with the Monodepth method [5]), a gradient-based L1 distance is used, which applies a larger weight on the disparity values at the scene edges:

$$\mathcal{L}_{L_1}(G) = \mathop{\mathbb{E}}_{x\sim P_R, \tilde{x}\sim P_G}[\exp(\alpha|\nabla(I)|)||x-\tilde{x}||_1] \tag{1}$$

In Equation 1, x is the ground truth and $\tilde{x}$ is the refined disparity map from the generator. $\nabla(I)$ is the gradient of the gray image in the scene. $\alpha \geq 0$ weights the gradient. $||\bullet||_1$ is L1 distance. The goal is to encourage disparity estimates near image edges (larger gradients) to get closer to the ground truth.

(2) A gradient-based smoothness term is added to propagate more reliable disparity values from near image edges to the other areas in the image under the assumption that the disparity of neighboring pixels should be similar if their intensities are similar.

$$\mathcal{L}_{sm}(G) = \mathop{\mathbb{E}}_{u\in\tilde{x}, v\in N(u), \tilde{x}\sim P_G}[\exp(1-\beta|\nabla(I)_{\tilde{x}_u}|)||\tilde{x}_u-\tilde{x}_v||_1] \tag{2}$$

In Equation 2, $\tilde{x}_u$ is a pixel in the refined disparity map $\tilde{x}$ from the generator. $\tilde{x}_v$ is a pixel in the neighborhood of pixel $\tilde{x}_u$. $\nabla(I)_{\tilde{x}_u}$ is the gradient at pixel $\tilde{x}_u$ in the intensity image of the scene. $\beta \geq 0$ controls the extent that the disparities are similar if the intensities are similar.

(3) The underlying assumption in $\mathcal{L}_{L_1}(G)$ is that the disparity relationship between pixels is independent. The disparity relationship in $\mathcal{L}_{sm}(G)$ is too simple to describe the real disparity distribution between neighbors in the real situation. To help the generator produce a disparity map whose disparity neighborhood relationships are closer to the real distribution, the network inputs disparity patches from the generator and the ground truth into the discriminator, which outputs the probability of the generated samples being from the same distribution as the ground truth. This probability is then used to update the generator through its loss function. By outputting that probability in different perceptive field size (we call it multi-scale strategy), the disparity neighborhood relationship in different window size from generator is forced by discriminator to get closer to the real distribution. In Equations 3 and 4 below, D_i is the probability at scale i that the input patch to the discriminator is from the real distribution at scale i. λ is the penalty coefficient and $\hat{x}$ are the random samples [16].

$$\mathcal{L}_{JS-GAN}(G, D_i) = \mathop{\mathbb{E}}_{x\sim P_R}[\log(D_i(x))] + \mathop{\mathbb{E}}_{\tilde{x}\sim P_G}[\log(1-D_i(\tilde{x}))] \tag{3}$$

To avoid GAN mode collapse during training and alleviate other training difficulties, we have also investigated replacing $\mathcal{L}_{JS-GAN}(G, D_i)$ with the Improved WGAN loss function [16]:

$$\mathcal{L}_{WGAN}(G, D_i) = \mathop{\mathbb{E}}_{\tilde{x}\sim P_G}[D_i(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim P_R}[D_i(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim P_{\hat{x}}}[(||\nabla_{\hat{x}} D_i(\hat{x})||_2 - 1)^2] \tag{4}$$

The experiments explored the difference in performance of these two GAN loss functions. We let $\mathcal{L}_{GAN}(G, D_i)$ be either $\mathcal{L}_{JS-GAN}(G, D_i)$ or $\mathcal{L}_{WGAN}(G, D_i)$ in the following context. The difference of performance of both the single and multi-scale strategy will also be explored.

(4) By inputting the output of the generator and its corresponding real disparity map into the discriminator, the discriminator is trained in a fully supervised manner through the disparity values and neighborhood relationships. The L1 distance in step (1) provides a guide to the generator about the disparity values as an alternative to real disparity values. Thus, in semi-supervised mode, the discriminator can be trained with a small ground truth dataset and additional more random disparity maps produced by the generator, which will only learn the underlying relationship between neighboring pixels. This semi-supervised strategy reduces the amount of training data needed while still achieving a similar performance.

The combined loss function for the fully supervised approach is:

$$\mathcal{L}_{sup}(G, D) = \theta_1 \mathcal{L}_{L_1}(G) + \theta_2 \mathcal{L}_{sm}(G) + \theta_3 \sum_{i=1}^{M} \mathcal{L}_{GAN}^{<x,\tilde{x}>}(G, D_i) \tag{5}$$

The combined loss function for the semi-supervised approach is:

$$\mathcal{L}_{semi}(G, D) = \theta_1 \mathcal{L}_{L_1}(G) + \theta_2 \mathcal{L}_{sm}(G) + \theta_3 \sum_{i=1}^{M} \mathcal{L}_{GAN}^{<x,\tilde{x}>}(G, D_i) + \theta_4 \sum_{i=1}^{M} \mathcal{L}_{GAN}^{(x,\tilde{x})}(G, D_i) \tag{6}$$

In Equation 5, $< x, \tilde{x} >$ denotes a pair of discriminator input and x is the ground truth for $\tilde{x}$. In Equation 6, $(x, \tilde{x})$ denotes a pair of discriminator input and x is not the ground truth for $\tilde{x}$. M is the number of the scales. θ_1, θ_2, θ_3, θ_4 are the weights for the different loss terms.

3.3 Network architectures

The architectures from [20,22,14] are adapted here for the generator and discriminator. The generator and discriminator use dense blocks to increase local nonlinearity in areas with more local detail. Transition layers change the size of the feature maps to reduce the time and space complexity [14] . In each dense block and transition layer, modules of the form ReLu-BatchNorm-convolution are used. The growth rate k for each dense block is dynamic (Unlike [14]). In the generator, the skip connections [13] from the previous layer to the latter layer preserve the local detail in order not to lose information after the network bottleneck. The discriminator uses a multi-scale strategy to classify if the disparity patch with different sizes and locations is from real distribution or not. Figure 2 and Figure 3 shows the main architecture of the generator and discriminator and some important hyperparameters as used in the experiments. For more details, see the supplementary materials.

4 Experimental Evaluation

The network is implemented using TensorFlow[23] and trained or tested with Intel Core i7-7820HK processor (quad-core, 8MB cache, up to 4.4GHZ) and Nvidia Geforce GTX

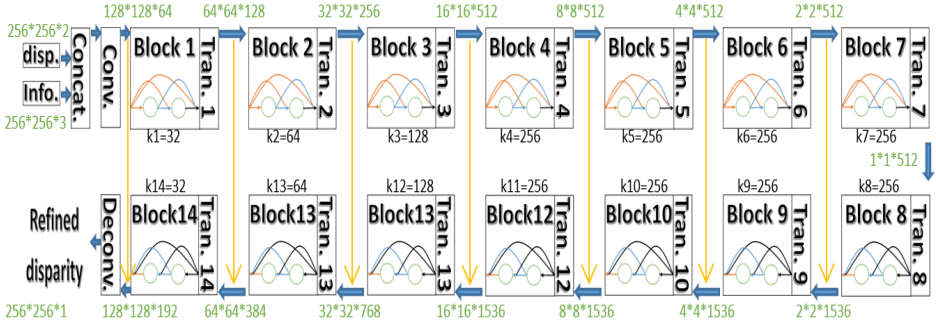


Fig. 2: This figure shows the proposed generator architecture and some important hyperparameters. The disparity maps from the two input algorithms and the supplementary information images (intensity image, direction of gradient, magnitude of gradient) are concatenated. All images are rescaled to 256*256 pixels. Each dense block has 2 internal layers and the corresponding growth rate k equals the number of input layers divided by the number of internal layers. Transition layers between dense blocks change feature-map sizes and layers. After the first convolution layer and each transition layer, long skip connections from the earlier layers connect to the later layers to preserve the local details (to avoid information loss after the bottleneck).

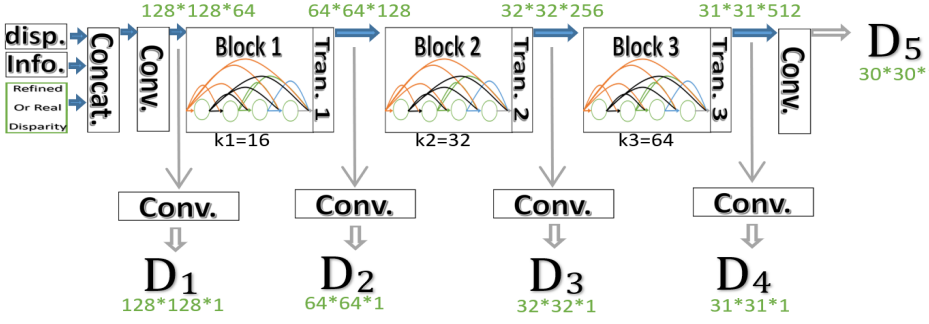


Fig. 3: This figure shows the proposed discriminator architecture and some important hyperparameters. The initial disparity maps, supplementary information, and refined or real disparity map (with 256*256 size) are concatenated as input into the discriminator. Each dense block has 4 internal layers and the corresponding growth rate k is equal to the number of input layers divided by the number of internal layers. Transition layers between dense blocks change feature-map size. After each convolution layer in the figure, a sigmoid function outputs the probability that the input disparity patch is from the ground truth at different scale to get neighborhood relationship closer to the real distribution in different perceptive field size.

1080Ti. Five groups of experiments were conducted to compare performance with recent algorithms (Fusion algorithm: [11,20,8,9] and input algorithm: [3,4,2,5]). First,

simulated data was used to analyze the four objective function designs (from Section 3.2). Secondly, four groups of experiments using different fusion input pairs (stereo-monocular, stereo-stereo, stereo-ToF, stereo-synthetic) with four different input quality-levels (low, medium, high, extremely high) have been done to show the robustness, accuracy and generality of the proposed algorithm using the Scene Flow [4] and SYNTH3 [9] datasets.

4.1 Training Procedure

For the stereo-ToF fusion the network was trained on the SYNTH3 dataset and in the other fusion tasks the network was trained on the Scene Flow dataset. All 40 SYNTH3 samples in train folder were used for training and 15 SYNTH3 samples in test folder for testing in stereo-ToF fusion experiment. 6000 samples (80%) in Scene Flow (Flying A) were used for supervised training and 600 samples (8%) for semi-supervised training. Another 1460 samples (20%) were used for testing in the other fusion experiments. It takes several minutes to train for 100 epochs on the SYNTH3 dataset. For the Scene Flow dataset, it takes on the order of 20 hours to train for 100 epochs. Inference is fast (25 frames per second) for a 256*256 input, including the data transfer time to or from the GPU.

The input disparity maps and information were first scaled into 256*256 pixels and normalized to [-1, 1]. Then the input was flipped horizontally with a 50% chance. Weights of all the neurons were initialized from a Gaussian distribution (standard deviation 0.02, mean 0). Training used a batch size of 8 in the supervised method and 4 in the semi-supervised method. The learning rate is fixed to 0.0002 from begin to end using Adam [24] with a momentum of 0.5. The method in [17] is used to optimize the generator and discriminator by alternating between one step on the discriminator and then one step on the generator. For more details, please see the supplementary material or open source code[1].

4.2 Analysis of the Objective Function by Simulation

Different levels of noise from the Gaussian distribution $N(0, \sigma^2)$ were added to the normalized ground truth in the Scene Flow dataset to get two noisy disparity maps to use as inputs. We aimed at testing the effectiveness of the objective function design from Section 3.2. Table 1 defines different combinations of the strategies that were evaluated, based on the objective functions defined in Section 3.2.

In the simulation, the weights are: $\alpha = 0.01$, $\beta = 1000$, $\lambda = 0.001$. In Equation 5, $\theta_1 = 199$, $\theta_2 = 1$, $\theta_3 = 1$. in Equation 6, $\theta_1 = 45$, $\theta_2 = 15$, $\theta_3 = 0.5$, $\theta_4 = 0.5$. All the other conditions are the same for the 4 models above in the simulation. The error measure for all the experiments uses the L1 distance: $\sum |result - ground\_truth|_1/N$, where N is the number of pixels.

Table 2 shows the performance of each model for 4 different noise levels (plus the error of the input images arising from the added noise). The supervised and semi-supervised methods achieved the similar good performance (smallest error). The error

[1] Our code: `https://github.com/Canpu999`

Table 1: Model definition

Model Name	Combination
Supervised	WGAN(Equation 4) + multiscale(M=5) + supervised(Equation 5)
Semi	WGAN(Equation 4) + multiscale(M=5) + semi-supervised(Equation 6)
Monoscale	WGAN(Equation 4) + monoscale(M=1) + supervised(Equation 5)
JS-GAN	JS-GAN(Equation 3) + multiscale(M=5) + supervised(Equation 5)

Table 2: Average error of each model on Scene Flow test dataset (1460 samples)

Noise Std.	input 1	input 2	JS-GAN	Monoscale	Semi	Supervised
$\sigma = 0.02$	0.0159	0.0159	0.0067	0.0051	**0.0040**	0.0048
$\sigma = 0.04$	0.0319	0.0319	0.0061	0.0059	**0.0043**	0.0053
$\sigma = 0.08$	0.0638	0.0638	0.0083	0.0070	**0.0063**	0.0068
$\sigma = 0.16$	0.1276	0.1276	0.0104	0.0081	0.0094	**0.0073**

of the refined disparity map output by each network is much lower than the error of the input disparity maps. In the remaining experiments, only the multi-scale supervised and semi-supervised networks are used with WGAN.

4.3 Robustness and Accuracy Test

Given that the proposed network does not need confidence values from the specific sensors, the network architecture can be generalized to different fusion tasks and isn't influenced by the specific fusion sources. Thus, the following experiments will input different quality disparity maps to test the robustness and accuracy of the proposed algorithm. On input, all disparity maps are normalized to [0,1] by dividing by the maximum disparity value. Errors will be calculated on the normalized scale. $\sum |Disp_{calc} - Disp_{GT}|/(N * Disp_{max})$, where N is the number of pixels and $Disp_{calc}$, $Disp_{GT}$, $Disp_{max}$ is calculated disparity value, ground truth and the maximum disparity value set by user. For the stereo-monocular fusion, stereo-stereo fusion, and stereo-synthetic fusion, experiments used the intensity, gradient magnitude and gradient direction as associated information input. As for information in ToF-stereo fusion, besides those above, the intensity and amplitude of ToF received signal were also included as inputs. DSF [11] was trained for 10 epochs (about 5 hours per epoch when the number of training samples is 6000) and Pix2pix [20] was trained for 100 epochs (0.15 hour per epoch when the number of training samples is 6000) in all the experiments. Their speed is stable in all of the experiments.

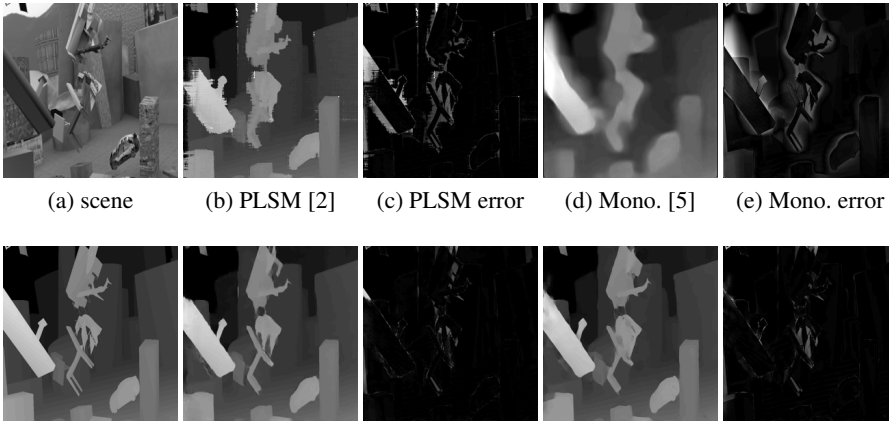
Low Quality Input into Stereo-monocular Fusion (See Figure 4.) The monocular depth estimation algorithm is usually less accurate than the stereo vision algorithms. Stereo vision algorithm PLSM [2] and monocular vision algorithm Monodepth [5] were used to input the relevant low quality disparity maps. Monodepth was trained on the Scene Flow dataset with 50 epochs to get its left disparity map (Fig 4d). PLSM with semi-global matching computed the left disparity map without refinement (Fig 4b). In

this experiment, the weight for each term was: $\alpha = 0.5$, $\beta = 100$, $\lambda = 0.001$. In Equation 5, $\theta_1 = 199$, $\theta_2 = 1$, $\theta_3 = 1$. In Equation 6, $\theta_1 = 199$, $\theta_2 = 1$, $\theta_3 = 0.5$, $\theta_4 = 0.5$.

Table 3: Average error of stereo-monocular fusion on Scene Flow (1460 test samples)

Training Data	**PLSM** [2]	**Monodepth** [5]	**DSF** [11]	**Pix2pix** [20]	**Supervised**	**Semi**
$Num = 600$	0.0188	0.0258	0.0156	0.0227	0.0152	**0.0125**
$Num = 6000$	0.0188	0.0258	0.0146	0.0207	**0.0121**	Null

The relevant error of each algorithm is shown in Table 3. The supervised method (Num=6000) and the semi-supervised method (Num=600) have achieved similar top performances while the semi-supervised method uses much less ground truth training data (9 times less than the supervised method). A qualitative result comparison can be seen in Figure 4.



(a) scene (b) PLSM [2] (c) PLSM error (d) Mono. [5] (e) Mono. error

(f) ground truth (g) supervised 1 (h) error 1 (i) semi 2 (j) error 2

Fig. 4: A qualitative result with inputs from PLSM [2] and Monodepth [5] in stereo-monocular fusion. The lighter pixels represent bigger values.

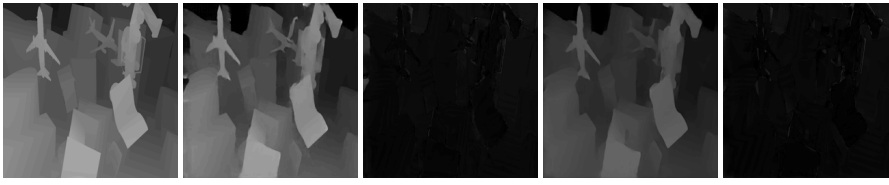
Medium Quality Input in Stereo-stereo Fusion Results from stereo vision algorithms PLSM [2] and MC-CNN [3] were the medium quality disparity maps input into the network. The network weights are the same as those in the stereo-monocular fusion above. The relevant results are shown in Table 4. The same conclusion can be made as with the stereo-monocular fusion: the proposed method is accurate and robust to

low and medium quality inputs. An example result of stereo-stereo fusion is shown in Figure 5.

Table 4: Average error of stereo-stereo fusion on Scene Flow (1460 test samples)

Training Data	**PLSM** [2]	**MC-CNN** [3]	**DSF** [11]	**Pix2pix** [20]	**Supervised**	**Semi**
$Num = 600$	0.0188	0.0138	0.0132	0.0208	0.0146	**0.0121**
$Num = 6000$	0.0188	0.0138	0.0127	0.0194	**0.0108**	Null

(a) scene (b) PLSM [2] (c) PLSM error (d) mc-cnn [3] (e) mc-cnn error



(f) ground truth (g) supervised 1 (h) error 1 (i) semi 2 (j) error 2

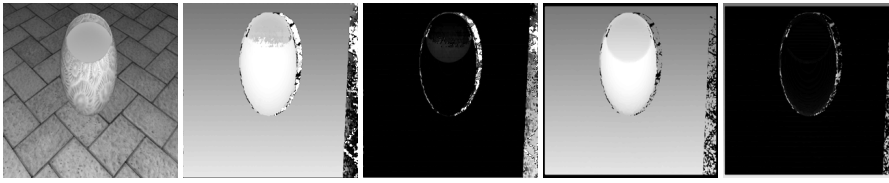
Fig. 5: One qualitative result when inputing medium quality disparity from PLSM [2] and MC-CNN [3] in stereo-stereo fusion. The lighter pixels represent bigger values.

High Quality Input into Stereo-ToF Fusion In this experiment, the weights for each term are: $\alpha = 1, \beta = 65, \lambda = 0.001$. In Equation 5, $\theta_1 = 195, \theta_2 = 5, \theta_3 = 1$. In Equation 6, $\theta_1 = 195, \theta_2 = 5, \theta_3 = 0.5, \theta_4 = 0.5$. The network was trained on the SYNTH3 dataset (40 training and 15 test samples). Semi-global matching from OpenCV was used to get the stereo disparity map, with the pointwise Birchfield-Tomasi metric, 7*7-pixel window size and 8-path optimization. The initial ToF disparity map was projected onto the right stereo camera image plane and upsampled. The preprocessed disparity maps and supplementary information were input into the network. Limited by the very small number of training samples, the network doesn't reach its best performance. But, compared with the input disparity maps and the other algorithms, the proposed method performs slightly better (See Table 5). The average error included the invalid pixels

in the input disparity maps. The proposed supervised method (0.0065) is less accurate than the semi-supervised method (0.0061) when the size of training dataset is same. One example of ToF-stereo fusion is shown in Figure 6.

Table 5: Average error of ToF-stereo fusion on SYNTH3 (15 test samples)

Training Data	**SGM** OpenCV	**ToF**	**DSF** [11]	**Pix2pix** [20]	**LC** [8]	**DLF** [9]	**Supervised**	**Semi**
$Num = 40$	0.0114	0.0136	0.0134	0.0120	0.0069	0.0068	0.0065	**0.0061**



(a) scene (b) ToF (c) ToF error (d) SGM opencv (e) SGM error

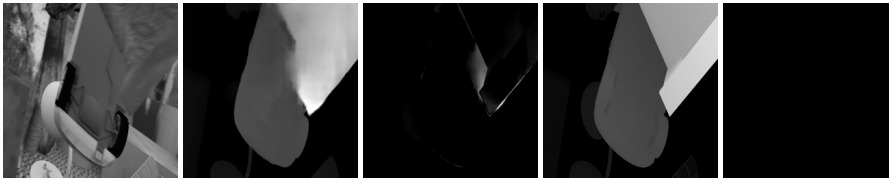
(f) ground truth (g) Supervised 1 (h) error 1 (i) Semi 2 (j) error 2

Fig. 6: One qualitative result for ToF-stereo fusion with many invalid pixels input. The inputs are from ToF and disparity calculation algorithm using SGM in OpenCV. The figure also shows that the proposed algorithm is good at "hole filling".

Extremely-high Quality Input in Stereo-synthetic Fusion To get an extremely-high quality input pair, Dispnet [4] and a very accurate synthetic disparity input (ground truth + noise from Gaussian distribution $N(0, 0.004)$) were used. In this experiment, the weights for each term are: $\alpha = 0.01$, $\beta = 1000$, $\lambda = 0.001$. In Equation 5, $\theta_1 = 495$, $\theta_2 = 5$, $\theta_3 = 1$. In Equation 6, $\theta_1 = 495$, $\theta_2 = 5$, $\theta_3 = 0.5$, $\theta_4 = 0.5$. From this experiment, the same conclusion as above can be drawn. Table 6 shows that the proposed fusion algorithm can refine the input disparity maps even if input disparity maps are highly accurate. An example result from stereo-synthetic fusion is shown in Figure 7.

Table 6: Average error of stereo-synthetic fusion on Scene Flow (1460 test samples)

Training Data	**Dispnet** [4]	**Synthetic**	**DSF** [11]	**Pix2pix** [20]	**Supervised**	**Semi**
$Num = 600$	0.0050	0.0032	0.0029	0.0139	0.0079	**0.0028**
$Num = 6000$	0.0050	0.0032	0.0025	0.0091	**0.0023**	Null



(a) scene (b) dispnet [4] (c) dispnet error (d) synthetic (e) synthetic error

(f) ground truth (g) supervised 1 (h) error 1 (i) semi 2 (j) error 2

Fig. 7: We show one qualitative result in stereo-synthetic fusion with a special case (One input is extremely accurate while the other is not). The algorithm still works robustly and accurately.

5 Conclusions

The paper has presented a method to refine disparity maps based on the output of multiple depth calculation algorithms and other supplementary image information (*e.g.* intensity, gradient magnitudes) by training a conditional multi-scale GAN. The trained GAN generalized enough to perform different fusion tasks and achieved better accuracy compared with several recent fusion algorithms. The objective function and network architecture are novel and effective. In addition, the proposed semi-supervised method greatly reduces the amount of ground truth training data needed, while achieving comparable performance with the proposed supervised method. The code and the experimental material from this paper are available at: `https://github.com/Canpu999`.

6 ACKNOWLEDGMENTS

The research is fully funded by TrimBot2020 project [Grant Agreement No. 688007, URL: `http://trimbot2020.webhosting.rug.nl/`] in the European Union

Horizon 2020 programme. We thank Chengyang Zhao, Radim Tylecek and Marija Jegorova for giving us good advice on this paper.

References

1. Zanuttigh, P., Marin, G., Dal Mutto, C., Dominio, F., Minto, L., Cortelazzo, G.M.: Time-of-flight and structured light depth cameras. Springer (2016)
2. Horna, L., Fisher, R.B.: 3d plane labeling stereo matching with content aware adaptive windows. In: VISIGRAPP (6: VISAPP). (2017) 162–171
3. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research **17** (2016) 1–32
4. 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4040–4048
5. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: CVPR. Volume 2. (2017) 7
6. Nair, R., Ruhl, K., Lenzen, F., Meister, S., Schäfer, H., Garbe, C.S., Eisemann, M., Magnor, M., Kondermann, D.: A survey on time-of-flight stereo fusion. In: Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications. Springer (2013) 105–127
7. Dal Mutto, C., Zanuttigh, P., Cortelazzo, G.M.: Probabilistic tof and stereo data fusion based on mixed pixels measurement models. IEEE transactions on pattern analysis and machine intelligence **37**(11) (2015) 2260–2272
8. Marin, G., Zanuttigh, P., Mattoccia, S.: Reliable fusion of tof and stereo depth driven by confidence measures. In: European Conference on Computer Vision, Springer (2016) 386–401
9. Agresti, G., Minto, L., Marin, G., Zanuttigh, P.: Deep learning for confidence information in stereo and tof data fusion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 697–705
10. Fácil, J.M., Concha, A., Montesano, L., Civera, J.: Single-view and multi-view depth fusion. IEEE Robotics and Automation Letters **2**(4) (2017) 1994–2001
11. Poggi, M., Mattoccia, S.: Deep stereo fusion: combining multiple disparity hypotheses with deep-learning. In: 3D Vision (3DV), 2016 Fourth International Conference on, IEEE (2016) 138–147
12. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
13. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer (2015) 234–241
14. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017)
15. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning. (2017) 214–223
16. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems. (2017) 5769–5779
17. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014) 2672–2680
18. Seki, A., Pollefeys, M.: Patch based confidence prediction for dense disparity map. In: BMVC. (2016)

19. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A.A.: Generative adversarial networks: An overview. IEEE Signal Processing Magazine **35**(1) (2018) 53–65
20. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. CVPR (2017)
21. Liu, M.Y., Tuzel, O.: Coupled generative adversarial networks. In: Advances in neural information processing systems. (2016) 469–477
22. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
23. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: OSDI. Volume 16. (2016) 265–283
24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)