Supplementary Material for DiffuStereo: High Quality Human Reconstruction via Diffusion-based Stereo Using Sparse Cameras

Ruizhi Shao, Zerong Zheng, Hongwen Zhang, Jingxiang Sun, and Yebin Liu

Tsinghua University, Beijing, China

A Diffusion Model

A.1 Derivation

As presented in Sec. 3.2.2 of the main paper, we adopt a novel diffusion kernel in our diffusion model for faster and more stable disparity refinement. Specifically, given our kernel in Eqn. (5) of the main paper, our diffusion model can be formulated as:

$$q(\mathbf{y}_{1:T}|\mathbf{y}_0) = \prod_{t=1}^{T} q(\mathbf{y}_t|\mathbf{y}_{t-1}), \tag{1}$$

$$q(\mathbf{y}_t|\mathbf{y}_{t-1}) = \mathcal{N}(\mathbf{y}_t|\mathbf{y}_{t-1} - \alpha_t\mathbf{y}_0, \alpha_t\mathbf{I}), \tag{2}$$

$$p_\theta(\mathbf{y}_{0:T}|\mathbf{s}) = p(\mathbf{y}_T)\prod_{t=1}^{T} p_\theta(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{s}). \tag{3}$$

In the following parts, we will give the necessary derivation for the generation of training samples and the reverse process under our new kernel.

Generation of Training Samples. To directly obtain the diffused data $\mathbf{y}_t$ at step t for training, we first derive the distribution of $\mathbf{y}_t$ given $\mathbf{y}_0$ from Eqn. (1) and Eqn. (2):

$$q(\mathbf{y}_t|\mathbf{y}_0) = \mathcal{N}(\mathbf{y}_t|(1-\gamma_t)\mathbf{y}_0, \gamma_t\mathbf{I}), \tag{4}$$

where $\gamma_t = \sum_{i=1}^{t}\alpha_i$. Then the training data $\mathbf{y}_t$ can be sampled from $q(\mathbf{y}_t|\mathbf{y}_0)$ and written as:

$$\mathbf{y}_t = (1-\gamma_t)\mathbf{y}_0 + \sqrt{\gamma_t}\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1}). \tag{5}$$

Reverse Process. In the reverse process, we use the posterior distribution of $q(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{y}_0)$ to represent $p_\theta(\mathbf{y}_{t-1}|\mathbf{y}_t)$. Hence, we need the posterior distribution of $\mathbf{y}_{t-1}$ given $\mathbf{y}_t, \mathbf{y}_0$, which can be written as follows:

$$q(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{y}_0) = \mathcal{N}(\mathbf{y}_{t-1}|\frac{\alpha_t}{\gamma_t}\mathbf{y}_0 + \frac{\gamma_{t-1}}{\gamma_t}\mathbf{y}_t, \frac{\alpha_t\gamma_{t-1}}{\gamma_t}\mathbf{I}). \tag{6}$$

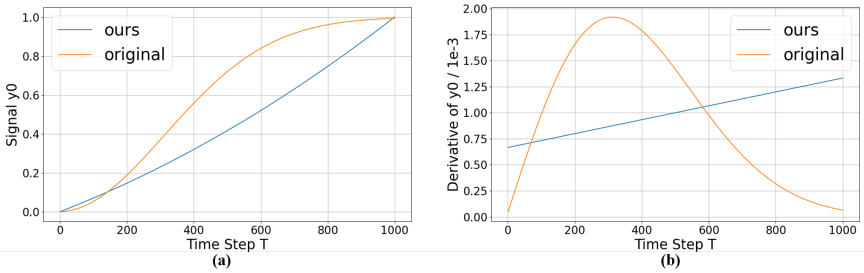

Fig. S1: Diffusion model with different diffusion kernels. (a) The signal $\mathbf{y}_0$ in the reverse process. (b) The derivative of $\mathbf{y}_0$ in the reverse process.

Given the posterior distribution, each step in the reverse process can be further derived as:

$$p_\theta(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{s}) = \mathcal{N}(\mathbf{y}_{t-1}|\mu_\theta(\mathbf{y}_t, \gamma_t, \mathbf{s}), \sigma_t^2\mathbf{I}), \tag{7}$$

$$\mu_\theta(\mathbf{y}_t, \gamma_t, \mathbf{s}) = \frac{\alpha_t}{\gamma_t}\mathbf{y}_0 + \frac{\gamma_{t-1}}{\gamma_t}\mathbf{y}_t, \tag{8}$$

$$\sigma_t^2 = \frac{\alpha_t\gamma_{t-1}}{\gamma_t}. \tag{9}$$

Based on the above derivations, we have the whole reverse process as Eqn.(8) in the main paper, which is written as:

$$\mathbf{y}_{t-1} \leftarrow \frac{\alpha_t}{\gamma_t}\mathbf{y}_0 + \frac{\gamma_{t-1}}{\gamma_t}\mathbf{y}_t + \sqrt{\frac{\alpha_t\gamma_{t-1}}{\gamma_t}}\epsilon_t, \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1}). \tag{10}$$

To supervise the network learning to predict $p_\theta(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{s})$, we have the training objective function as following:

$$L_t = \mathbb{E}_{\mathbf{y}_0,\epsilon,t}\left[\|f_\theta((1-\gamma_t)\mathbf{y}_0 + \gamma_t\epsilon, \mathbf{s}, t) - \mathbf{y}_0\|^2\right]. \tag{11}$$

A.2 Analysis

In the following, we will give additional insight and analysis about our choice of the kernels. In Fig. S1, we show the signal $\mathbf{y}_0$ and its derivative $\frac{d\mathbf{y}_0}{dt}$ under different kernels in the whole reverse process. It can be seen that the derivative of $\mathbf{y}_0$ of the original diffusion kernel is too small at the early stage, which makes it difficult for the diffusion model to recover $\mathbf{y}_0$. We think such property is helpful for generative tasks as it encourages the network to learn the diversity of data distribution. However, this property also strengthens the uncertainty and makes the flow estimation unstable at the beginning. As shown in Fig. S1, our diffusion kernel can preserve the the derivative of $\mathbf{y}_0$ in the whole process, which encourages the network to gradually recover $\mathbf{y}_0$ in a more stable manner.

B Implementation Details

Network Structure. We adopt the same U-Net [5] model as the architecture of the global network and the diffusion network. Our U-Net architecture follows the backbone of Guided-Diffusion [1], which is a U-Net [5] model based on a Wide ResNet [7]. The U-Net model contains 5 downsampling and upsampling layers. Each layer has 3 standard residual blocks and a downsampling/upsampling block. The downsampling block uses 2D convolution to downsample the image feature. The upsampling block first adopts the nearest interpolation to upsample the image feature and then applies 2D convolution. We set 32 as the number of channels of the first layer. Then the numbers of channels are multiplied by (1, 2, 4, 8, 8) in 5 downsampling layers and (8, 8, 4, 2, 1) in upsampling layers, respectively. We adopt the Group Norm [6] with size of 16 as the normalization layer in each residual block.

Diffusion Schedule. As mentioned in Sec. 4.1 of the main paper, we use a 30-step diffusion schedule in our method. In our implementation, we design a linear interpolation from $[1/45, 2/45]$ for α_t: $\alpha_t = \frac{1}{45} + \frac{t}{45T}$, where $T = 30$ is the number of diffusion steps. Our linear schedule constrains the scale of $\mathbf{y}_0$ in $[0, 1]$.

Patch-based Training. In Sec. 3.2.4 of the main paper, we present a multi-level network structure to deal with ultra-high resolution (up to 4K) images. Such a multi-level structure allows us to train the network in a patch-based manner. Specifically, at the global level, we downsample the 4K images and the ground truth flow maps to 512×512 as the training data. At the diffusion level, we randomly crop a rectangle region with a resolution of 1024×1024 from the high-resolution images as the patch training sample for diffusion-based stereo learning.

Training Details. In training process, we set the batch size to 2 and adopt Adam [3] from PyTorch as our optimizer. In the first 200k iterations, we train the network with the learning rate of 1e-4. Then we lower the learning rate to 1e-5 for another 200k iterations.

Light-weight Multi-view Fusion. In light-weight multi-view fusion (Sec. 3.3 of the main paper), we set λ_d to 1 and λ_s to 10. We adopt an Adam [3] optimizer to optimize the refined depth point cloud $\widetilde{\mathbf{p}}^i$. The learning rate is set to 1e-3 and the total iterations is 500. The final model is reconstructed by applying Poisson reconstruction [2] on the union set of $\widetilde{\mathbf{p}}^i$ and the coarse point cloud $\mathbf{p}^c$.

Mask Computation for Training Samples. As mentioned in the Experiment section of the main paper, we remove the occlusion regions with bad depth initialization for more stable learning and meaningful evaluation. Given the coarse model, the ground-truth model and a reference view n, we remove those pixels in view m that are not visible in view n, with the visibility determined by z-buffering both the coarse model and the ground-truth model. We also remove pixels in regions with large depth errors ($\geq$ 2cm) between the ground truth depth map and the coarse depth map.

Table S1: Additional ablation study on the THUman2.0 dataset.

Method	AvgErr			1/2pix (%)			1pix (%)			3pix (%)		
	20°	30°	45°	20°	30°	45°	20°	30°	45°	20°	30°	45°
Our method (w./o. $\mathbf{D}_c$)	11.88	14.28	17.35	-	-	-	-	-	-	-	-	-
Global level	1.572	1.578	2.358	27.9	23.0	18.3	50.1	44.3	34.9	87.1	86.7	74.2
Iterative model (w. Noise)	0.553	0.683	0.899	67.6	61.6	52.1	87.3	83.6	77.4	97.9	96.8	94.8
Our method (w./o. Global)	0.507	0.669	1.384	76.5	67.1	43.2	91.0	85.4	65.9	97.7	96.5	89.1
Our method (10 Steps)	0.518	0.569	0.674	69.8	66.3	64.2	88.9	87.5	82.6	97.6	96.4	95.3
Our method (50 Steps)	**0.462**	**0.491**	**0.606**	**72.5**	**71.7**	**68.4**	**91.5**	**90.9**	**86.3**	**99.1**	**98.3**	**97.9**
Our method (30 Steps)	0.483	0.515	0.632	71.3	70.0	67.6	90.6	89.0	85.9	98.6	97.9	96.6

Table S2: Additional quantitative human geometry reconstruction results. Both our diffusion-based stereo and the Light-weight Multi-view Fusion (LMF) contribute to higher reconstruction quality.

Method	THuman2.0 (8 views)				
	Chamfer	P2S	1mm(%)	2mm(%)	5mm(%)
Raft-Stereo [4] (w. LMF)	1.359	1.434	41.6	84.3	93.7
Our Method (w./o. LMF)	1.266	1.321	63.4	91.4	95.4
Our Method (w. LMF)	**1.198**	**1.258**	**68.1**	**91.9**	**96.6**

C Additional Ablation Study

In the main paper, we have included ablation experiment results of the diffusion stereo network, which is our core contribution. In this section, we provide additional ablation studies and analyses for the whole reconstruction system. Specifically, we evaluate the performances of the following ablation methods and compare their results in Tab. S1:

1) Our method (w./o. $\mathbf{D}_c$): Our method without the coarse depth map as initial value;

2) Global level: The global network in our method;

3) Iterative model (w. Noise): remove the diffusion model in our method and add noise to the input;

4) Our method (w./o. Global): our method with only the diffusion level;

5) Our method (10 Steps): Our method with 10 diffusion iteration steps;

6) Our method (50 Steps): Our method with 50 diffusion iteration steps;

7) Our method (30 Steps): Our method with 30 diffusion iteration steps. Our final solution uses this scheme.

C.1 Multi-level Network Structure

As shown in Tab. S1, the performance of our method is degraded if we remove the global network from our diffusion network (see Our method (w./o. Global)). Despite the degraded results, our method still outperforms Raft-Stereo [4] without the guidance of the global network. In Tab. S1, we also report the results of the global network (see Global level). The performance of the global level is much worse than that of the diffusion level since it predicts the flow in the low-resolution domain with only one iteration.

Fig. S2: Qualitative comparisons with RAFT-Stereo (w. LMF).

C.2 Iterative Model with Noise

In the main paper, we evaluate our method without the diffusion model in Tab. 1. The stereo process without diffusion can be regarded as an iterative model with 5 iterations similar to RAFT [4]. To further validate the efficiency of adding noises for sub-pixel continuous flow estimation, we directly add noise to the initial coarse flow and evaluate our method without diffusion. As shown in Tab. S1, adding noise can indeed improve the performance of the iterative model (see Iterative model (w. Noise)), but its performance is not as good as our method with diffusion.

C.3 Quantitative Results of Different Steps

In the main paper, we qualitatively ablate different diffusion steps in Fig. 7. To comprehensively evaluate our design, we report the quantitative results of our method with different diffusion steps in Tab. S1, where we can see that more diffusion steps lead to higher accuracy. As more diffusion steps take higher time cost (500ms per iteration for single 4K image), our final solution uses 30 steps and we found that its performance is enough for high quality reconstruction.

C.4 Light-weight Multi-view Fusion

We also ablate our light-weight multi-view fusion by replacing it with a straight-forward fusion. In the straight-forward fusion, the human model is reconstructed directly from the refined depth point cloud. As shown in Tab. S2, the performance of such a straight-forward fusion is worse than our method since it neglect the occlusion regions and depth edges with large errors.

C.5 Different coarse model as initial value

In the main paper, we qualitatively evaluate our diffusion network with different initial coarse models. In Tab.S3, we further quantitatively compare the performance of our methods with different initial values provided by Visual Hull,

Method	THuman2.0 (8 views)				
	Chamfer	P2S	1mm(%)	2mm(%)	5mm(%)
DoubleField	2.879	2.389	23.2	61.6	90.8
Ours(Visual hull)	1.767	1.982	31.2	70.9	88.1
Ours(PIFuHD)	1.346	1.455	61.6	90.2	93.5
Ours(PIFu)	1.246	1.402	64.7	91.4	94.7
Ours(DbField 128^3)	1.319	1.324	63.9	91.7	95.8
Ours(DbField 256^3)	1.254	1.264	65.4	91.3	96.1
Our Method	**1.198**	**1.258**	**68.1**	**91.9**	**96.6**

Table S3: Quantitative results given different initial values. We report result of DoubleField in the first row as a reference.

PIFu, PIFuHD, and DoubleField. We also downsample the geometry results reconstructed by DoubleField (default 512^3) to 128^3 and 256^3 for additional comparisons. In this experiment, we use the same trained diffusion model, i.e., trained with default DoubleField initial values and the ground truth, for evaluation of different initial values without re-training. Our method can still recover high-quality geometry given the initial values even as coarse as Visual Hull, which proves the robustness and generalization capability.

D Additional Result

D.1 Human Reconstruction by Raft-Stereo

We report additional results reconstructed by Raft-Stereo [4] in Fig. S2 and Tab. S2. Here, we first adopt Raft-Stereo [4] to estimate the refined depth maps and then use our fusion method to reconstruct human. These results further validate the efficacy of our diffusion-based stereo for high-quality depth refinement.

References

1. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. NeurIPS **34** (2021)
2. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: ESGP. vol. 7 (2006)
3. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (2014)
4. Lipson, L., Teed, Z., Deng, J.: Raft-stereo: Multilevel recurrent field transforms for stereo matching. In: 3DV. pp. 218–227 (2021)
5. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241. Springer (2015)
6. Wu, Y., He, K.: Group normalization. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
7. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: Richard C. Wilson, E.R.H., Smith, W.A.P. (eds.) Proceedings of the British Machine Vision Conference (BMVC). pp. 87.1–87.12. BMVA Press (September 2016). https://doi.org/10.5244/C.30.87, https://dx.doi.org/10.5244/C.30.87