OPTIMIZED COLOR-GUIDED FILTER FOR DEPTH IMAGE DENOISING

Mostafa M. Ibrahim ⋆†, *Qiong Liu* ⋆

⋆ School of Electronic Information and Communication
Huazhong University of Science and Technology, Wuhan, China
† Department of Electrical Engineering, Minia University, Minia, Egypt

ABSTRACT

Color Guided Depth image denoising often suffers from the texture coping from the color image as well as the blurry effect at the depth discontinuities. Motivated by this, we propose an optimized color-guided filter for depth image denoising from different types of noises. This is a new framework that helps to mitigate the texture coping and enhance the depth discontinuities, especially in heavy noises. This framework consists of two parts namely *depth driven color flattening model* and *patch synthesis-based Markov random field model*. The first part which is a prepare step for the second part is used to mitigate the texture coping problem that faces all color guided methods. This first model consists of a modified joint bilateral filter which is used to mitigate the noise from the noisy depth image and an iterative guided bilateral filter that is proposed to flatten the colors in the color image for mitigating the texture coping problem. Based on the first part, Markov random field with an optimization technique is used for mitigating the blurry effect. Experiments indicate that our method outperforms counterpart filters with guided and non-guided manners in terms of a variety of evaluation metrics.

Index Terms— Depth image, bilateral filter, MRF, depth sensors

1. INTRODUCTION

Depth images are crucial in a variety of visual communication and computer vision applications such as 3DTV broadcasting [1], 3D reconstruction [2] and visual saliency [3]. These depth images are often corrupted by a variety of noises that degrade their quality which are necessary to support their applications. These noises appear through the acquisition or transmission step such as Gaussian, speckle, and Salt and Pepper (S&P) noises. Therefore, depth images denoising is an essential task. To the best of our knowledge, the depth image denoising research can be divided into two categories:

1) **Depth image denoising without auxiliaries**: In this category, denoising process is performed on the depth image alone without other auxiliary. Typical local methods include Bilateral Filter (BF) [4] and self Guided Filter (GF) [5]. Non-local methods depend on non-local similarity in the depth image such as Non-Local Means (NLM) method [6] and its variants [7, 8]. This kind of methods face the challenge of detecting depth discontinuities and filling the hole pixels.

2) **Depth image denoising guided by auxiliaries**: If priors can be involved for guiding, the denoising performance of depth image can be improved. In this category, methods can be divided according to where priors come from, such as depth images from neighboring views (i.e. multi-view depth) or corresponding color images. Multi-view depth is now a typical data representation adopted by MPEG, and depth image denoising models guided by neighboring views were proposed to process the quantization caused by compression [9, 10]. For these methods, a new challenge of misalignment arises among different viewpoint depth images. To avoid that, color image guided models were proposed because correspondence exists between color and depth images. Among these methods, including Joint Bilateral Filter (JBF) [11], Noise-Aware Filter for Depth Up-sampling (NAFDU) [12] and Markov Random Field (MRF) based model [13], color image is helpful in recognizing the depth discontinuities in the noisy environment. However, fake texture copying problem arises from this guide where textures should be found only in color instead of depth image. Moreover, all these filter-based methods often suffer from the blurry effect that appears along depth discontinuities [14]. Recently, deep learning based methods with using color image as a guide were also proposed [15, 16, 17]. How can we completely eradicate this texture coping problem becomes a new challenge in the community of depth image denoising. Motivated by this, we propose a new framework that also depends on the color image guiding to denoise the corrupted depth image in heavy noisy environment with mitigating the texture coping and blurry effect problems. There are two models involved in this framework, namely Depth Driven Color Flattening (DDCF) model and patch synthesis-based MRF model. In our method, the filter-based and MRF models are used to highly mitigating the texture coping and blurry effect problems respectively and finally, depth image is obtained with little texture and blur. For evaluation, our method

This work was partly supported by National Key R&D Program (Grant No. 2017YFC0806202), and National Natural Science Foundation of China (Grant No. 91848107 and 61571204).

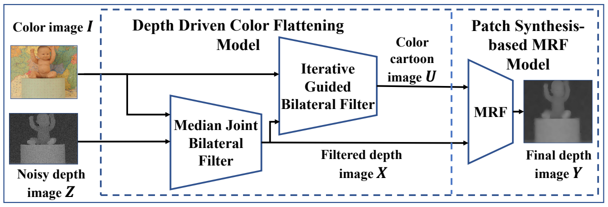

Fig. 1. Overview of the proposed framework.

is compared with guided and non-guided algorithms, and the results indicate the effectiveness of our method.

2. THE PROPOSED FRAMEWORK

As shown in Fig. 1, our method consists of two models; DDCF and MRF. The first model consists of Median Joint Bilateral Filter (MJBF) and Iterative Guided Bilateral Filter (IGBF). Patch synthesis-based MRF is useful for overcoming the blurry effect that appears at the depth discontinuities compared with all filter-based methods, so it is used in our framework. However, MRF also suffers from the texture coping, so DDCF is proposed as a prepare step for overcoming what MRF model faces, especially at heavy noises. So, our framework has the benefits of both modified filter-based and MRF models for overcoming the texture coping and blurry effect. Our method works as follows: firstly, the noisy depth image Z guided by the co-aligned color image I are applied to MJBF which results the filtered depth image X. MJBF is used to mitigate the noise and help our MRF model to work better. Then, guided with the output of MJBF, the color image is filtered iteratively by IGBF to flatten the colors in the color image regions corresponding to the homogeneous regions in the depth image. In other words, IGBF is used to smooth the non-depth edges in the color image by converting it to an image U like a cartoon. So, IGBF is useful for overcoming the texture coping. At last, these outputs from the two filters, which include X and U are taken for defining the data and the smoothness terms of our MRF model respectively to get the final depth image Y.

2.1. Depth Driven Color Flattening Model

Median Joint Bilateral Filter: Traditional color guided filters fail to denoise the depth image from S&P noise in addition to the hole pixels in the captured depth image. The best solution for dealing with the salt and pepper noise and the hole pixels is the median filter [18]. Motivated by the benefit of the median filter for solving this problem, we propose MJBF, where in which if the polluted depth pixels are a pure black or a pure white, we take the median value of this polluted pixel and its local neighborhood pixels in a w_1 window size. Note that the values of the pure black and the pure white pixels are 0 and 1 respectively, where the noisy depth image is normalized in the range $\{0,1\}$. In contrast, if the polluted pixel is neither a pure black nor a pure white, we apply the traditional JBF on the noisy depth image as follows:

$$X_p = \begin{cases} \text{Median } [Z_q, Z_p] & Z_p \in \{0,1\}, q \in \Omega_p \\ \frac{1}{k_p} \sum_{q \in \Omega_p} Z_q f_s(|p-q|) f_r(\|I_{p_c} - I_{q_c}\|) & \text{Otherwise} \end{cases} \tag{1}$$

where k_p is the a normalizing factor given by

$$k_p = \sum_{q \in \Omega_p} f_s(|p-q|) f_r(\|I_{p_c} - I_{q_c}\|) \tag{2}$$

where f_s is the spatial domain kernel of the noisy depth image and f_r is the range domain kernel. All of these kernels are Gaussian kernels where σ_{s1} and σ_{r1} are the parameters controlling the fall off of weights in f_s and f_r respectively. p is the center pixel in the kernel window wanted to be denoised, q is every pixel in the local neighborhood area Ω_p for pixel p, and Z_q is the depth value of the noisy depth image at pixel q. I_{p_c} and I_{q_c} are the intensity values of the color image at p_c and q_c, where p_c and q_c are the corresponding locations to p and q in the noisy depth image.

Iterative Guided Bilateral Filter: The texture copying is caused by the fact that algorithms can hardly recognize the texture of object contour or details. Actually, object contour is needed because it can be also depicted in the depth image, while object details can not. To solve this problem, IGBF is used to convert the color image to a cartoon-like image by flattening the colors in the color image. This filter is guided by the depth image resulting from MJBF and also has an iterative manner with n iterations. The traditional iterative property of BF (IBF) is good for mitigating the texture coping but in the other hand causes smoothing for some of the depth edges, especially when two objects at different sides of these depth edges have a similar color. To solve this, we guide IBF by the output of MJBF for enhancing the weak edges in the color image that co-align the depth image edges. Moreover, the guided property drives the colors flattening operation to be happened in the corresponding regions to the depth smooth regions. The proposed IGBF is specified by:

$$U_{p_c} = \frac{1}{k_c} \sum_{q_c \in \Omega_{p_c}} I_{q_c} f_s(|p_c - q_c|) f_{r1}(\|I_{p_c} - I_{q_c}\|) f_{r2}(\|X_p - X_q\|) \tag{3}$$

where k_c is the a normalizing factor and given by:

$$k_c = \sum_{q_c \in \Omega_{p_c}} f_s(|p_c - q_c|) f_{r1}(\|I_{p_c} - I_{q_c}\|) f_{r2}(\|X_p - X_q\|)$$

where f_{r1} and f_{r2} are the range domain kernels and f_s is the spatial domain kernel of the filtered depth image X. All of these kernels are Gaussian kernels where σ_{s2} , σ_{rc} and σ_{rd} are parameters controlling the fall off of weights in f_s , f_{r1} and f_{r2} respectively and w_2 is the window size. p_c is the center pixel in the kernel window of the color image wanted to

Table 1. Parameters for our proposed framework

Filter	MJBF			MRF	IGBF				
Parameters	w_1	σ_{s1}	σ_{r1}	c	w_2	σ_{s2}	σ_{rc}	σ_{rd}	n
Value	9	0.1	0.1	0.05	3	9	0.05	0.01	10

be denoised, q_c is every pixel in the neighborhood area Ω_{p_c} for pixel p_c, and I_{q_c} is the intensity value of the color image at pixel q_c. X_p and X_q are the depth values for the corresponding locations of p_c and q_c in the filtered depth image X.

2.2. MRF Model

In our MRF model, MRF model [13] is modified to fit our specific purpose. Based on the outputs of DDCF model, we define MRF through the two potentials; the data term which is the depth measurement potential and specified as:

$$E_d = \sum_{i=1}^{P} w_d (Y_i - X_i)^2 \tag{4}$$

where w_d is the weight for the data term, i is the index for all pixels P in the filtered depth image X and the corresponding pixels of the final depth image Y resulting from the MRF model. The second potential (the smoothness term) is specified as:

$$E_s = \sum_{p} \sum_{q \in \Omega_p} w_{pq} (Y_p - Y_q)^2 \tag{5}$$

where Ω_p is the pixels set in the final depth image Y that correspond to p's neighborhood, Y_p and Y_q are the final depth values at p and q respectively and w_{pq} is the relative weight between pixel p and every pixel q in Ω_p as specified by:

$$w_{\text{pq}} = \exp(-c\|U_{p_c} - U_{q_c}\|_2^2) \tag{6}$$

where c is a parameter whose value is used to govern the smoothing degree in the final depth image Y and $\|U_{p_c} - U_{q_c}\|_2^2$ is the square Euclidean distance between the intensity values of the corresponding patches of the color cartoon image U at p_c and q_c as follows. To get the final denoised depth image, we compute the posterior mode of MRF instead of the full posterior because the full one requires a lot of time for convergence [13]. As the processing time is important for real time applications, we use the fast CG algorithm [19].

3. EXPERIMENTAL RESULTS

In these experiments, we take 23 test depth images from Middlebury [20] [21] and add to them four different types of noise individually with the following parameters: For Gauss., we set the mean $\mu = 0$, and the variance $\sigma^2 = 0.01$. For Gaussian with local variance (GaussLV), $\mu = 0$ and the variance is different from pixel to pixel $\sigma^2 = rand(M, N)$ where M and N are the number of rows and columns of

Table 2. ΔPSNR of all methods for all types of noise

Num	DataSets	Metric	Proposed	JBF	NAFDU	CGF	AMF	MRF
1	Cones	Gauss.	**11.2447**	9.9874	10.7635	10.3265	10.784	6.7355
		GaussLV	**16.8714**	12.8475	15.7775	13.8223	4.7025	0.088133
		S&P	**13.3466**	10.9631	12.4756	11.9451	1.6207	2.0953
		Speckle	**12.8644**	11.2059	12.6715	12.248	8.4912	7.1632
2	Teddy	Gauss.	**13.0207**	11.1045	12.2192	10.9688	12.3747	5.6667
		GaussLV	**16.5629**	14.0279	12.0647	13.8898	3.3097	-1.3498
		S&P	**14.7863**	11.7478	14.2985	11.5511	2.3791	-0.89773
		Speckle	**12.9469**	10.6765	12.0501	10.5403	11.3095	6.918
3	Venus	Gauss.	**20.5211**	19.4429	15.6575	18.9865	16.7185	2.2614
		GaussLV	**12.3631**	10.4651	2.6188	10.2522	1.7005	-3.4657
		S&P	**17.9616**	17.286	15.5366	16.8834	0.9217	-8.036
		Speckle	**17.9168**	17.5197	15.7881	17.5263	14.21	6.9234
4	Bowling1	Gauss.	7.2312	9.8373	9.743	11.2614	**15.2662**	1.1785
		GaussLV	**12.6345**	10.9564	4.9705	10.9085	1.5162	-1.3428
		S&P	9.013	11.388	**12.7988**	12.5831	1.4595	-8.0768
		Speckle	8.3596	11.2633	**13.3716**	12.5401	9.2521	2.6648
5	Plastic	Gauss.	**19.5021**	19.2649	19.1111	19.0635	17.4792	6.987
		GaussLV	**18.1218**	14.3022	12.6075	15.07	1.7063	0.34272
		S&P	**18.8507**	18.4371	18.2497	18.6701	2.2929	1.2287
		Speckle	**20.9996**	20.6817	20.3839	20.5466	9.6243	7.1299

Table 3. Comparison of different denoising algorithms gains

Num	DataSets	Metric	Proposed	JBF	NAFDU	CGF	AMF	MRF
1	Cones	ΔRMSE	**-37.349**	-34.169	-31.562	-34.937	-15.493	-9.600
		ΔSSIM	**0.716**	0.683	0.588	0.637	0.290	0.528
		ΔPE	**-8.655**	-4.962	-3.232	-2.865	-5.356	22.040
		ΔPSNR	**13.539**	11.251	12.922	12.085	6.400	3.933
2	Teddy	ΔRMSE	**-36.635**	-34.211	-28.226	-34.040	-14.997	-1.748
		ΔSSIM	**0.759**	0.737	0.639	0.664	0.345	0.576
		ΔPE	**-15.628**	-11.695	-2.740	-8.184	-9.270	24.331
		ΔPSNR	**14.123**	11.889	12.658	11.738	7.344	2.547
3	Venus	ΔRMSE	**-35.051**	-33.492	-21.882	-33.144	-14.001	20.525
		ΔSSIM	**0.849**	0.812	0.692	0.720	0.401	0.520
		ΔPE	**-20.389**	-14.870	-2.334	-12.288	-20.335	25.179
		ΔPSNR	**17.082**	16.178	12.400	15.912	8.388	-0.659
4	Bowling1	ΔRMSE	-32.035	-32.705	-26.525	**-33.537**	-14.894	13.391
		ΔSSIM	**0.760**	0.750	0.725	0.659	0.320	0.659
		ΔPE	**-10.230**	-7.094	-4.189	-3.868	-9.121	23.920
		ΔPSNR	9.5389	10.861	10.221	**11.823**	6.874	-1.404
5	Plastic	ΔRMSE	**-40.876**	-39.133	-34.480	-39.034	-16.872	-10.029
		ΔSSIM	**0.816**	0.779	0.774	0.703	0.339	0.750
		ΔPE	**-25.319**	-17.791	-11.201	-12.362	-10.360	25.408
		ΔPSNR	**19.346**	18.171	17.588	18.338	7.776	4.019
6	Aloe	ΔRMSE	**-30.652**	-20.167	-28.961	-29.809	-11.138	17.064
		ΔSSIM	**0.656**	0.506	0.624	0.570	0.340	0.462
		ΔPE	-3.435	3.391	-3.604	-0.414	**-12.429**	25.796
		ΔPSNR	9.574	8.945	8.877	**9.715**	5.794	-1.174
7	Art	ΔRMSE	**-37.851**	-32.503	-36.298	-36.701	-15.268	-9.418
		ΔSSIM	**0.417**	0.381	0.395	0.380	0.183	0.291
		ΔPE	**-15.531**	-8.054	-11.778	-7.380	-10.255	24.121
		ΔPSNR	14.127	14.215	13.489	**14.251**	6.799	3.822
8	Baby1	ΔRMSE	**-34.760**	-23.977	-33.492	-33.970	-13.180	10.445
		ΔSSIM	**0.816**	0.705	0.793	0.715	0.385	0.662
		ΔPE	**-16.612**	-4.665	-13.926	-13.150	-16.263	26.062
		ΔPSNR	13.677	12.147	13.234	**14.462**	7.571	0.371
9	Baby2	ΔRMSE	**-35.748**	-24.975	-33.800	-34.177	-14.151	9.873
		ΔSSIM	**0.817**	0.717	0.791	0.713	0.380	0.662
		ΔPE	**-19.974**	-5.233	-15.648	-12.668	-14.277	26.212
		ΔPSNR	**15.059**	13.555	13.376	14.402	8.112	0.659
10	Bowling2	ΔRMSE	**-36.820**	-32.772	-33.697	-34.119	-15.619	-7.689
		ΔSSIM	**0.740**	0.693	0.725	0.630	0.280	0.674
		ΔPE	**-9.420**	-7.654	-8.119	-3.496	-9.042	24.040
		ΔPSNR	12.649	**13.255**	10.436	10.835	6.678	2.864
11	Cloth1	ΔRMSE	**-39.930**	-30.339	-39.105	-38.508	-16.249	-4.723
		ΔSSIM	**0.893**	0.680	0.864	0.774	0.398	0.787
		ΔPE	**-23.806**	0.445	-19.098	-12.909	-12.255	25.649
		ΔPSNR	**20.351**	14.717	19.641	18.406	9.490	3.273
12	Cloth2	ΔRMSE	**-41.268**	-37.188	-37.844	-37.294	-16.260	-19.310
		ΔSSIM	**0.839**	0.716	0.817	0.719	0.283	0.800
		ΔPE	**-10.556**	-4.299	-7.286	-3.031	-2.921	21.911
		ΔPSNR	**17.804**	15.983	15.000	14.216	6.951	6.377
13	Cloth3	ΔRMSE	**-38.924**	-28.993	-37.152	-36.942	-15.486	-2.339
		ΔSSIM	**0.841**	0.682	0.814	0.741	0.357	0.732
		ΔPE	**-14.320**	-2.260	-12.091	-8.309	-9.883	25.581
		ΔPSNR	**18.138**	14.239	16.166	16.123	8.018	2.831
14	Cloth4	ΔRMSE	**-37.309**	-32.742	-35.703	-36.134	-16.929	-8.665
		ΔSSIM	**0.818**	0.656	0.792	0.702	0.282	0.758
		ΔPE	**-11.300**	-1.265	-9.683	-4.890	-7.703	23.861
		ΔPSNR	**13.089**	12.413	12.291	12.811	7.409	2.860
15	Lampshade1	ΔRMSE	-34.987	-27.675	-33.641	**-35.221**	-14.669	4.595
		ΔSSIM	**0.770**	0.729	0.740	0.681	0.327	0.644
		ΔPE	-9.917	-7.449	-9.681	-9.153	**-9.955**	24.830
		ΔPSNR	12.197	12.709	11.767	**14.098**	6.820	0.833
16	Lampshade2	ΔRMSE	**-36.603**	-29.921	-35.162	-36.451	-15.606	-1.114
		ΔSSIM	**0.810**	0.779	0.780	0.697	0.328	0.713
		ΔPE	**-14.493**	-7.968	-12.685	-9.771	-10.169	24.964
		ΔPSNR	13.516	14.257	12.832	**14.760**	7.166	1.939
17	Midd1	ΔRMSE	**-35.955**	-26.901	-34.318	-34.591	-15.230	4.961
		ΔSSIM	**0.731**	0.665	0.714	0.638	0.324	0.595
		ΔPE	-10.208	-5.466	-10.071	-7.851	**-11.236**	25.069
		ΔPSNR	**13.842**	12.789	12.888	13.438	7.200	1.134
18	Midd2	ΔRMSE	**-33.924**	-23.789	-32.692	-33.148	-14.041	18.245
		ΔSSIM	**0.699**	0.636	0.686	0.617	0.319	0.521
		ΔPE	-9.028	-0.959	-8.438	-7.389	**-13.756**	25.179
		ΔPSNR	12.741	11.967	12.595	**13.387**	6.937	-1.017
19	Monopoly	ΔRMSE	-30.609	-20.566	-30.309	**-31.827**	-13.008	32.964
		ΔSSIM	**0.696**	0.612	0.675	0.641	0.350	0.476
		ΔPE	-5.332	-0.074	-5.216	-4.684	**-14.208**	25.098
		ΔPSNR	10.483	10.443	10.744	**12.940**	7.904	-3.356
20	Rocks1	ΔRMSE	**-38.790**	-30.071	-36.544	-36.125	-15.316	-4.621
		ΔSSIM	**0.781**	0.676	0.766	0.672	0.345	0.693
		ΔPE	**-19.535**	-4.250	-14.519	-8.229	-9.875	25.742
		ΔPSNR	**17.102**	14.573	14.240	13.801	8.217	3.359
21	Rocks2	ΔRMSE	**-38.852**	-30.730	-36.381	-35.946	-16.702	-7.534
		ΔSSIM	**0.782**	0.671	0.763	0.665	0.413	0.705
		ΔPE	**-21.707**	-3.538	-16.760	-10.017	-10.590	25.804
		ΔPSNR	**16.867**	14.659	13.436	13.210	9.282	4.225
22	Wood1	ΔRMSE	**-38.708**	-31.243	-37.028	-37.093	-15.110	-7.027
		ΔSSIM	**0.867**	0.807	0.843	0.713	0.351	0.788
		ΔPE	**-22.057**	-7.751	-16.755	-9.424	-12.150	25.056
		ΔPSNR	**16.085**	14.344	14.878	15.096	7.699	3.602
23	Wood2	ΔRMSE	**-39.829**	-34.948	-37.024	-37.309	-16.509	-10.460
		ΔSSIM	**0.853**	0.804	0.820	0.702	0.303	0.785
		ΔPE	**-17.501**	-9.428	-12.005	-6.505	-10.641	23.364
		ΔPSNR	**16.214**	15.475	14.160	14.734	7.426	3.572
	Average	ΔRMSE-Avg	**-36.656**	-28.79	-34.907	-35.22	-15.075	1.213
		ΔSSIM-Avg	**0.77**	0.675	0.746	0.668	0.332	0.651
		ΔPE-Avg	**-14.283**	-4.355	-11.469	-7.776	-10.959	24.874
		ΔPSNR-Avg	**14.665**	13.325	13.409	13.938	7.489	1.945

the depth image respectively. For S&P, the noise density $d = 0.05$. For speckle, we set $\mu = 0$ and $\sigma^2 = 0.04$. All of these parameters values are put in $imnoise$ function in MATLAB. The parameters setting for our method is shown in Table 1. Moreover, our method is compared with five algorithms, which include MRF [13], Adaptive Manifold Filter (AMF) [22], JBF [11], NAFDU [12], and Color Guided Filter (CGF) [5]. We evaluate the performance of these methods in terms of Root Mean Square Error (RMSE), Structure Similarity (SSIM) [23], Percent of Error (PE) [24] and Peak Signal to Noise Ratio (PSNR). Table 2 shows the comparison between our method and the other methods in the case of every noise in terms of ΔPSNR. For fair comparison, our method performance is also compared with the other methods in terms of all evaluation metrics as shown in Table 3 where all values represent the gains of every method. Note that every gain value is actually the average value of the summation of all of the gains for the four types of noises. For example in case of PSNR, the final ΔPSNR for every method is calculated as follows:

$$\Delta PSNR = \frac{A_{gain} + B_{gain} + C_{gain} + D_{gain}}{4} \qquad (7)$$

where $A_{gain} = PSNR_{Af_{(Gauss)}} - PSNR_{Bf_{(Gauss)}}$,
$B_{gain} = PSNR_{Af_{(GaussLV)}} - PSNR_{Bf_{(GaussLV)}}$,
$C_{gain} = PSNR_{Af_{(S\&P)}} - PSNR_{Bf_{(S\&P)}}$, and
$D_{gain} = PSNR_{Af_{(Speckle)}} - PSNR_{Bf_{(Speckle)}}$.

$PSNR_{Bf}$ is the PSNR of the noisy depth image Z before the denoising process and $PSNR_{Af}$ is the PSNR of the final depth image Y after the denoising process. This calculation is also valid for the other evaluation metrics. The gains of RMSE and PE are by negative but those of SSIM and PSNR are by positive. From Table 3, it is clearly seen that our method outperforms the other algorithms and has the first rank in terms of two evaluation metrics (ΔRMSE and ΔSSIM) except for three datasets. In the final row of Table 3, we also calculate the average gains of all evaluation metrics over all test images. Therefore, we observe that our method outperforms all of the other algorithms in terms of all these evaluation metrics averages.

In addition, the visual quality of our method is compared with those of other algorithms as shown in Fig 2 in the case of the Gaussian noise. Our method outperforms other color guided methods, especially in mitigating the texture coping in the heavy noise as well as hole pixels filling and S&P noise removal. It also keeps the depth image structure and almost depth discontinuities. Moreover, it has the benefit of the optimization technique where it gets the best performance related to the prior model. As a counterpart, JBF suffers from the blurry effect at the depth edges as well as coping some of the texture to the final depth image. This blurry effect appears because the intensity differences in some regions of the color image do not appear significantly. In our method, this problem effect is highly decreased because IGBF has the guidance property, which in turn enhances the depth image edges. We also see that NAFDU is robust for preserving the depth discontinuities but a lot of texture still clearly appears. This occurs because NAFDU switches to JBF in some heavy noise in the smooth regions. As a result, our method is better than these filter-based methods as well as MRF alone in mitigating the texture coping as appears from the visual quality. Finally, Fig 3 compares between the boundary performance of our method and CGF for every noise type. However, our method does not totally prevent coping some of texture because the color cartoon result is limited by the contrast between the intensities in the color image.

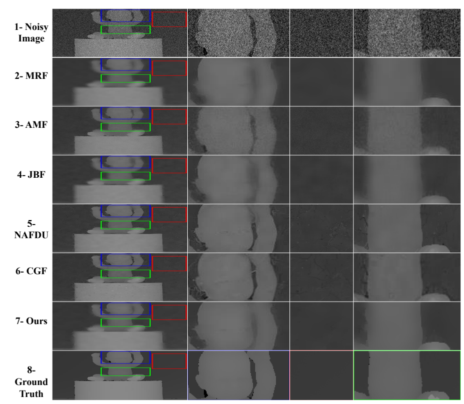

Fig. 2. Different denoising algorithms for *Baby1* corrupted with Gaussian noise and its image fragments.

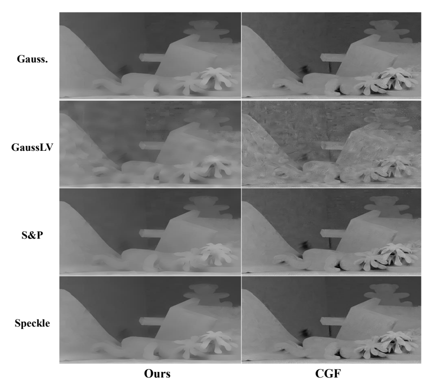

Fig. 3. Our proposed method and CGF in different noise.

4. CONCLUSION

In this paper, a new method via depth driven color flattening filter and MRF is proposed for depth images denoising. Our method can filter different types of noise with mitigating the blurry effect and the texture coping problems. Experimental results show that our method outperforms other denoising algorithms in terms of all metrics where has 14.7 dB and 14.3 % average gains on PSNR and PE respectively.

5. REFERENCES

[1] Mansi Sharma, Santanu Chaudhury, and Brejesh Lall, "Kinect-variety fusion: A novel hybrid approach for artifacts-free 3dtv content generation," in *Proc. Int. Conf. Pattern Recognit (ICPR)*. IEEE, 2014, pp. 2275–2280.

[2] Rui Fan, Xiao Ai, and Naim Dahnoun, "Road surface 3d reconstruction based on dense subpixel disparity map estimation," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3025–3035, 2018.

[3] You Yang, Bei Li, Pian Li, and Qiong Liu, "A two-stage clustering based 3d visual saliency model for dynamic scenarios," *IEEE Trans. Multimedia*, 2018.

[4] Carlo Tomasi and Roberto Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th IEEE Int. Conf. Comput. Vis. (ICCV)*. IEEE, 1998, pp. 839–846.

[5] Kaiming He, Jian Sun, and Xiaoou Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, 2013.

[6] Antoni Buades, Bartomeu Coll, and J-M Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, 2005, vol. 2, pp. 60–65.

[7] Lu Lu, Weiqi Jin, and Xia Wang, "Non-local means image denoising with a soft threshold," *IEEE Signal Process. Lett.*, vol. 22, no. 7, pp. 833–837, 2015.

[8] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.

[9] Zhen Liu, Qiong Liu, You Yang, Yuchi Liu, Gangyi Jiang, and Mei Yu, "Cluster-based cross-view filtering for compressed multi-view depth maps," in *Proc. Vis. Commun. Image Process. (VCIP)*. IEEE, 2016, pp. 1–4.

[10] You Yang, Qiong Liu, Xin He, and Zhen Liu, "Cross-view multi-lateral filter for compressed multi-view depth video," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 302–315, 2019.

[11] Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele, "Joint bilateral upsampling," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 96–100, Jul 2007.

[12] Derek Chan, Hylke Buisman, Christian Theobalt, and Sebastian Thrun, "A noise-aware filter for real-time depth upsampling," in *Proc. ECCV Workshop Multi-Camera Multi-Modal Sensor Fusion Algorithm Appl.*, Oct 2008, pp. 1–12.

[13] James Diebel and Sebastian Thrun, "An application of markov random fields to range sensing," in *Advances in neural information processing systems*, 2006, pp. 291–298.

[14] Anh Vu Le, Seung-Won Jung, and Chee Sun Won, "Directional joint bilateral filter for depth images," *Sensors*, vol. 14, no. 7, pp. 11362–11378, 2014.

[15] Shuhang Gu, Wangmeng Zuo, Shi Guo, Yunjin Chen, Chongyu Chen, and Lei Zhang, "Learning dynamic guidance for depth image enhancement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, 2017, pp. 712–721.

[16] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang, "Deep joint image filtering," in *Proc. European Conf. Comput. Vis. (ECCV)*. Springer, 2016, pp. 154–169.

[17] Xu Wang, Pingping Zhang, Yun Zhang, Lin Ma, Sam Kwong, and Jianmin Jiang, "Deep intensity guidance based compression artifacts reduction for depth map," *Journal of Visual Communication and Image Representation*, vol. 57, pp. 234–242, 2018.

[18] Richard Szeliski, *Computer vision: algorithms and applications*, Springer Science & Business Media, 2010.

[19] Mary C Seiler and Fritz A Seiler, "Numerical recipes in c: the art of scientific computing," *Risk Analysis*, vol. 9, no. 3, pp. 415–416, 1989.

[20] Daniel Scharstein and Richard Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, Jun 2003, vol. 1, pp. I–195–I–202.

[21] Heiko Hirschmuller and Daniel Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, Jun 2007, pp. 1–8.

[22] Eduardo SL Gastal and Manuel M Oliveira, "Adaptive manifolds for real-time high-dimensional filtering," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 33:1–33:13, Jul 2012.

[23] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[24] Jun Xie, Rogerio Schmidt Feris, and Ming-Ting Sun, "Edge-guided single depth image super resolution," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 428–438, 2016.