

Supplemental Material for SoundSil-DS: Deep Denoising and Segmentation of Sound-field Images with Silhouettes

Risako Tanigawa^{1,2}, Kenji Ishikawa¹, Noboru Harada¹, and Yasuhiro Oikawa²

¹NTT Communication Science Laboratories, NTT Corporation, Atsugi, Kanagawa 243-0198, Japan

²Intermedia Art and Science, Waseda University, Shinjuku, Tokyo 169-8555, Japan

{risako.tanigawa, ke.ishikawa, harada.noboru}@ntt.com, yoikawa@waseda.jp

We summarize the content of the supplementary material as follows. Section 1 presents the issue with using the existing denoising/segmentation methods in supporting the data underlying the motivation of our task. Section 2 provides details on the creation of noise data for silhouette regions based on the experimental data. Section 4 provides the implementation details of the compared models.

1. Issue with Existing Methods

DSFD [4] focuses only on the sound field without objects. Thus, it cannot be directly applied to a sound field with objects. To provide evidence for this, denoising results for sound-field images with object silhouettes are shown in Fig. 1. The denoising was carried out by DSFD trained with the without-silhouette (w/o silhouette) dataset. The trained model was obtained from the publicly available GitHub repository of the author of DSFD [3]. We created the evaluation data, which included object silhouettes. The sound waves appeared inside the silhouette regions on the second-row images. Therefore, the DSFD cannot properly denoise the sound-field images, especially in the silhouette regions.

For segmentation, it may be natural to use a foundational model designed for natural image segmentation. To con-

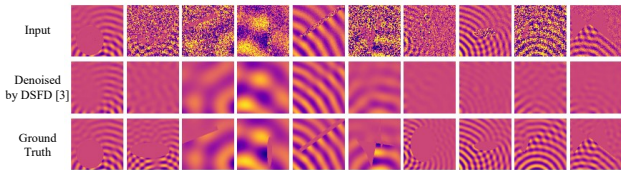


Figure 1. Denoising results estimated using DSFD trained with w/o silhouette dataset

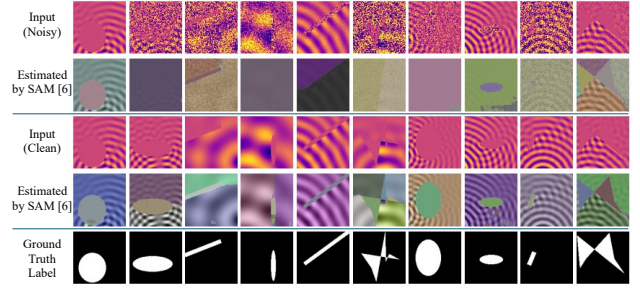


Figure 2. Segmentation results estimated by SAM

firm the applicability of the image segmentation foundation model, we conducted a preliminary experiment. Segment Anything Model (SAM) [6] was used to estimate zero-shot segmentation labels. To handle sound field data with SAM, we extracted only the real part channel of the input sound field images (floating-point numbers), converted them to ranging from 0 to 255, and then transformed them into 1-channel images similar to Grayscale images. Subsequently, these images were converted to RGB for input into SAM. The segmentation results are shown in Fig. 2. The top two rows are the input and segmented images for noisy data, the next two rows are the input and segmented images for clean data, and the last row is the ground truth of the segmentation labels. The visualization of the segmentation masks obtained by SAM is performed by overlaying randomly assigned colors for each mask on the input images. Therefore, the same color represents a single segmentation mask. For the results with noisy images as input (See the second-row of Fig. 2), the segmentation does not perform well where the noises in the input image are high, for example, the second, fourth, sixth, and ninth columns from the left in the Fig. 2. For the results with clean images as input (See the fourth-

row of Fig. 2), there are no images where the object silhouettes are entirely unsegmented. However, some images show multiple segments within the same object silhouettes, for example, the first, second, and fourth columns from the left in the Fig. 2. From these results, it can be concluded that even with the foundation model for image segmentation, SAM, attempting zero-shot segmentation on the noisy data is ineffective. Furthermore, the performance, even with the clean data, is inadequate. Hence, we considered the task of joint training and inferring denoising and segmentation.

2. Noise Creation based on Experimental Data

As mentioned in the main paper regarding dataset creation, we calculated the noise for silhouette regions from experimentally obtained data. In this section, we provide supplemental information for data collection and noise creation.

We estimated the probability density function (PDF) by kernel density estimation (KDE) on the basis of experimentally measured data. The data were collected by parallel phase-shifting interferometry (PPSI) [5]. The experimental setup is shown in Fig. 3(a). We installed a shielding object between two optical flats and recorded the data five times. The frame rate of the high-speed camera in the PPSI system was set to 20,000 frames per second, and 200 images were collected for each recording. To remove the low-frequency noise, a high-pass filter with a 500-Hz cut-off frequency was applied to the recorded images along the time direction. The real and imaginary parts of the Fourier-transformed data were regarded as one image. The single pixel value was regarded as one sample, and 28,800,000 samples in total were used for estimating the PDF. Histograms of the measured data and estimated PDF are shown in Fig. 3(b). There is good agreement between the estimated PDF and histogram of the measured data.

Noise data for silhouette regions were generated on the basis of the estimated PDF by using the inverse transform sampling method. An example of the generated data is shown in Fig. 3(c). The left and right figures show the measured and generated data, respectively. The generated data were sampled data based on the estimated PDF corresponding to the number of pixels in the image and reshaped to match the image dimensions. We confirmed that the generated noise data was similar to the measured data, except for spatial patterns originating from the optical elements.

3. Preliminary experiment for loss function

To determine the loss function for the proposed method, we conducted a preliminary experiment to compare performance by loss functions. For denoising loss L_{denoise} , mean squared error (MSE), mean absolute error (L1), and negative peak signal-to-noise ratio (N-PSNR) losses were com-

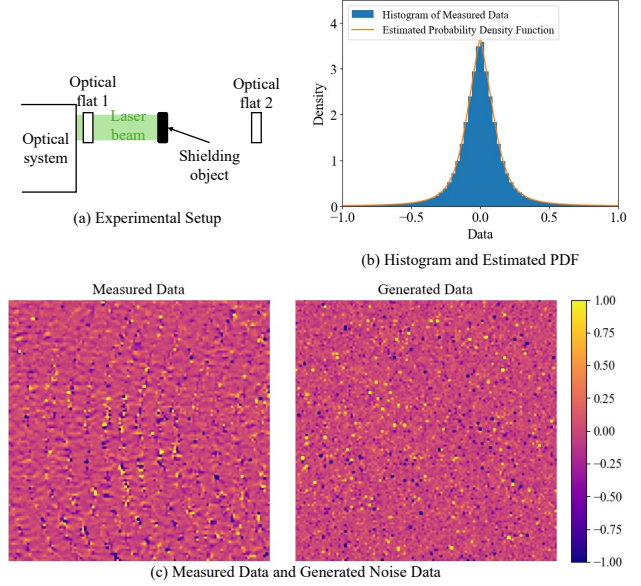


Figure 3. Noise-data creation based on experimental data. (a) Experimental setup of data collection. (b) Histogram of measured data and estimated probability density function (PDF). (c) Measured data and generated noise data.

L_{denoise}	L_{seg}	λ	PSNR [dB]	SSIM	IoU
MSE	BCE	0.001	40.8	0.983	0.981
MSE	BCE+Dice	0.001	41.5	0.984	0.984
L1	BCE	0.01	42.2	0.986	0.980
L1	BCE+Dice	0.01	42.3	0.987	0.982
N-PSNR	BCE	10	43.2	0.987	0.985
N-PSNR	BCE+Dice	10	43.2	0.987	0.986

Table 1. Comparison of loss function. Negative PSNR loss and balanced BCE and Dice loss were best for denoising and segmentation, respectively.

pared. For segmentation loss L_{seg} , binary cross entropy (BCE) and balanced BCE and dice (BCE+Dice) losses were compared. We conducted training and evaluation with 6 patterns of all combinations of 3 loss functions for denoising and 2 loss functions for segmentation. The evaluation result is shown in Tab. 1. The weighting coefficient λ was set to roughly matching digits of loss values. Using N-PSNR as L_{denoise} was the best performance for denoising. For segmentation, using BCE+Dice loss as L_{seg} was the best performance for segmentation. Since combination of N-PSNR and BCE+Dice marked best performance in both denoising and segmentation, we selected them as loss functions for proposed method.

4. Implementation Details

In this section, the implementation details for the compared models are provided. The following parameters were common to all models. All models were implemented by PyTorch. The loss function for segmentation L_{seg} was the combination of binary cross entropy loss L_{BCE} and dice loss L_{Dice} : $L_{\text{seg}} = (1 - \alpha)L_{\text{BCE}} + \alpha L_{\text{Dice}}$ with the weighting coefficient $\alpha = 0.5$. The number of epochs was set to 20. The number of channels of input and output layers were set to 2 and 3, respectively. The parameters that differ for each model are listed below.

DnCNN [7] The denoising and segmentation model based on DnCNN was implemented by referencing publicly available code from the DSFD repository [3]. The network architecture was almost the same as in the original paper [7] except for the number of input/output channels. The Adam optimizer was used where the learning rate was 0.001, and β_1 and β_2 were 0.9 and 0.999, respectively. The exponential learning rate scheduler was used where the multiplicative factor γ was 0.95. MSE loss was used as the loss function for denoising L_{denoise} .

LRDUNet [2] The denoising and segmentation model based on LRDUNet was implemented by referencing publicly available code from the DSFD repository [3]. The network architecture was almost the same as in the original paper [2] except for the number of input/output channels. The Adam optimizer was used where the learning rate was 0.001, and β_1 and β_2 were 0.9 and 0.999, respectively. The exponential learning rate scheduler was used where the multiplicative factor γ was 0.95. L1 loss was used as the loss function for denoising L_{denoise} .

NAFNet [1] The denoising and segmentation model based on NAFNet was implemented by referencing publicly available code from the DSFD repository [3]. The network architecture was almost the same as in the original paper [1] except for the number of input/output channels. The Adam optimizer was used where the learning rate was 0.001, and the β_1 and β_2 were 0.9 and 0.999, respectively. The exponential learning rate scheduler was used where the multiplicative factor γ was 0.95. MSE loss was used as the loss function for denoising L_{denoise} .

KBNet [9] The denoising and segmentation model based on KBNet was implemented by referencing publicly available code from the KBNet repository [8]. The network architecture was almost the same as in the original paper [9] except for the number of input/output channels. The AdamW optimizer was used where the learning rate was $3e-4$, weight decay was $1e-4$, and β_1 and β_2 were 0.9 and

Evaluation data	PSNR [dB]	SSIM	IoU
w/o silhouettes	43.5	0.991	1.00 (for class 0)
w/ silhouettes	43.2	0.987	0.986 (for class 1)

Table 2. Evaluation for w/o silhouette dataset

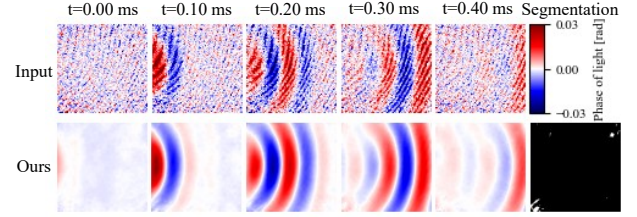


Figure 4. Experimental results of w/o silhouette sound field. Color indicates phase of light detected with PPSI.

0.999, respectively. The cosine annealing with the restart learning rate scheme was used where the periods for each cosine annealing cycle were set to 92000 and 208000, the restart weights at each restart iteration were all set to 1, and the minimum learning rates at each cycle were set to $3e-4$ and $1e-6$. L1 loss was used as the loss function for denoising L_{denoise} .

5. Evaluation of Denoising Performance for Sound Fields without Objects

To confirm the applicability of the proposed method to sound-field images without object silhouettes, we created an evaluation dataset without objects. The parameters of the dataset, such as the positions, frequencies, and sound pressures of the sound sources, are the same as those of the dataset described in Sec 3.2 of the main paper except for the existence of objects. The evaluation result is shown in Tab. 2. The trained model with the with-silhouette dataset (w/ silhouettes) was used for the evaluation. The IoU was calculated for class 0 (sound fields), where w/o silhouette data was used for the evaluation. These results indicate that the proposed method can be applied to sound fields without objects even if the network is only trained on data w/ silhouettes.

For further verification, the results applied to the experimental data without objects are shown in Fig. 4. In this experiment, sound images of a 12-kHz burst wave generated from a loudspeaker (FOSTEX FT48D) [4] were used. The top row is the input data where the burst wave propagated from left to right. The noise was eliminated by our method. For segmentation, although all values should be 0 (black), some pixels were falsely detected as silhouette class (white).

References

- [1] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *ECCV*, pages 17–33, 2022. 3
- [2] Javier Gurrola-Ramos, Oscar Dalmau, and Teresa Alarcón. U-Net based neural network for fringe pattern denoising. *Optics and Lasers in Engineering*, 149:14 pages, 2022. 3
- [3] Kenji Ishikawa, Daiki Takeuchi, Noboru Harada, and Takehiro Moriya. deep-sound-field-denoiser. <https://github.com/nttcs-lab/deep-sound-field-denoiser>, 2023. Accessed on 1st, July, 2024. 1, 3
- [4] Kenji Ishikawa, Daiki Takeuchi, Noboru Harada, and Takehiro Moriya. Deep sound-field denoiser: optically-measured sound-field denoising using deep neural network. *Optics Express*, 31:33405–33420, 2023. 1, 3
- [5] Kenji Ishikawa, Kohei Yatabe, Nachanant Chitanont, Yusuke Ikeda, Yasuhiro Oikawa, Takashi Onuma, Hayato Niwa, and Minoru Yoshii. High-speed imaging of sound using parallel phase-shifting interferometry. *Optics Express*, 24(12):12922–12932, Jun 2016. 2
- [6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *ICCV*, pages 4015–4026, October 2023. 1
- [7] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 3
- [8] Yi Zhang, Dasong Li, Xiaoyu Shi, Dailan He, Kangning Song, Xiaogang Wang, Honwei Qin, and Hongsheng Li. KBNet. <https://github.com/zhangyi-3/kbnet>, 2023. Accessed on 1st, July, 2024. 3
- [9] Yi Zhang, Dasong Li, Xiaoyu Shi, Dailan He, Kangning Song, Xiaogang Wang, Honwei Qin, and Hongsheng Li. KBNet: Kernel basis network for image restoration. *arXiv preprint arXiv:2303.02881*, 2023. 3