Structure-Texture-Noise Decomposition for Noisy Images with Two-Stage Network

Takamichi Miyata
Chiba Institute of Technology
Chiba, Japan
takamichi.miyata@it-chiba.ac.jp

***Abstract*—Structure-texture decomposition refers to the decomposition of an input image into three components: a structure component consisting of edges and smooth surface, a texture component consisting of local patterns. Most existing methods do not take the existence of the noise into account, but in practice, the noise is often included in the input image. In this study, we propose a deep learning based structure-texture-noise decomposition method that enables texture-noise separation using the context of structure components by sequentially connected two-stage network. Experimental results show that the proposed method can decompose noisy images into structure, texture, and noise components. Furthermore, we show that the proposed method can be applied to the tone mapping application with noisy input.**

***Index Terms*—structure-texture-noise decomposition, deep learning, edge preserving image smoothing, tone mapping**

I. Introduction

Structure-texture decomposition (or edge-preserving image smoothing) from an image is an important inverse problem in the image processing research field. It can be applied to tone mapping and texture enhancement and is also useful as preprocessing for high-level computer vision tasks. In general, structure and texture are defined as piecewise smooth component and local variation pattern component, respectively. This definitional ambiguity leads to a large number of decomposed image pairs corresponding to a single input image and makes structure-texture decomposition an inherently ill-posed problem.

Existing optimization based methods for structure-texture decomposition can be classified into those that try to represent both structure and texture features [1], [2], and those that extract only the structure component [3]–[5]. In the latter case, the texture component is obtained by subtracting the structure component from the input image. These optimization based methods are required to solve the large scale optimization problems by using iterative methods and thus require significant computational time.

In recent years, deep learning based approaches have been applied to structure-texture decomposition [6]–[9]. The advantage of deep learning based methods is that their inference is extremely fast compared to the optimization based methods. Most of these deep learning based methods use the definition that the texture component is the difference between the input image and the structure.

On the other hand, real images often contain noise that is unavoidable during the image acquisition process, which degrades the performance of the structure-texture decomposition. Optimization based methods can decompose such noise relatively easily by incorporating the log-likelihood of the probability distribution of the noise into their objective function. However, deep learning based methods cannot distinguish between the noise and the texture because the texture is defined as the difference between the input and the structure.

In this paper, we propose a structure-texture-noise decomposition method using deep learning. By employing a two-stage architecture and loss function, the proposed method is extremely fast and accurate in decomposing the three components. Experimental results show that the proposed method can also be applied to tone mapping for noisy high dynamic range (HDR) images.

II. Proposed Method

The existing structure-texture decomposition methods assume that an input original image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ is the sum of $\mathbf{s}$ and $\mathbf{t}$ such that $\mathbf{x} = \mathbf{s} + \mathbf{t}$, and finds $\mathbf{s}$ and $\mathbf{t}$ from $\mathbf{x}$ under this constraint. Where H, W, and C are image height, width, and number of channels. However, as we mentioned before, real images are often contaminated by noise which is inevitable in the image acquisition process. The noise component included in the input image may degrade the performance of the image decomposition methods.

We proposed a method for decomposing a noisy input image into structure, texture, and noise components. We assume that the input image is noisy observation $\mathbf{y} = \mathbf{x} + \mathbf{n}$, which is the addition of the original image $\mathbf{x}(= \mathbf{s} + \mathbf{t})$ and the instance of i.i.d. Gaussian noise $\mathbf{n}$ with standard deviation σ. Our objective is to decompose $\mathbf{y}$ into the three intrinsic components $\mathbf{s}$, $\mathbf{t}$, and $\mathbf{n}$ respectively.

A. Network architecture

We propose a two-stage, cascaded network architecture for structure-texture-noise decomposition (Fig. 1). Each stage (stage-1 and stage-2) consists of a very deep convolutional neural network (VDCNN) [8] which is a deep learning based edge-preserving image smoothing method.

This work was supported by research grant from the Telecommunications Advancement Foundation and JSPS KAKENHI Grant Number JP23K03871. The author would like to thank Hiroaki Hirano for his contributions to this work.

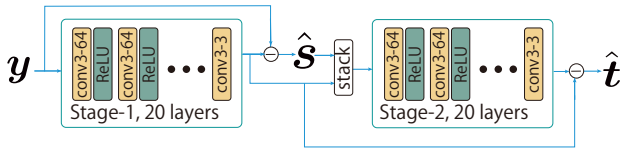


Fig. 1: Architecture of the proposed method.

The kernel sizes of all convolutional layers of VDCNN are 3×3, with 64 output channels from the first to the 19th layers and 3 output channels in the final layer.

The first stage (stage-1 in Fig, 1) takes an observed image $\mathbf{y}$ containing noise as input and the output is the $\hat{\mathbf{s}}$. The neural network $\Phi_1 : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{H\times W\times C}$ is trained to estimate $\mathbf{t}+\mathbf{n}$ which is the sum of two components, texture, and noise, from the input image $\mathbf{y}$. Then, the estimated structural component $\hat{\mathbf{s}}$ which is the output of stage-1 is obtained as the difference between the input $\mathbf{y}$ and the output $\Phi_1(\mathbf{y})$, i.e. $\hat{\mathbf{s}} = \mathbf{y} - \Phi_1(\mathbf{y})$.

The second stage (stage-2 in Fig, 1) decomposes texture and noise components. The estimated structure component $\hat{\mathbf{s}}$ obtained in the first stage and $-\Phi_1(\mathbf{y})$, are stacked in the channel direction. Thus $\mathrm{stack}(\hat{\mathbf{s}}, -\Phi_1(\mathbf{y}))$ is used as the input of stage-2. Where $\mathrm{stack}(\mathbf{a}, \mathbf{b})$ is an operator that stacks the input tensors $\mathbf{a}, \mathbf{b}$ in the channel direction. This is expected to improve separation performance by enabling texture-noise separation using the context of structural components.

Let $\Phi_2 : \mathbb{R}^{H\times W\times 2C} \rightarrow \mathbb{R}^{H\times W\times C}$ be the VDCNN(2) process The output estimated texture component $\hat{\mathbf{t}}$ is obtained as the sum of $-\Phi_1(\mathbf{y})$ and the VDCNN(2) output $\Phi_2(\mathrm{stack}(\hat{\mathbf{s}}, -\Phi_1(\mathbf{y})))$. The estimated original image $\hat{\mathbf{x}} = \hat{\mathbf{s}}+\hat{\mathbf{t}}$ is obtained from the sum of the estimated structure and texture components.

B. Loss function

The loss function of the proposed method consists of four terms.

$$L = L_{\mathrm{NB}}\,(\hat{\mathbf{s}}, \mathbf{s}) + \alpha \parallel \hat{\mathbf{s}} - \mathbf{s} \parallel_1 + \beta \parallel \hat{\mathbf{t}} - \mathbf{t} \parallel_1 + \gamma \parallel \hat{\mathbf{x}} - \mathbf{x} \parallel_2, \tag{1}$$

where $\hat{\mathbf{s}}$, $\hat{\mathbf{t}}$, and $\hat{\mathbf{x}}$ are the estimated structure, texture and image, respectively, and $\mathbf{s}$, $\mathbf{t}$, and $\mathbf{x}$ are the grandtruth of each component and image. The α, β, and γ are arbitrary parameters to be determined by the user. The term L_{NB} is the neighborhood loss [8] is defined by the following equation.

$$L_{\mathrm{NB}}(\hat{\mathbf{s}}, \mathbf{s}) = \sum_{i,j} \sum_{(p,q)\in N_{i,j}} \left\| (\hat{\mathbf{s}}_{i,j} - \hat{\mathbf{s}}_{p,q}) - (\mathbf{s}_{i,j} - \mathbf{s}_{p,q}) \right\|_1 \tag{2}$$

where $N_{i,j}$ denotes the 5×5 neighborhood centered at the pixel location (i, j). The neighborhood loss is to encourage the estimated structural component to have the same local variation as the groundtruth structural component.

III. Experimental results

A. Dataset and Training

We used 400 selected images from the BSDS500 dataset [10] to train the proposed network. The remaining 100 images were used for testing. For comparison experiments, train/test splitting was performed exactly as in Zhu et al [8]. Each clean image $\mathbf{x}$ were decomposed into $\mathbf{t}$ and $\mathbf{s}$ by L1smooth [5]. We used these components as the groundtruth of texture and structure components, respectively. The Gaussian noise with standard deviation σ was added to the image x to simulate the observed image y.

We use ADAM optimizer for the training. The hyperparameters of the proposed method are shown in Table I.

TABLE I: Hyperparameters of the proposed method.

σ	α	β	γ	epochs	learning rate
0	1.0	1.0	1.0	2.0×10^4	1.0×10^{-4}
15	1.0	1.0	1.0	2.0×10^4	1.0×10^{-4}
25	1.0	1.0	1.5	4.0×10^4	5.0×10^{-5}
50	1.0	1.0	2.0	1.0×10^5	2.5×10^{-5}

B. Evaluation of Separation Performance

To evaluate the decomposition performance of the proposed method, we compared it with VDCNN [8]. The proposed method is trained by the input images without and with noise, respectively. We evaluate the decomposed structure and texture component compared with their grandtruth by using weighted mean absolute error (wMAE) and weighted root mean squared error (wRMSE) as evaluation criterion which are defined as follows:

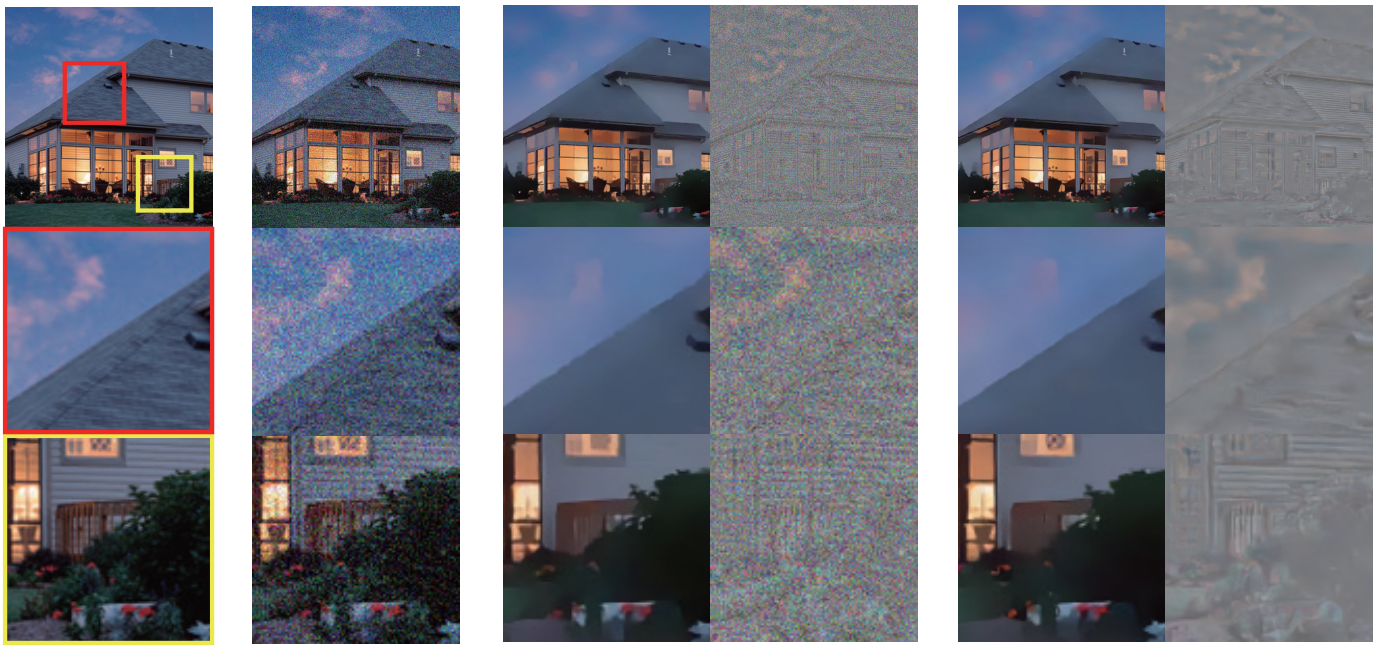
$$\mathrm{wMAE} = \frac{1}{HWC} \sum_{i,j} \sum_{k=1}^{5} w_k \left\| \hat{\mathbf{s}}_{i,j} - \mathbf{s}^k_{i,j} \right\|_1, \tag{3}$$

$$\mathrm{wRMSE} = \left(\frac{1}{HWC} \sum_{i,j} \sum_{k=1}^{5} w_k \left\| \hat{\mathbf{s}}_{i,j} - \mathbf{s}^k_{i,j} \right\|_2^2 \right)^{\frac{1}{2}}, \tag{4}$$

where the parameter $w_k \in \mathbb{R}$ are the weights based on the user evaluated values of the components $\mathbf{s}^k$ and $\mathbf{t}^k$ obtained from the top five structure-texture decomposition methods. Note that the weights are normalized so that $\sum_{k=1}^{5} w_k = 1$ and depend on the image. We used the weights published by Zhu et al. [8].

The averaged performance of VDCNN and our proposed method are shown in Table II. There is no significant difference in the estimation performance of the structure components regardless of the presence or absence of noise. However, for the estimation of texture components from the noisy images, the proposed method outperforms the VDCNN. This result suggests that the proposed method can successfully decompose the texture component from the noise component by the network in the second stage.

Figure 2 shows the example of the decomposition results. While the VDCNN fails to separate the texture and noise components, the proposed method can successfully separate these components from the noisy input image.



(a) (b) (c) (d) (e) (f)

Fig. 2: Decomposition results with noise level $\sigma = 25$: (a) original image $\mathbf{x}$, (b) noisy input $\mathbf{y}$, (c) structure $\hat{\mathbf{s}}$ by VDCNN [8], (d) texture $\hat{\mathbf{t}}$ by VDCNN [8], (e) structure $\hat{\mathbf{s}}$ by proposed method, (d) texture $\hat{\mathbf{t}}$ by proposed method.

TABLE II: Evaluation of estimated structure and texture components for each noise level.
The standard deviation of the Gaussian noise is σ; the larger the value, the stronger the noise. The best results for each noise level and criteria are indicated in bold.

		structure		texture	
σ	method	wMAE	wRMSE	wMAE	wRMSE
0	VDCNN	**6.20**	**9.78**	**6.20**	**9.78**
	Ours	6.21	9.87	6.21	9.87
15	VDCNN	**7.00**	**10.55**	15.02	19.17
	Ours	7.03	10.57	**7.21**	**10.65**
25	VDCNN	7.98	11.74	20.90	26.35
	Ours	**7.85**	**11.51**	**7.87**	**11.26**
50	VDCNN	9.26	13.55	40.76	51.66
	Ours	**9.18**	**13.45**	**8.25**	**11.71**

C. Computational time

We compared the computational time of the proposed method with that of L1 smoothing [5], an optimization-based method. L1 smoothing is run on a CPU (Intel Corei7-11700 2.50GHz), while the proposed method is run on a CPU and GPU (NVIDIA GeForce RTX3080ti), respectively. L1 smoothing was implemented in Matlab, and the proposed method was implemented in Python.

Ten images were randomly selected from the dataset used to evaluate the separation performance, and the processing time for each image was measured. The table III shows the processing time results. This table shows that the proposed method takes less time than the optimization-based method for both CPU and GPU. The proposed method can be executed in tens of milliseconds when using a GPU, which is fast enough for various applications.

TABLE III: The average computational time for 10 images for each method.

method	Processor	time [s]
L1 smoothing [5]	CPU	85
Ours	CPU	4.0
Ours	GPU	**0.08**

D. Ablation Study

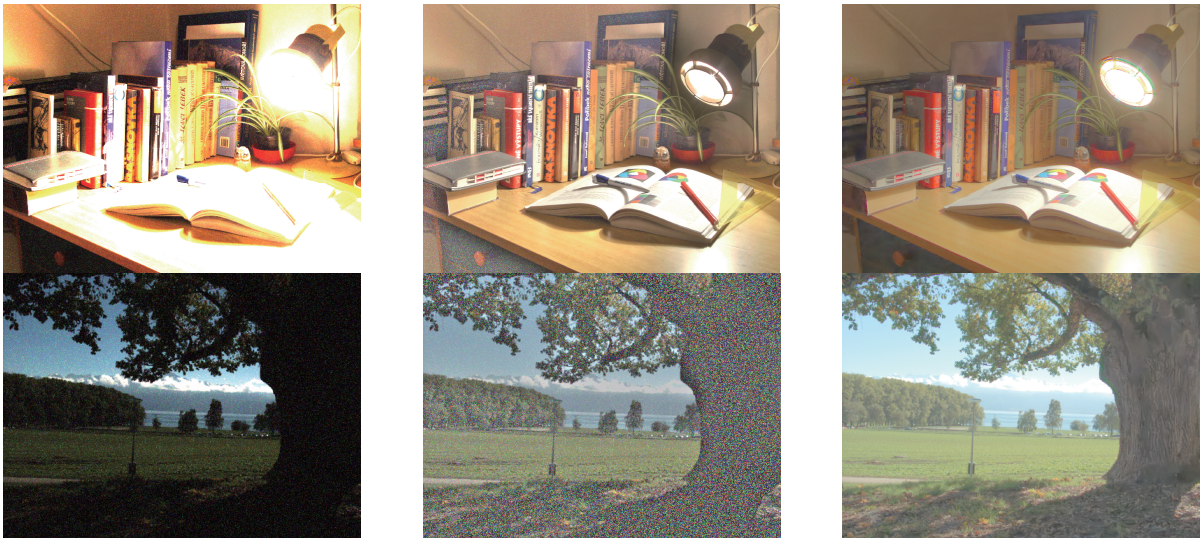
In the proposed method we use the estimated structure component $\hat{\mathbf{s}}$ stacked with the sum of the texture component and noise $-\Phi_1(\mathbf{y})$ as input to stage-2. The table IV shows the results when this input is changed to

- Only the sum of texture component and noise (-$\Phi_1(\mathbf{y})$))
- Further stacked with the observed image ($\mathrm{stack}(\hat{\mathbf{s}}, -\Phi_1(\mathbf{y}))$).

This result means that the input selected in this paper show the best performance.

TABLE IV: Comparison of decomposition performance for different inputs of the stage-2 network.

	structure		texture	
input of Φ_2	wMAE	wRMSE	wMAE	wRMSE
$\mathrm{stack}(\hat{\mathbf{s}}, -\Phi_1(\mathbf{y}))$ (ours)	**7.98**	**11.51**	**7.87**	**11.26**
$\Phi_1(\mathbf{y})$	8.15	11.82	8.03	11.52
$\mathrm{stack}(\hat{\mathbf{s}}, -\Phi_1(\mathbf{y}), \mathbf{y})$	7.99	11.58	7.96	11.30



(a) (b) (c)

Fig. 3: Tone mapping results with noise level $\sigma = 25$: (a) tone mapped noisy HDR image, (b) tone mapped result of WLS [11], (c) tone mapped result by the proposed method.

IV. Application to Tone Mapping

Tone mapping is a technique for converting high dynamic range (HDR) images to low dynamic range (LDR) images. In this section, we apply our proposed method to tone mapping from noisy HDR images. Our implementation is based on the methods proposed in [11], [12]. We use tone mapped image quality index (TMQI) [13] as the evaluation metric and 14 HDR images from the subject-rated image database for tone-mapped images [14] as the input images. Table V shows the quantitative evaluation results. The table shows that the proposed method outperforms WLS [11] when the input image contains noise. Figure 3 shows the qualitative results. These images show that the proposed method can adequately distinguish the texture and noise, although artifacts are observed at the high contrast edges.

TABLE V: Average TMQI of tone mappped 14 HDR images for each noise level. The higher the value, the better the result.

σ	0	15	25	50
WLS [11]	**0.920**	0.787	0.739	0.681
Ours	0.908	**0.875**	**0.870**	**0.865**

V. Conclusion

We proposed a new two-staged structure-texture-noise decomposition network for the noisy input images. The experimental results show that the proposed method can decompose the noisy input image into three components. We also show that the proposed method can be applied to tone mapping for noisy HDR images. Our future work includes the application of the proposed method to other image processing tasks such as texture enhancement.

References

[1] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher, "Structure-Texture image Decomposition—Modeling, algorithms, and parameter selection," *Int. J. Comput. Vis.*, vol. 67, no. 1, pp. 111–136, 2006.

[2] S. Ono, T. Miyata, and I. Yamada, "Cartoon-texture image decomposition using blockwise low-rank texture characterization," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1128–1142, 2014.

[3] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via l 0 gradient minimization," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6, 2011, p. 174.

[4] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 1–10, Nov. 2012.

[5] S. Bi, X. Han, and Y. Yu, "An l1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition," *ACM Trans. Graph.*, vol. 34, no. 4, 2015.

[6] L. Xu, J. Ren, Q. Yan, R. Liao, and J. Jia, "Deep edge-aware filters," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 1669–1678.

[7] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with Fully-Convolutional networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2516–2525.

[8] F. Zhu, Z. Liang, X. Jia, L. Zhang, and Y. Yu, "A benchmark for edge-preserving image smoothing," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3556–3570, 2019.

[9] Y. Feng, S. Deng, X. Yan, X. Yang, M. Wei, and L. Liu, "Easy2hard: Learning to solve the intractables from a synthetic dataset for structure-preserving image smoothing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 7223–7236, 2022.

[10] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[11] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graph.*, vol. 27, no. 3, 2008.

[12] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Graph.*, vol. 21, no. 3, p. 257–266, 2002.

[13] H. Yeganeh and Z. Wang, "Objective quality assessment of tone-mapped images," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 657–667, 2013.

[14] "TMQI: Tone mapped image quality index project page," https://ece.uwaterloo.ca/~z70wang/research/tmqi/.