Structure-Texture Image Decomposition using Deep Variational Priors

Youngjung Kim, *Member, IEEE,* Bumsub Ham, *Member, IEEE,* Minh N. Do, *Fellow, IEEE,* and Kwanghoon Sohn, *Senior Member, IEEE*

***Abstract*—Most variational formulations for structure-texture image decomposition force structure images to have small norm in some functional spaces, and share a common notion of edges, i.e., large-gradients or -intensity differences. However, such definition makes it difficult to distinguish structure edges from oscillations that have fine spatial scale but high contrast. In this paper, we introduce a new model by learning deep variational prior for structure images without explicit training data. An alternating direction method of multiplier (ADMM) algorithm and its modular structure are adopted to plug deep variational priors into an iterative smoothing process. The central observations are that convolution neural networks (CNNs) can replace the total variation prior, and are indeed powerful to capture the natures of structure and texture. We show that our learned priors using CNNs successfully differentiate high-amplitude details from structure edges, and avoid halo artifacts. Different from previous data-driven smoothing schemes, our formulation provides another degree of freedom to produce continuous smoothing effects. Experimental results demonstrate the effectiveness of our approach on various computational photography and image processing applications, including texture removal, detail manipulation, HDR tone-mapping, and non-photorealistic abstraction.**

***Index Terms*—Structure-texture image decomposition, total variation, adaptive neighborhood filtering, alternating direction method of multiplier algorithm, texture filtering.**

I. Introduction

DECOMPOSING an image into meaningful components is an essential operation in the fields of image processing, computational photography, and image analysis. In particular, it aims to decompose an image $\mathbf{f}$ into a structural part $\mathbf{u}$, corresponding to the main objects, and a residual texture part $\mathbf{v}$, containing fine scale-details. These representations are then reconciled to produce visually striking effects, such as detail enhancement [1], image composition [2], and non-photorealistic rendering [3]. Image decomposition is also useful in a range of computer vision applications [4]–[6]. It is effective in suppressing and/or extracting detrimental content in images, making it easier for subsequent high-level vision tasks including saliency detection [4], segmentation [5], and intrinsic image decomposition [6].

Since the number of unknowns $\mathbf{u}$ and $\mathbf{v}$ are larger than the input $\mathbf{f}$, image decomposition is a challenging ill-posed problem. Linear shift-invariant (LSI) methods using fixed kernels are simple, but unable to represent edges. They are not suitable for structure-preserving image decomposition [7]. This argument is often referred to as a motivation for more sophisticated schemes. We roughly classify existing methods into three categories: adaptive neighborhood filters, variational frameworks, and data-driven approaches. A variety of popular neighborhood filters can be expressed as functions of local histogram. The bilateral filter [8] computes a mean value of local histogram estimated by the Gaussian kernel weights, i.e., through a weighted averaging process. Several variants such as bilateral texture [9] and rolling guidance [10] filters have also been proposed to extract structure from texture. The median filter gives a value that corresponds to half of the local cumulative distribution [11], and similarly the mode filter [12] seeks all the modes in the local histogram. While these neighborhood filters are simple and intuitive to use, they are not suitable for image decomposition and introduce halo artifacts [13] due to their local nature.

Alternatively, structure-texture image decomposition can be achieved by modern variational techniques. Although the aim of image decomposition is hard to formulate explicitly, various variational models have been employed to decompose an image into meaningful structural and textural parts. Total Variation (TV) [14] is the most popular variational prior that regularizes the L_1-norm of image gradients. The success of TV prior prompted the in-depth studies on variational framework for image decomposition, including L_0 smoothing [3], relative TV (RTV) [2], and nonlocal TV [15]. These methods are commonly based on the assumption that the structure image $\mathbf{u}$ tends to have small TV, L_0-norm of gradients, and so on. However, such contrast-based assumption might fail to separate high-frequency components that are related to fine image details or textures. Recently, convolutional neural networks (CNNs) have been applied in numerous image processing and computational photography problems, including structure-texture image decomposition [16], [17]. Since no pairs of real $(\mathbf{u}, \mathbf{v})$ are available for supervised training, the authors of [16], [17] try to imitate the existing methods [1]–[3] within a unified CNN architecture. Therefore, they do not produce new decomposition results, and inherit the aforementioned limitations. They also require heuristic interpolations to continuously adjust the smoothing strength.

Y. Kim is with the Institute of Defense Advanced technology Research, Agency for Defense Development, Daejeon 340-60, South Korea (e-mail: read12300@add.re.kr).

B. Ham and K. Sohn are with the School of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, South Korea (e-mail: mimo@yonsei.ac.kr; khsohn@yonsei.ac.kr).

M. N. Do is with the University of Illinois at Urbana-Champaign, Urbana, IL 61820, USA (e-mail: minhdo@illinois.edu).

This research was supported by Multi-Ministry Collaborative R&D Program (R&D program for complex cognitive technology) through the National Research Foundation of Korea (NRF) funded by MSIT, MOTIE, KNPA (NRF-2018M3E3A1057289).

In this paper, we take advantages of both the variational framework and recent data-driven approach for structure-texture image decomposition. Specifically, we first introduce a deep variational prior for structure components $\mathbf{u}$, which is characterized by deep projection networks. The network configuration is inspired from the Chambolle's projection algorithm [18] for minimizing the classical TV prior [7], [14], and its parameters are trained without explicit $(\mathbf{u}, \mathbf{v})$ training data. An alternating direction method of multiplier (ADMM) algorithm and its modular property are then adopted to plug the deep variational prior into iterative smoothing process. We show that the projection networks can replace the TV prior [7], [14] which has been widely used in variational image decomposition, and are indeed powerful to capture the nature of texture. The proposed method successfully differentiates high-amplitude details from structure edges, and can be applied to a wide range of texture scales. Another appealing aspect is that, unlike previous data-driven approaches [16], [17], our method provides the degree of freedom to continuously adjust the smoothing strength. We also express the necessary and sufficient condition, guaranteeing that the proposed method converges to a fixed point. We take this into consideration when designing the deep projection network. The main contributions of this work are summarized as follows:

- Inspired from the classical TV minimization [7] for image decomposition, we introduce a deep variational prior, which can be trained without explicit $(\mathbf{u}, \mathbf{v})$ training data.
- Our formulation takes advantages of both the variational framework and data-driven approach. It can be used to a wide range of texture scales and can produce continuous smoothing effects.
- We identify the necessary condition for the fixed point convergence of the proposed method, and show that a few iterations are enough to obtain the decomposition results.

Experimental results demonstrate the effectiveness of our approach in several image processing and computational photography applications.

The remainder of this paper is organized as follows. Section II describes related works for structure-texture image decomposition. We present the proposed method and training details in Section III. A comprehensive experimental analysis is then provided in Section IV to demonstrate the flexibility of the proposed method. In Section V, we apply our method to image processing and computational photography applications. Finally, conclusion and discussion are given in Section VI.

II. Related Work

In this section, we briefly review the structure-texture image decomposition methods. Among various methodologies, we discuss three lines of research that are most relevant to ours: adaptive neighborhood filter, variational framework, and data-driven approach.

A. Adaptive Neighborhood Filter

The bilateral filter (BF) [8] is one of the most widely used weighted-average filters for image decomposition, and has been re-discovered several times by [19], [20]. It computes the structure part $\mathbf{u}$ at each pixel as the average of neighboring pixels, weighted by both spatial and range kernels. The BF smooths the image without crossing structure edges, and finds several applications such as HDR tone-mapping [21], noise estimation [22], and image abstraction [23]. The guided filter [13], adaptive manifolds [24], and domain trainsform [25] are popular alternatives to the BF. These methods can produce a similar smoothing effect to the BF, while keeping a linear time complexity. Kass *et al.* [12] expressed a weighted averaging process as functions of the local histogram, and generalized histogram-based operators, such as median and mode filters.

Note that the aforementioned methods are not designed to explicitly handle textures, and thus applying them directly to the image decomposition task does not provide satisfactory results. Karacan *et al.* [26] adopted the region covariance descriptor to leverage the repetition property of textures. However, computing the region covariance matrix for per-pixel affinity is very time-consuming. The rolling guidance filter (RGF) [10] exploited the BF [8] iteratively guided by a Gaussian-blurred input image to eliminate textures smaller than a certain scale. Cho *et al.* [9] devised patch-shift mechanism where patches are found that likely stay clear of the structure edge, and incorporated it into the BF [8] for texture smoothing. Bao *et al.* [27] introduced a connectedness kernel by treating pixels as nodes in a minimum spanning tree to achieve the similar goal. Jeon *et al.* [28] used patch-based statistics and found an optimal per-pixel smoothing scale of the Gaussian kernel. The neighborhood filters are intuitive and easy to implement, but have some drawbacks. They do not maintain the global consistency of the output, and may often result in halo artifacts.

B. Variational Framework

Variational models have been developed for a wide range of imaging applications during the last decades. In that context, image decomposition methods based on variational models have become increasingly popular. The aim of these models is to find appropriate priors (or functional spaces) to describe structures and textures, respectively. For example, the ROF [14] model for image denoising can be regarded as a decomposition model [7]. It decomposes the image to a structure component belonging to the bounded variation and a noise or small-scale texture component in L_2. Aliney [29] pointed out that the ROF [14] model erodes structures and suggested to replace L_2 norm with L_1. In [30], Meyer introduced a new prior for texture components named as the space G to model oscillating patterns. A texture belonging to G may have large oscillations and nonetheless have a small norm. Very recently, Gu *et al.* [31] modeled texture components using joint convolutional analysis and synthesis sparse coding.

Beyond the TV [14] for structural components, many other priors have also been proposed. Farbman *et al.* [1] advocated the weighted least squares (WLS) framework where weights are computed from the input image itself. Xu *et al.* [3] proposed a sparse gradient counting scheme that is based on L_0-norm of image gradients. They used the quadratic relaxation

Fig. 1. (Left) the *Gypsy girl mosaic* image, (Right) the structure images u obtained by different $H_1 - H_2$ decompositions: (a) TV$-L_2$ (ROF) [14], (b) TV$-\Delta^{-1}$ [7], (c) RTV$-L_2$ [2], and ours. The existing variational methods for image decomposition have a difficulty in separating fine texture from structure. In contrast, our deep variational prior successfully differentiates them, and preserves shading information.

[32] to address the corresponding non-convex and non-smooth minimization problem. Similarly, Ono [33] employed L_0-norm of gradients, but formulated it as the hard constraint which is solved by L_0 projection algorithm. Bi *et al.* [15] enforced the global sparsity to the structure image by using the non-local TV. The RTV [2] used a pixel-wise windowed TV normalized by a windowed inherent variation. It is equivalent to applying the WLS [1] iteratively, where the weight is computed using the output of previous iteration. The SD filter [34] defined its structure prior with the Welsch's function, and adopted the majorize-minimize algorithm. The RTV [2] and SD filter [34] are only different in their affinity computations, except that the latter additionally incorporates the static guidance. The SD filter computes the weights with the exponential function while the RTV uses the derivative of logarithm function. Guo *et al.* [35] modified the RTV [2] by incorporating the concept of relaxed mutual response. The existing variational formulations rely on magnitudes of gradients (or pixel differences) at their heart. This makes it challenging to capture textures that have fine spatial scale but high contrast. Furthermore, the resulting image usually looks like piecewise constant, and thus shading information is not preserved.

C. Data-driven Approach

Recent progress in machine learning has motivated several data-driven approaches for image decomposition. As a pioneering work, Yang *et al.* [36] approximated the BF [8] using support vector regression (SVR). For each pixel, they defined a feature vector using the exponentiation of the image, and trained the mapping function to the bilateral filtered value with the SVR. Xu *et al.* [17] showed that the CNN can accelerate many conventional implementations of various image filters. Liu *et al.* [37] parameterized the weight maps for image decomposition that are propagated to the entire image through a recurrent neural network (RNN). Chen *et al.* [16] approximated a wide variety of variational models including the ROF [14], RTV [2], and L_0 minimization [3]. They showed that all models can be imitated by the same architecture [16]. These data-driven approaches often have a significant advantage in terms of running time. The current strategy, however, trains parametric models to approximate the output of existing neighborhood filters [8] or variational frameworks [2], [14].

Very recently, several attempts have been made to learn a natural image prior using CNNs for image restoration. Kim *et al.* [38] unrolled the AM algorithm and learned the corresponding proximal mapping using the CNN. Similarly, Zhang *et al.* [39] plugged the CNN denoiser into the AM process for image deblurring and super-resolution. Bigdeli *et al.* [40] devised deep mean-shift priors for the natural image distribution. Chang *et al.* [41] replaced the image prior with a convolutional autoencoder, and adopted the adversarial learning [42]. Ulyanov *et al.* [43] showed that a randomly-initialized CNN can be used as an image prior when it is fitted to a single degraded image. All these methods use the CNN for image restoration tasks, aiming at producing a natural image with realistic textures.

III. Proposed Method

A. Background and Motivation

We denote by $\mathbf{f}$ an input image. The subscripts i, j are the location of a pixel. We are interested in decomposing $\mathbf{f}$ into two components $\mathbf{f} \rightarrow \mathbf{u} + \mathbf{v}$ where $\mathbf{u}$ and $\mathbf{v}$ represent structure and texture components, respectively. The general variational framework for image decomposition is given as an energy minimization problem:

$$\min_{(\mathbf{u},\mathbf{v})} \{\lambda H_1(\mathbf{u}) + H_2(\mathbf{v}) : \mathbf{f} = \mathbf{u} + \mathbf{v}\}, \tag{1}$$

where H_1 and H_2 are functional which forces $\mathbf{u}$ and $\mathbf{v}$ to have a desired statistical properties of the solution. That is, if $\mathbf{u}$ is structure image then $H_1(\mathbf{u}) \ll H_2(\mathbf{u})$, and vice-versa. The constant λ is the balancing parameter. The rich literature of this problem can be listed according to choices of both H_1 and H_2. The representative one for H_1 is the total variation (TV) of $\mathbf{u}$, that excludes strong oscillations but permits large-scale edges:

$$H_1(\mathbf{u}) = \sum\nolimits_{i,j} ||(\nabla \mathbf{u})_{i,j}||_2. \tag{2}$$

Using the TV, various $H_1 - H_2$ decomposition models can be formulated, i.e., TV-Δ^{-1} [7], TV$-L_2$ [14], and TV$-L_1$ [29]. Aujol *et. al.* [7] compared these models and concluded that using L_1-norm for H_2 is most suitable for structured texture patterns. Other choices for H_1 include the relative total variation (RTV) [2] and L_0-norm of gradients [3].

Most variational methods for image decomposition employ differences in the brightness values or gradient magnitudes, and assume that the structure image $\mathbf{u}$ has a small norm in some functional spaces H_1. However, we find that the existing methods have limited ability to distinguish between structure edges and fine texture, as shown in Fig. 1[1]. The TV prior [14] often degrades image structures exhibiting blur artifacts (Fig. 1(a) and (b)). Although the RTV [2] effectively preserves the structure edges, it does not completely remove textures that have fine spatial scale, and makes the resulting image flat and

[1]We implemented the TV-L_2 [14] and TV-Δ^{-1} [7] models using the gradient descent with 200 and 400 iterations, respectively.

blocky (Fig. 1(c)). The variational models based on the TV [14] also suffer from global intensity shifting [13], and do not reproduce the original colours.

In what follows, we introduce a new model by learning deep variational priors without explicit $(\mathbf{u}, \mathbf{v})$ training data. The proposed method successfully differentiates high-amplitude details from structure, and preserves shading information as shown in Fig. 1(d).

B. Formulation

Setting H_2 to L_1-norm, we reformulate the constrained problem of (1) using its augmented Lagrangian (AL) function:

$$\begin{aligned}\mathcal{L}_{\mathrm{AL}}(\mathbf{u}, \mathbf{v}, \boldsymbol{\gamma}) = \lambda H_1(\mathbf{u}) + \|\mathbf{v}\|_1 - \boldsymbol{\gamma}^T(\mathbf{f} - (\mathbf{u} + \mathbf{v})) \\ + \tfrac{\beta}{2}\|\mathbf{f} - (\mathbf{u} + \mathbf{v})\|_2^2,\end{aligned} \tag{3}$$

where $\beta > 0$ is a penalty parameter and $\boldsymbol{\gamma}$ is a Lagrange multiplier. That is, the constraint in (1) is realized by augmenting the energy with quadratic penalty function and $\boldsymbol{\gamma}$. The parameter β should be large enough so that we almost have $\mathbf{f} = \mathbf{u} + \mathbf{v}$. We have experimented with various choices of H_2, including Gabor wavelet [7], G-norm [30], and recent convolutional sparse coding [31]. However, there was no significant difference in the results, and L_1-norm is more amenable to efficient implementation.

The minimizers of (1) correspond to the saddle point of $\mathcal{L}_{\mathrm{AL}}$, which can be obtained by using the ADMM algorithm [44]. It consists of the three following iterations:

- $(\mathbf{u}, \boldsymbol{\gamma})$ being fixed, we minimize (3) with respect to $\mathbf{v}$:

$$\mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \frac{\beta^k}{2}\left\|\mathbf{v} - (\mathbf{f} - \mathbf{u}^k - \bar{\boldsymbol{\gamma}}^k)\right\|_2^2 + \|\mathbf{v}\|_1, \tag{4}$$

- $(\mathbf{v}, \boldsymbol{\gamma})$ being fixed, we minimize (3) with respect to $\mathbf{u}$:

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{u}} H_1(\mathbf{u}) + \frac{\beta^k}{2\lambda}\left\|\mathbf{u} - (\mathbf{f} - \mathbf{v}^{k+1} - \bar{\boldsymbol{\gamma}}^k)\right\|_2^2, \tag{5}$$

- $\boldsymbol{\gamma}$-update:

$$\boldsymbol{\gamma}^{k+1} = \boldsymbol{\gamma}^k - \beta^k(\mathbf{f} - \mathbf{u}^{k+1} - \mathbf{v}^{k+1}), \tag{6}$$

where $\bar{\boldsymbol{\gamma}} = \boldsymbol{\gamma}/\beta$, $\beta^{k+1} = \alpha^k\beta^k$, and $\alpha^k > 1$ is the continuation parameter which gradually increases β to the final value. We denote by k the iteration index. Now, the priors on $\mathbf{u}$ and $\mathbf{v}$ are decoupled into two individual subproblems. Since, for a fixed $(\mathbf{u}, \boldsymbol{\gamma})$, all the terms in (4) is separable with respect to $\mathbf{v}_{i,j}$, the $\mathbf{v}$-update step is straightforward:

$$\mathbf{v}^{k+1} = ST(\mathbf{f} - \mathbf{u}^k - \bar{\boldsymbol{\gamma}}^k, 1/\beta^k), \tag{7}$$

where $ST(\cdot)$ denotes the soft-thresholding operator [32]. The last step of (6) is the dual ascent for $\boldsymbol{\gamma}$.

C. Learning Structure Prior H_1

The choice of H_1 in (5) provides a generic prior on structure image $\mathbf{u}$, and is the subject of much interest. In this paper, we parameterize $H_1(\mathbf{u})$ with the CNNs for structure-texture decomposition. How can we do this? The key observations

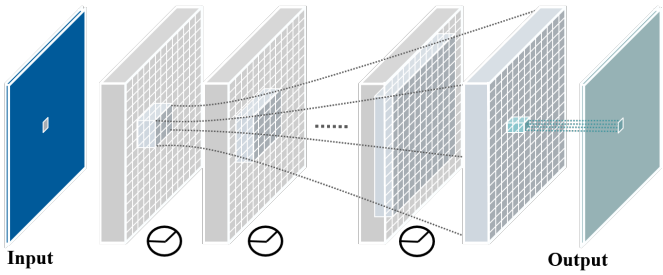


Fig. 2. Schematic illustration of the architecture. We use the multi-scale context aggregation network (CAN) [48] for the projection operator $\mathcal{P}$. The shaded dark blue pixels show the receptive field of each layer. Circles show the ReLU nonlinearity. The CAN uses dilated convolutions with increasing dilation factor in depth, and consequently global information can be aggregated to produce each output pixel.

are as follows: First, the $\mathbf{u}$-update step can be regarded as a denoising step. If we define $\tau^k = \lambda/\beta^k$ and $\hat{\mathbf{u}}^k = \mathbf{f} - \mathbf{v}^{k+1} - \bar{\boldsymbol{\gamma}}^k$, (5) becomes

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{u}} \left\{ H_1(\mathbf{u}) + \frac{1}{2\tau^k}\left\|\mathbf{u} - \hat{\mathbf{u}}^k\right\|_2^2 \right\}. \tag{8}$$

It corresponds to the MAP estimation problem, aiming at cleaning the signal $\hat{\mathbf{u}}^k$ corrupted by additive Gaussian noise of variance τ^k. For example, when we set H_1 as in (2), then (8) is the standard TV denoising problem [14], [32]. Second, many denoising priors have been successfully employed for image decomposition [1], [3], [7]. Finally, for the wide range of H_1, we can deduce the solution of (8) using the Moreau decomposition [45]:

$$\begin{aligned}\mathbf{u}^{k+1} &= \hat{\mathbf{u}}^k - \tau^k\left(I + \tfrac{1}{\tau^k}\partial H_1^*\right)^{-1}\left(\tfrac{\hat{\mathbf{u}}^k}{\tau^k}\right), \\ &= \hat{\mathbf{u}}^k - \tau^k \partial H_1^{\mathrm{e}}(\hat{\mathbf{u}}^k), \\ &= \hat{\mathbf{u}}^k - \mathcal{P}_{\tau^k}(\hat{\mathbf{u}}^k),\end{aligned} \tag{9}$$

where $(\cdot)^{-1}$ is the resolvent of the sub-differential ∂ [45]. H_1^* and H_1^{e} are the Fenchel dual function and the Moreau envelope of H_1, respectively. Note that the last equality has exactly the form of Chambolle's algorithm [18] for TV denoising, equipped with the nonlinear projection operator $\mathcal{P}$[2]. Hence, the solution of (8) at iteration k hinges on computing the projection operator $\mathcal{P}$.

Building upon these observations, we directly learn the projection operator $\mathcal{P}(\cdot, \Theta)$ using the CNN, which provides more flexibility than parameterizing H_1 and then solving the denoising problem of (8). For Gaussian denoising, an ideal projection operator parameterized by Θ would satisfy:

$$\mathcal{P}(\mathbf{f} + \mathbf{n}_\tau, \Theta_\tau) = \mathbf{n}_\tau, \tag{10}$$

where $\mathbf{n}_\tau$ is the zero-mean Gaussian noise of variance τ. We expect that $\mathcal{P}(\cdot, \Theta)$, which will act as the modular part (8) to solve the decomposition problem of (3), exhibits a large

[2]For the one-homogeneous TV function, $\mathcal{P}$ can be computed numerically [18]. It is, however, not the case for general H_1.

capacity to modeling the variational prior $H_1(\mathbf{u})$ with deep architecture.

D. Network Architecture and Training

The projection network $\mathcal{P}$ needs to have large receptive fields to effectively model the global prior $H_1(\mathbf{u})$. Using a deeper architecture [46] or pooling/unpooling [47] is an easy way to ensure large receptive fields. Both methods, however, introduce more parameters to be learned and increase the computational complexity. Instead, we use the multi-scale context aggregation network (CAN)[3] [48] which is originally developed for semantic segmentation. The schematic of the architecture is illustrated in Fig. 2. The CAN [48] employs the dilated convolutions with increasing dilation factor in depth, and thus global information can be aggregated to produce each output pixel. Note that this architecture was also used by Chen *et. al.* [16] to imitate the existing image processing operators, such as tone mapping, dehazing, and image decomposition.

We construct the CAN architecture [48] using 7 convolution layers with 32 intermediate feature maps and kernels of size 3×3. The last convolution layer applies a 1×1 convolution that predicts the final output $\mathbf{n}$. We set the dilation factor of each convolution layer to 1, 1, 2, 4, 8, 16, and 1, respectively (the receptive field of our network is 65×65). The rectified linear unit (ReLU) is used as nonlinearities in all convolution layers, except for the last layer. The intermediate feature maps are symmetrically padded to avoid boundary artifacts. We train the projection network on several $\sqrt{\tau}$ ranging from $[45 : -1 : 1]$ for the continuation scheme (see next subsection for details). The possible outputs of the network are constrained to be between $-\sqrt{\tau}$ and $\sqrt{\tau}$ using a scaled $tanh$ function. We use the L_1 loss function for training:

$$\mathcal{L}_{\text{CAN}}(\Theta_\tau) = \sum_{m=1}^{M} \frac{1}{M} \left\| \mathcal{P}(\mathbf{z}_\tau^{(m)}, \Theta_\tau) - \mathbf{n}_\tau^{(m)} \right\|_1, \qquad (11)$$

where $\mathbf{z}_\tau^{(m)} = \mathbf{f}^{(m)} + \mathbf{n}_\tau^{(m)}$. According to (8) and (10), we synthetically generate the training data $\mathbf{z}_\tau$ by adding $\mathbf{n}_\tau$ to the clean 65×65 patch, sampled from the BSD300 dataset [49]. The total number of training patches M is set to 1.5×10^5.

During decomposition, $(\mathbf{u} - \hat{\mathbf{u}}^k)$ in (8) may not follow a Gaussian distribution exactly. The adoption of $(\mathbf{z}, \mathbf{n})$ as training data is purely based on the formal equivalence between a Gaussian denoising and (8). However, it is reasonable to assume that $\mathbf{u}$ is rarely correlated with $\mathbf{v}$ [7], and the mean value in a local neighborhood of $\mathbf{v}$ is almost zero [50] (recall that we use small patches for training). In Section V, we will demonstrate that the resulting projection network is very effective in separating texture from structure.

E. Decomposition Algorithm and Parameters

The overall procedure for structure and texture image decomposition is summarized in Algorithm 1. We begin by initializing $\mathbf{u}^1$ and letting $\mathbf{v}^1 = \mathbf{f} - \mathbf{u}^1$ (Line 4). The basic

[3]It allows feeding images with arbitrary size without resizing and produces the output, having the same resolution as the input.

Algorithm 1 Structure-Texture Image Decomposition

1: **Input:** $\mathbf{f}$ (an input image)
2: **Parameters:**
λ (a smoothing parameter, $\lambda > 0$)
$\boldsymbol{\gamma}$ (a Lagrangian multiplier, $\boldsymbol{\gamma}^1 = \mathbf{0}$ (zero vector))
c (a common difference for continuation, $c < 0$)
$\mathcal{P}$ (the set of learned projection operators)
K (the maximum number of iterations)
3: **Procedure:** Image Decomposition with $\mathcal{P}$
4: Initialize $\mathbf{u}^1$ and define $\mathbf{v}^1 = \mathbf{f} - \mathbf{u}^1$
5: **for** $k = 1 : K$ **do**
6: (Specify $\sqrt{\tau^k}$, β^k, and $\bar{\boldsymbol{\gamma}}^k$)
7: $\sqrt{\tau^k} = \max(45 + c(k-1), 1)$
8: $\beta^k = \lambda/\tau^k$, $\bar{\boldsymbol{\gamma}}^k = \boldsymbol{\gamma}^k/\beta^k$
9: (Compute $\mathbf{v}^{k+1}$ using the soft-thresholding)
10: $\mathbf{v}^{k+1} = ST(\mathbf{f} - \mathbf{u}^k - \bar{\boldsymbol{\gamma}}^k, 1/\beta^k)$
11: (Compute $\mathbf{u}^{k+1}$ using the projection operators)
12: $\hat{\mathbf{u}}^k = \mathbf{f} - \mathbf{v}^{k+1} - \bar{\boldsymbol{\gamma}}^k$
13: $\mathbf{u}^{k+1} = \hat{\mathbf{u}}^k - \mathcal{P}(\hat{\mathbf{u}}^k, \Theta_{\tau^k})$
14: (Update the Lagrangian multiplier $\boldsymbol{\gamma}^{k+1}$)
15: $\boldsymbol{\gamma}^{k+1} = \boldsymbol{\gamma}^k - \beta^k(\mathbf{f} - \mathbf{u}^{k+1} - \mathbf{v}^{k+1})$
16: **end for**
17: **Output:** $\mathbf{u}^{K+1}$ (structure part) and $\mathbf{v}^{K+1}$ (texture part)

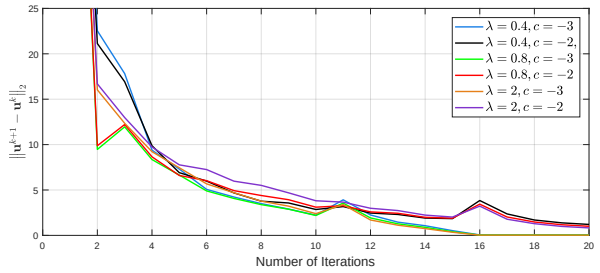
requirement for $\mathbf{u}^1$ is to have the image whose textures are properly smoothed out. However, we do not need to carefully handle the structure edge as it will be restored through the iterations. Two methods are used for initialization: the input image $\mathbf{f}$ itself or the median-filtered version of $\mathbf{f}$. The influence of different initialization for $\mathbf{u}^1$ will be further analyzed in Section IV.

Since training $\mathcal{P}(\cdot, \Theta_\tau)$ for all possible values of $\tau = \lambda/\beta$ is practically impossible, we instead specify $\sqrt{\tau^k}$ in each iteration using an arithmetic progression with common difference $c < 0$, i.e., $\sqrt{\tau^k} = \max(45 + c(k-1), 1)$ (Line 7). This process implicitly determines the penalty parameter $\beta^k = \lambda/\tau^k$ (Line 8), and allows β^k to increase at each iteration (continuation scheme). We always set $\boldsymbol{\gamma}^1$ to $\mathbf{0}$ (zero vector). Now, we have two parameters (λ, c) to adjust. They are pre-defined by the user and fixed during iterations. At each iteration, the algorithm then estimates the texture part $\mathbf{v}^{k+1}$ via the soft-thresholding (Line 10) and the structure part $\mathbf{u}^{k+1}$ using the learned projection network $\mathcal{P}(\cdot, \Theta)$ (Line 12 and 13). The Lagrangian multiplier $\boldsymbol{\gamma}$ is also updated according to (6) (Line 15). Finally, the algorithm is terminated after $k = K$ iterations. The above iterations will not lose structure edges but gradually remove textures. Furthermore, the proposed method is able to separate the repetitive textures while preserving the illuminance of structure layer unchanged, as will be confirmed by our results.

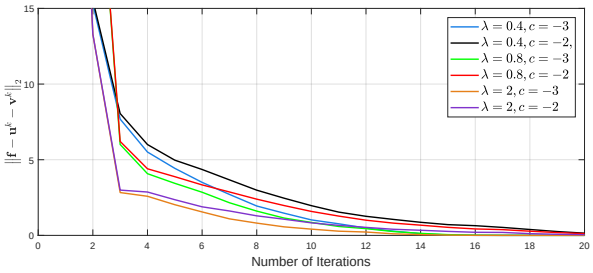
IV. Analysis

In this section, we analyze the properties of the proposed method, including convergence, parameter adjustment, and runtime. The influence of different initializations for large-scale texture is also presented.

(a) input and $\mathbf{u}^1$ (b) $k = 1$ (c) $k = 5$ (d) $k = 10$ (e) Snippets



(f) Sum of intensity difference ($\left\|\mathbf{u}^{k+1} - \mathbf{u}^k\right\|_2$) between iterations



(g) Sum of difference ($\left\|\mathbf{f} - \mathbf{u}^k - \mathbf{v}^k\right\|_2$) between iterations

Fig. 3. (a) input image, (b)-(d) our results at iterations k=1, 3, and 5, (e) Snippets of structure images obtained by RegionCov [26], BTF [9], RTV [2], and the proposed method. Although the convergence behavior of Algorithm 1 is not monotone, we can see that it converges to a fixed point within few iterations (f). During iterations, Algorithm 1 automatically satisfies the constraint in (1) as in (g). We initialize $\mathbf{u}^1$ with $\mathbf{f}$.

A. Convergence

In this subsection, we analyze the convergence properties of Algorithm 1. An important feature of the ADMM algorithm [44] is its modular structure which was established in [51]. Based on the theorems in [51], we can show that, as $k \to \infty$, the iterates of Algorithm 1 demonstrates a fixed point convergence with the bounded projection networks:

$$\|\mathcal{P}(\mathbf{u}, \Theta_\tau)\|_2^2 / N \leq \tau C, \tag{12}$$

where N is the total number of pixels and C is some universal constant. Note that the condition of (12) is automatically satisfied by the last *tanh* function in our network. The proof is straightforward from Theorem 1 in [51] and we omit the details here. Intuitively, the condition of (12) states that Lines 10 and 13 of Algorithm 1 become the identity mapping as $k \to \infty$ and $\tau \to 0$ (continuation). Hence, Algorithm 1 asymptotically converges to a fixed point, i.e., $\left\|\mathbf{u}^{k+1} - \mathbf{u}^k\right\|_2 \to 0$ and $\left\|\mathbf{v}^{k+1} - \mathbf{v}^k\right\|_2 \to 0$.

In practice, Algorithm 1 is terminated early to reduce the runtime. Figure 3 shows how the differences ($\left\|\mathbf{u}^{k+1} - \mathbf{u}^k\right\|_2$ and $\left\|\mathbf{f} - \mathbf{u}^k - \mathbf{v}^k\right\|_2$) evolve over iterations with various (λ, c). We normalize the intensity range to $[0, 1]$ and an input image is shown in Fig. 3(a). Although the convergence behavior of $\mathbf{u}$ is not monotone, we can see that it converges to a fixed point within a few iterations (Fig. 3(f)). The smaller value of $|c|$ yields a slow convergence rate. However after a few iterations (typically from 10 to 15), Algorithm 1 gives almost the same results which are hard to distinguish by human eyes. The average value of the per-pixel intensity differences is 3.4×10^{-6} after 10 iterations. During iterations, Algorithm 1 gradually extracts structure from texture (Fig. 3(a)-(d)), and satisfies the constraint in (1) as shown in Fig. 3(g).

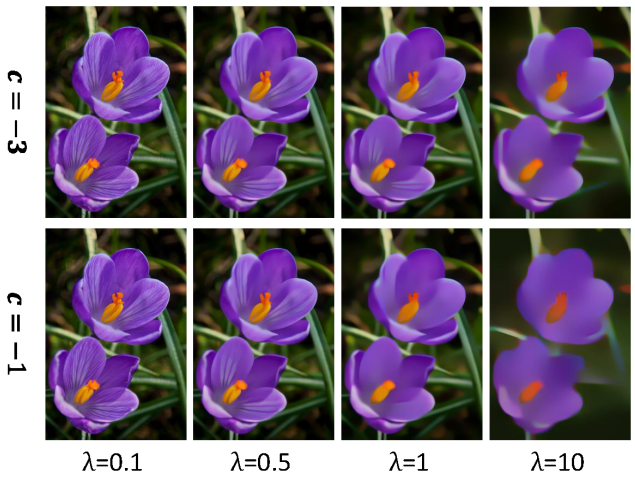
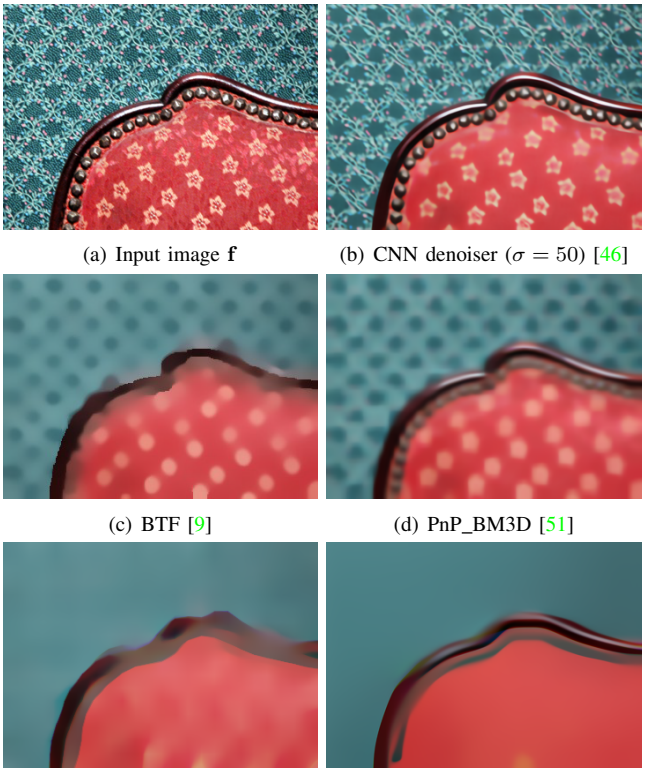


Fig. 4. The sturcture image $\mathbf{u}$ with various choices of λ and c. The larger value of $|c|$ makes the subsequent projection network weaken faster, yielding less smoothing effects. The proposed method can produce continuous smoothing effects by adjusting λ and c. This is a noticeable difference with respect to the existing data-driven schemes [16], [17].

B. Adjusting Degree of Smoothing

The balancing parameter λ is empirically used to adjust degree of smoothing in variational frameworks [2], [7]. Similarly, the proposed method can work with λ that controls the



(a) Input image $\mathbf{f}$ (b) CNN denoiser ($\sigma = 50$) [46]

(c) BTF [9] (d) PnP_BM3D [51]

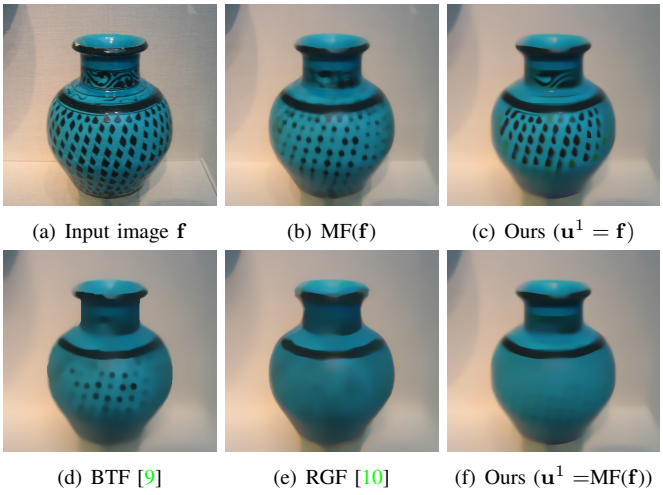
(e) RGF [10] (f) Ours with $\mathbf{u}^1 = \mathbf{f}$

Fig. 5. Decomposition results on the image containing multiple-scale textures: (a) input image, (b) CNN denoiser [46], (c) BTF [9], (d) PnP_BM3D [51], (e) RGF [10], and (f) the proposed method. (a) and (c) are taken from [9]. The result of RGF (c) [10] is obtained with $\sigma_s = 12$ and $\sigma_r = 0.05$. Although the projection network is trained with additive gaussian noises, it effectively extracts multiple-scale texture and preserves structure edges.

relative strength of the deep structure prior $H_1(\mathbf{u})$. Further, it provides another degree of freedom, i.e, the continuation parameter c. The larger value of $|c|$ makes the subsequent projection network weaken faster, yielding less smoothing effects (and vice-versa). In other words, the large $|c|$ quickly shrinks the projection radius of (12), and thus texture parts will appear much more in the structure image. We show the resulting structure image $\mathbf{u}$ with various choices of λ and c in Fig. 4. The number of iterations K is fixed to 10 and $\mathbf{u}^1$ is initialized with the input image itself for all cases. Figure 4 demonstrates that by varying λ or c, we can adjust the degree of smoothing using a single set of projection operators. This is a noticeable difference with the existing data-driven smoothing schemes [16], [17]. For example, we do not require the bi-cubic interpolation of results generated with similar smoothing parameters [17], or additional input channels to alter the behavior for networks [16].

C. Handling Large Variations inside Texture Region

For the projection network, we have generated the training data by adding additive Gaussian noise. This can be regarded as adding a form of small-scale texture to images and training the network to extract this information. Hence, the proposed method may fail to preserve structure edges when multiple-scale or extremely varying textures should be removed. In



(a) Input image $\mathbf{f}$ (b) MF($\mathbf{f}$) (c) Ours ($\mathbf{u}^1 = \mathbf{f}$)

(d) BTF [9] (e) RGF [10] (f) Ours ($\mathbf{u}^1$ =MF($\mathbf{f}$))

Fig. 6. Decomposition results on the image containing extremely varying textures: (a) input image, (b) 11×11 median filtered (MF) version of $\mathbf{f}$, (c) ours with $\mathbf{u}^1 = \mathbf{f}$, (d) BTF [9], (e) RGF [10], and (f) ours with $\mathbf{u}^1$ =MF($\mathbf{f}$). The input (a) and result of BTF (d) are taken from [9]. The result of RGF (c) [10] is obtained with $\sigma_s = 15$ and $\sigma_r = 0.06$. The proper initialization of $\mathbf{u}^1$ allows handling large-scale textures.

Fig. 5, we first apply our method to the mixture of large-scale textures, and compare the results with the CNN denoiser ($\sigma = 50$) [46], BTF [9], PnP_BM3D ($\lambda = 10$, $\rho_0 = 1$) [51], and RGF [10]. The result of BTF is taken from the original paper [9]. We use ($\lambda = 3.5$, $c = -1$, $\mathbf{u}^1 = \mathbf{f}$) and ($\sigma_s = 12$, $\sigma_r = 0.05$) for the proposed method and RGF [10], respectively. Applying the CNN denoiser [46] directly to image decomposition does not produce satisfactory results (Fig. 5(b)). The BTF [9], PnP_BM3D [51], and RGF [10] have difficulty in distinguishing between obscure borders and multi-scale textures. In contrast, our projection network can be generalized to multiple scale of textures, and can preserve structure edges and corners. Furthermore, our method effectively restores the shading on the object surface, compared to [9], [10] (Fig. 5(c) and (e)).

Next, we apply the proposed method to the image containing extreme variations inside a texture region in Fig. 6. Our method has trouble when it is initialized with the input image itself, as shown in Fig. 6(c). We set λ and c to 1.5 and -1, respectively. Using the larger value of $\lambda = 1.5$ overblurs some of the structure edges. We also observe that the BTF [9] has difficulty with extreme variations inside a texture region (Fig. 6(d)), and the RGF [10] erodes the main structure edges (Fig. 6(e)). Note that the minimization problem of (3) may be nonconvex since it is characterized by the deep projection network. Thus, different initializations for $\mathbf{u}^1$ will make Algorithm 1 converges to different fixed points. For extremely varying textures, we compute the initial $(\mathbf{u}^1, \mathbf{v}^1)$ pair by using the median filter (Fig. 6(b)). Existing techniques for structure-preserving smoothing can be employed here, but at unnecessary costs and yield little difference. The proper initialization of $\mathbf{u}^1$ enables the proposed method to handle extreme variations inside texture regions as shown in Fig. 6(f). We empirically found that, in this case, 5×5 to 11×11 median filter is a good choice for the initialization.

(a) Input RGB (b) RTV [2] (c) BTF [9] (d) JCAS [31] (e) Ours

Fig. 7. Comparsions of the texture removal: (From left to right) input RGB image, RTV [2], BTF [9], JCAS [31], and the proposed method. The regions of red boxes are highlighted at the bottom of each results. We set all parameters through extensive experiments for the best qualitative performance. Our method effectively preserves surface shading (top) and does not suffer from staircasing artifacts (bottom). Best viewed in colour.

D. Runtime

The main computational cost of Algorithm 1 is incurred from the deep projection network which is comprised of 7 convolution layers. The other procedures can be implemented efficiently by point-wise operations. We have implemented the projection network[4] using the MatConvnet library [52], and have tested on 12GB NVIDIA GeForce GTX Titan and cuDNN v7. The storage capacity of the deep projection network is 0.15MB for each $\sqrt{\tau}$. We measure the run time of the proposed method using *gputimeit* function built in MATLAB. In this setting, the projection network takes about 0.04 seconds per iteration for a color image of size 500×600. It operates in a fully convolution manner, allowing reuse of computations between overlapping patches. In total, running $K = 10$ iterations of Algorithm 1 with $\mathbf{u}^1 = \mathbf{f}$ takes about 0.55 seconds for the same image size. In our GPU implementation, Algorithm 1 has a time complexity linear to an image size.

V. Application

We now showcase several applications including texture removal, detail manipulation, HDR tone-mapping, and non-photorealistic abstraction to demonstrate the effectiveness of our approach. All the following applications share the same model as well as the run-time. The learned models and source code will be made publicly available at the project website[5]. In each application, we compare the proposed method to the current state-of-the-art methods. The results for the comparison were obtained from the source codes provided by the authors. All the parameters were carefully tuned to yield the best performance through extensive experiments.

[4]We use the random initialization using Gaussian distributions for the convolution layers

[5]http://diml.yonsei.ac.kr/∼deepImDecomp.

TABLE I

Average correlation value between $\mathbf{u}$ and $\mathbf{v}$ components on the RTV dataset [2]. The parameters for each method are fixed on all the 200 images. L_0 smoothing [3] ($\lambda = 0.08$), RTV [2] ($\sigma_s = 5, \sigma_c = 0.1$), RGF [10] ($\sigma_s = 3, \sigma_r = 0.05$), SD filter [34] ($\mu = 5, \nu = 40, \lambda = 10^2$), JCAS [31] ($\alpha = 0.05, \gamma = 0.2$), and ours ($\lambda = 1, c = -3$). The proposed method archieves the lowest correlation values, satisfying the orthogonality assumption between $(\mathbf{u}, \mathbf{v})$ very well.

corr($\mathbf{u}, \mathbf{v}$)					
L_0 smoothing [3]	RTV [2]	RGF [10]	SD filter [34]	JCAS [31]	Ours
0.2072	0.0907	0.0455	0.0595	0.0633	**0.0402**

A. Texture Removal

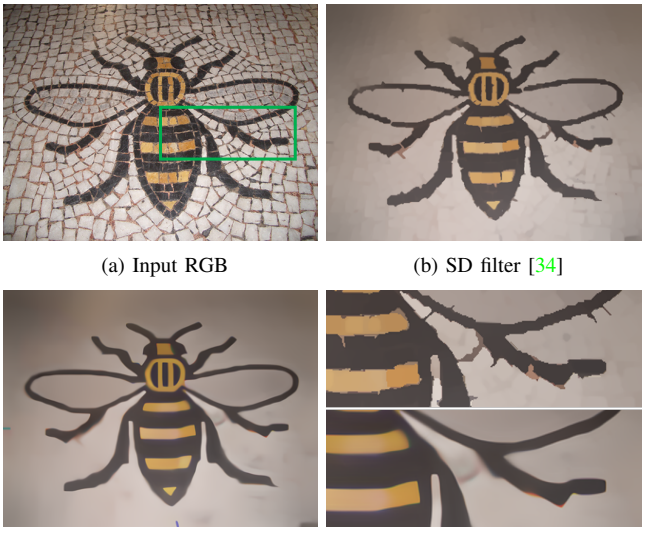
Our method can be directly applied to texture removal that is useful for various tasks such as image analysis, composition, and inverse halftoning. The tunable parameters c and $\boldsymbol{\gamma}^1$ are fixed to -3 and $\mathbf{0}$ (zero vector), respectively. We alter the scale of texture to be removed by adjusting the balancing parameter $\lambda \in [0.1, 5]$. In this setting, we apply $K = 10$ iterations of Algorithm 1 for all the results in this paper. We use the test images that are completely separated from the training set, i.e., BSD300 dataset [49]. Examples of texture removal comparing with the RTV [2], BTF [9], and JCAS [31] are shown in Fig. 7. Most methods generally succeeded in extracting prominent image structures while removing textures. The RTV [2] optimizes a non-convex energy function, resulting

(a) Input RGB

(b) RGF [10]

(c) Ours

Fig. 8. More results on the texture removal: (Top) input RGB image, (Middle) RGF [10], and (Bottom) the proposed method. Our method effectively preserves directional structure edges and does not produce spurious regions. Best viewed on the electronic version.



(a) Input RGB (b) SD filter [34]

(c) Ours (d) Snippets

Fig. 9. Examples of the texture removal for fine textures. The input image is taken from [9].

in the iterative WLS smoothing [1], where the weights are computed from the structure image estimated in the previous iteration. The use of non-convex norm [2] prevents the global intensity of structure image to be shifted [13]. However, the RTV [2] smooths some structural edges, and cannot perfectly eliminate textures that have similar scale and appearance to the underlying structure image (Fig. 7(b)). The BTF [9] describes the representative texture by finding the patch that is least likely to contain a prominent structures. Its likelihood is simply computed based on the assumption that texture component has smaller amplitude than the neighboring structure edge. Thus, the BTF [9] has difficulty in removing fine-scale oscillations that have high-contrast (Fig. 7(c)). The JCAS [31] incorporates the TV prior [7] as a structure norm, and models the texture component as convolutional synthesis dictionary. However, this method can not overcome the inherent limitations of TV prior [7]. It suffers from staircasing artifacts and global intensity shifting [13], as shown in Fig. 7(d). Comparisons with the RGF [10] are made in Fig. 8. The RGF [10] controls the scale of texture to be removed using the Gaussian kernel, and thus does not protect corners and/or directional edges (Fig. 8(b)). In contrast, the proposed method removes texture without notable artifacts, and preserves important structures and original colours (Figs. 7(e) and 8(c)). Using the fine texture image, we compare the result of SD filter [34] in Fig. 9. We use the Gaussian filter with standard deviation 2 to obtain the static guidance image, as suggested in [34]. $(\mu,\ \nu,\ \lambda)$ are set to $(5,\ 40,\ 2\times10^3)$, respectively. The SD filter [34] extracts the image structure while filtering out fine textures. However, it smooths out object boundaries (feeler in Fig. 9(b)), and shows spurious directional details (wings and legs in Fig. 9(b) and (d)). Contrarily, our method maintains sharp object boundaries without artifacts, as shown in Fig. 9(c) and (d).

There are no invariably adopted evaluation method for texture removal. Following the assumption that the structure

(a) WLS [1] (b) JCAS [31] (c) BTF [9] (d) Ours

Fig. 10. Examples of the detail and contrast manipulation in LAB color space: (a) WLS [1], (b) JCAS [31], (c) BTF [9], and (d) ours. The WLS [1] and JCAS [31] tend to produce smooth structure componentss. The averaging process of BTF [9] is often done to sharpen the input step. Boosting from these components causes halo and gradient reversal artifacts along discontinuities. Best viewed by zooming in on the electronic copy.

(a) Input LDR RGB (b) L_0 smoothing [3] (c) JCAS [31] (d) Ours

Fig. 11. Examples of the HDR tone-mapping in log luminance channel: (a) input low dynamic range (LDR) RGB, (b) L_0 smoothing [3], (c) JCAS [31], and (d) ours. The L_0 smoothing [3] shows gradient reversal artifacts along strong edges and the JCAS [31] loses significant image details. Contrary, the proposed method generates high-quality tone-mapped images with less artifacts.

and the texture of an image are not correlated, we measure the correlation between resulting $(\mathbf{u}, \mathbf{v})$ components:

$$\mathrm{corr}(\mathbf{u}, \mathbf{v}) = \frac{\mathrm{cov}(\mathbf{u}, \mathbf{v})}{\sqrt{V(\mathbf{u})V(\mathbf{v})}}, \tag{13}$$

where cov and V denote the covariance and the variance, respectively. Note that the correlation criterion was used in [7] for automatically selecting the balancing parameter λ. We use 200 image in the RTV dataset [2], and report the average value of $\mathrm{corr}(\mathbf{u}, \mathbf{v})$ in Table I. The parameters for each method are fixed on all the 200 images. From Table I, one can see that the proposed method archieves the lowest correlation values, satisfying the orthogonality assumption very well.

B. Detail Manipulation

We follow the implementation of WLS [1] in the LAB color space for detail manipulation. Given an input image, we fisrt decompose the lightness channel into base and detail layers, i.e., $(\mathbf{u}, \mathbf{v})$ components by applying Algorithm 1. The decomposed layers are then manipulated with a sigmoid curve (appropriately normalized and shifted) to avoid the hard clipping when the detail layer is boosted.

Figure 10 shows an example of detail enhancement from WLS [1], JCAS [31], BTF [9], and the proposed method. One challenging issue in edge-aware manipulations, which process the layered signal, is annoying artifacts at image discontinuities. If decomposition is done in a way to smooth the image edges, the result of detail enhancement will overshoot, causing to halo artifacts. This problem often arises when the structure prior penalizes large gradients (WLS [1] and JCAS [31], Fig. 10(a)-(b)). Conversely when the image edges are sharpened[6], the result shows gradient reversal artifacts exhibiting unwanted profiles around edges (BTF [9], Fig. 10(c)). The proposed method does not produce such artifacts, as shown in Fig. 10(d). We set c and λ to -3 and 0.8, respectively.

C. HDR Tone-mapping

HDR tone-mapping is concerned with compressing the intensity of a high dynamic range image while maintaining details. Our implementation follows the previous work of Durand and Dorsey [21], but replaces the BT filter-based [8] decomposition with Algorithm 1 and other baseline methods. The luminance image is first computed using $\mathbf{l}^{\mathrm{in}} = (0.299\mathbf{r}^{\mathrm{in}} + 0.587\mathbf{g}^{\mathrm{in}} + 0.114\mathbf{b}^{\mathrm{in}})$. We then decompose the logarithm of luminance $\log(\mathbf{l}^{\mathrm{in}})$ into base and detail components. Since our projection network is trained on the intensity range of $[0, 1]$, we normalize $\log(\mathbf{l}^{\mathrm{in}})$, and scale the output after decomposition to match the min-max intensity values to be same as those of $\log(\mathbf{l}^{\mathrm{in}})$. The base component is compressed with a factor

[6]This frequently happens when applying the weighted-average filters [8] or the variational minimization including non-convex priors [3].

TABLE II
QUANTITATIVE COMPARISON OF HDR TONE-MAPPING USING TMQI INDEX [53] (HIGHER VALUE IS BETTER). WE USE 15 HDR IMAGES PROVIDED IN [53]. THE PARAMETERS FOR EACH METHOD ARE FIXED FOR ALL THE 15 HDR IMAGES. PARAMETERS: L_0 SMOOTHING [3] ($\lambda = 0.2$), JCAS [31] ($\lambda = 1e^{-3}, \gamma = 1e^{-4}$), AND OURS ($\lambda = 0.8, c = -3$).

Sequence	L_0 smoothing [3]	JCAS [31]	Ours
1	0.9375	0.9207	**0.9506**
2	0.8827	0.8882	**0.8991**
3	**0.9558**	0.9504	0.9553
4	0.9620	0.9722	**0.9742**
5	0.9704	**0.9721**	0.9548
6	0.8559	0.8758	**0.8961**
7	0.8746	**0.8797**	0.8691
8	0.8981	**0.9128**	0.9043
9	0.9436	0.9532	**0.9818**
10	0.9273	0.9264	**0.9516**
11	0.9578	0.9604	**0.9836**
12	0.9548	0.9688	**0.9737**
13	0.9485	0.9549	**0.9663**
14	0.9088	0.9174	**0.9354**
15	0.9628	**0.9690**	0.9546
Average	0.9293	0.9348	**0.9433**

of 0.4, and added back to the detail component. Given the output luminance $\mathbf{l}^{\text{out}}$, we reproduce chrominance information as follows [54]:

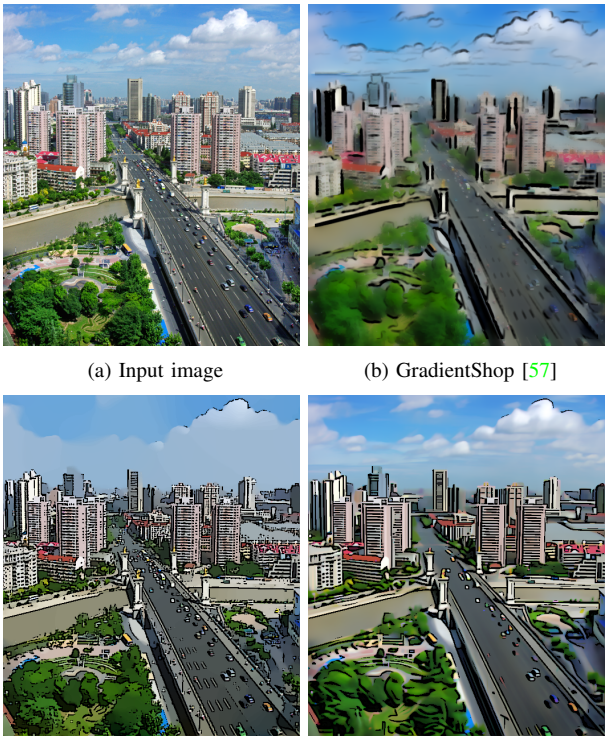
$$\mathbf{p}^{\text{out}} = \left(\frac{\mathbf{p}^{\text{in}}}{\mathbf{l}^{\text{in}}}\right)^{s} \mathbf{l}^{\text{out}}, \tag{14}$$

for $\mathbf{p} = \mathbf{r}, \mathbf{g}$, or $\mathbf{b}$. the exponent s controls the color saturation (gamma correction). Finally, the output image is clamped to $[0, 1]$ for display.

We evaluate the proposed method with state-of-the-art tone mapping methods on 15 HRD images provided in [53]. The competing algorithms include the L_0 smoothing introduced in [3] and recently proposed JCAS [31] adopting the convolutional sparse coding for texture components. The tunable parameters for each method including ours are fixed on all the 15 images: L_0 smoothing [3] ($\lambda = 0.2$) and JCAS [31] ($\lambda = 1e^{-3}, \gamma = 1e^{-4}$). For Algorithm 1, we set the balancing and continuation parameter (λ, c) to 0.8 and -3, respectively. The tone-mapped image quality index (TMAI) [53] is used to assess the quality of tone-mapped images quantitatively. Table II summarize the results of quantitative evaluation on 15 HDR images [53]. This table demonstrates that the proposed method shows the highest TMQI value on most images (10 out of 15 images), and the average value is higher than other methods with large margin. Figure 11 shows visual examples of tone-mapped images. The result of [3] presents gradient reversal artifacts along the strong edge (see the book in Fig. 11(b)). The JCAS loses significant details and shows halo artifact, e.g. the lamp in highlighted region (Fig. 11(c)). In contrast, the proposed method generates high-quality tone-mapped images with natural color and less artifacts (Fig. 11(d)).

D. Non-photorealistic Abstraction

Our decomposition result fits non-photorealistic abstraction with simultaneous edge highlighting and detail suppressing.



(a) Input image (b) GradientShop [57]

(c) L_0 smoothing [3] (d) Our results

Fig. 12. Example of non-photorealistic abstraction in RGB color space: (a) input image, (b) GradientShop [57], (c) L_0 smoothing [3], and (d) our abstraction results. We process an input RGB image with $\lambda = 0.3$ and $c = -3$ to obtain a structure component, and then overlay this with DoG edges.

Traditional methods [23], [55] follow two main steps, i.e., image decomposition using the BF [8] or mean-shift [56] and edge extraction with difference-of-Gaussian (DoG). The extracted edges are enhanced and overlapped with structure component to give a non-photorealistic abstraction effect. We apply Algorithm 1 to an input RGB image with parameters $\lambda = 0.3$ and $c = -3$ to obtain structure image, and then overlay this with its DoG edges. Figure 12 shows an example of non-photorealistic abstraction by GradientShop [57], L_0 smoothing [3], and ours. Our model is comparable to the state-of-the-art methods. It augments the distinctiveness of different regions, producing visually striking effects.

VI. CONCLUSION

We have explored a new framework for structure-texture image decomposition, which takes advantages of both the variational framework and recent data-driven approach. Contrary to the classical formulations forcing a structure image into the space of bounded variation, we proposed a deep variational prior characterized by deep CNNs. The network configuration is inspired from the Chambolle's projection method for the TV minimization, and its parameters are trained without explicit structure-texture pairs as super-vision. A fast alternating minimization algorithm and its modular structure are then adopted to plug deep variational priors into an iterative smoothing process. We showed that the CNNs can replace the TV prior, and are indeed powerful to capture textures. The proposed

method successfully differentiates high-amplitude details from structure edges, and converges to a fixed point within few iterations. Experimental results demonstrated the effectiveness of our approach in a great variety of image processing and computational photography applications.

REFERENCES

[1] 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.

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

[3] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L_0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, 2011.

[4] M. Cheng, J. Warrell, W. Lin, S. Zheng, V. Vineet, and N. Crook, "Efficient salient region detection with soft image abstraction," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1529-1536, Dec. 2013.

[5] W. Casaca, A. Paiva, E. Nieto, and P. Joia, "Spectral image segmentation using image decomposition and inner product-based metric," *J. Math. Imaging Vis.* , vol. 45, no. 3, pp. 227-238, 2013.

[6] J. Jeon, S. Cho, X. Tong, and S. LEE, "Intrinsic image decomposition using structure-texture separation and surface normals," in *Proc. Eur. Conf. Comput. Vis.*, pp. 218-233, 2014.

[7] J. Aujol, G. Gilboa, T. Chen and S. Osher, "Structure-texture image decomposition: modeling, algorithms, and parameter selection," *Int. J. Comput. Vis.*, vol. 67, no. 1, pp. 111-136, 2006.

[8] C. Tomasi and R. Manduchi, "Bilateral filtering fro gray and color images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1998.

[9] H. Cho, H. Lee, H. Kang, and S. Lee, "Bilateral texture filtering," *ACM Trans. Graph.*, vol. 33, no. 4, 2014.

[10] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *Proc. Eur. Conf. Comput. Vis.*, 2014.

[11] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu, "Constant time weighted median filtering for stereo matching and beyond," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 49-56, Dec. 2013.

[12] M. Kass and J. Solomon, "Smoothed local histogram filters," *ACM Trans. Graph.*, vol. 29, no. 4, 2010.

[13] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. Eur. Conf. Comput. Vis.*, 2010.

[14] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259-268, 1992.

[15] S. Bi, X. Han, and Y. Yu, "An L_1 transform for edge-preserving smoothing and scene-level intrinsic decomposition," *ACM Trans. Graph.*, vol. 34, no. 4, 2015.

[16] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with fully convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017.

[17] L. Xu, J. Ren, Q. Yan, R. Liao, and J. Jia, "Deep edge-aware filters," in *Proc. Int. Conf. Mach. Learning*, 2015.

[18] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imaging. Vis.*, vol. 20, no. 1, pp. 89-97, 2004.

[19] S. Smith and J. Brady, "Susan-a new approach to low level image processing," *Int. J. Comput. Vis.*, vol. 23, 1995.

[20] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Trans. Image Process.*, vol. 11, no. 10, pp. 1141-1151, 2002.

[21] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Graph.*, vol. 21, no. 34, 2002.

[22] C. Liu, W.T. Freeman, R. Szeliski, and S.B. Kang, "Noise estimation from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[23] H. Winnemoller, S.C. Olsen, and B. Gooch, "Real-time video abstraction," *ACM Trans. Graph.*, vol. 25, no. 3, 2006.

[24] E.S.L. Gastal and M.M. Oliveira, "Adaptive manifolds for real-time high-dimensional filtering," *ACM Trans. Graph.*, vol. 31, no. 4, 2012.

[25] E.S.L. Gastal and M.M. Oliveira, "Domain transform for edge-aware image and video processing," *ACM Trans. Graph.*, vol. 30, no. 4, 2011.

[26] L. Karacan, E. Erdem, and A. Erdem, "Structure-preserving image smoothing via region covariances," *ACM Trans. Graph.*, vol. 32, no. 6, 2013.

[27] L. Bao, Y. Song, Q. Yang, H. Yuan, and G. Wang, "Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 555-569, 2014.

[28] J. Jeon, H. Lee, H. Kang, and S. Lee, "Scale-aware structure-preserving texture filtering," *Pacific Graph.*, vol. 35, no. 7, 2016.

[29] S. Aliney, "A property of the minimum vectors of a regularizing functional defined by means of the absolute norm," *IEEE Trans. Signal Process.*, vol. 45, no. 4, pp. 913-917, 1997.

[30] Y. Meyer, "Oscillating patterns in image processing and in some nonlinear evolution equations," in *The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*, 2001.

[31] S. Gu, D. Meng, W. Zuo, and L. Zhang, "Joint convolutional analysis and synthesis sparse representation for single image layer separation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017.

[32] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Imag. Sci.*, vol. 1, no. 3, pp. 248-272, 2008.

[33] S. Ono, "L_0 Gradient Projection," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1554-1564, 2017.

[34] B. Ham, M. Cho, and J. Ponce, "Robust guided image filtering using non-convex potentials," in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 192-207, 2018.

[35] X. Guo, Y. Li, and J. Ma, "Mutually guided image filtering," in *Proc. ACM on Multimedia*, 2017.

[36] Q. Yang, S. Wang, and N. Ahuja, "Svm for edge-preserving filtering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010.

[37] S. Liu, J. Pan, and M.H. Yang, "Learning recursive filters for low-level vision via a hybrid neural network," in *Proc. Eur. Conf. Comput. Vis.*, 2016.

[38] Y. Kim, H. Jung, D. Min, and K. Sohn, "Deeply aggregated alternating minimization for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[39] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[40] S. Bigdeli, M. Jin, P. Favaro, and M. Zwicker, "Deep mean-shift priors for image restoration," in *Advances in Neural Information Processing Systems*, 2017.

[41] J. Chang, C. Li, B. Poczos, B. Kumar, and A. Sankaranarayanan, "One network to solve them all — solving linear inverse problems using deep projection models," in *Proc. IEEE Int. Conf. on Comput. Vis.*, 2017.

[42] I. Goodfellow, J. Abadie, M. Mirza, B. Xu, D. Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014.

[43] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018.

[44] M. V. Afonso, J. M. Dias, and A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345-2356, 2010.

[45] N. G. Polson, J. G. Scott, and B. T. Willard, "Proximal algorithms in statistics and machine learning," *Statistical Science*, vol. 30, no. 4, pp. 559-581, 2015.

[46] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142-3155, 2017.

[47] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015.

[48] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learning Represent.*, 2016.

[49] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2001.

[50] S. Cho, H. Lee, and S. Lee, "Image decomposition using deconvolution," in *Proc. IEEE Int. Conf. Image. Process.*, 2010.

[51] S.H. Chan, X. Wang, and O. Elgendy, "Plug-and-play ADMM for image restoration: fixed-point convergence and applications," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 84-98, 2017.

[52] http://www.vlfeat.org/matconvnet.

[53] H. Yeganeh and Z. Wang, "Objective quality assessment of tone-mapped images," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 657-667, 2013.

[54] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," *ACM Trans. Graph.*, vol. 21, no. 3, 2002.

[55] D. Decarlo and A. Santella, "Stylization and abstraction of photographs," *ACM Trans. Graph.*, vol. 21, no. 3, 2002.

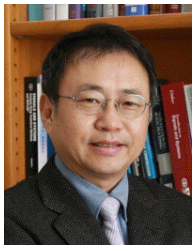
[56] D. Comaniciu and P. Meer, "Mean shift analysis and applications," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999.

[57] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless, "Gradient domain high dynamic range compression," *ACM Trans. Graph.*, vol. 29, no. 2, 2010.

Youngjung Kim (M'18) is a Senior Researcher of Institute of Defense Advanced technology Research (IDAR) at Agency for Defense Development (ADD) in Daejeon, South Korea. He received the B.S. and Ph.D. degrees in Electrical and Electronic Engineering from Yonsei University in 2013 and 2018, respectively. His current research interests include variational method, continuous optimization, and machine learning, both in theory and applications in image processing and computer vision.

Bumsub Ham (M'13) is an an Assistant Professor of Electrical and Electronic Engineering at Yonsei University in Seoul, Korea. He received the B.S. and Ph.D. degrees in Electrical and Electronic Engineering from Yonsei University in 2008 and 2013, respectively. From 2014 to 2016, he was Post-Doctoral Research Fellow with Willow Team of INRIA Rocquencourt, École Normale Supérieure de Paris, and Centre National de la Recherche Scientifique. His research interests include computer vision, computational photography, and machine learning, in particular, regularization and matching, both in theory and applications.

Minh N. Do (M'01, SM'07, F'14) was born in Vietnam in 1974. He received the B.Eng. degree in computer engineering from the University of Canberra, Australia, in 1997, and the Dr.Sci. degree in communication systems from the Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland, in 2001. Since 2002, he has been on the faculty at the University of Illinois at Urbana-Champaign (UIUC), where he is currently a Professor in the Department of Electrical and Computer Engineering, and hold joint appointments with the Coordinated Science Laboratory, the Beckman Institute for Advanced Science and Technology, the Advanced Digital Sciences Center, the Department of Bioengineering, and the Department of Computer Science. His research interests include signal processing, computational imaging, geometric vision, and data science. He received a Silver Medal from the 32nd International Mathematical Olympiad in 1991, a University Medal from the University of Canberra in 1997, a Doctorate Award from the EPFL in 2001, a CAREER Award from the National Science Foundation in 2003, and a Young Author Best Paper Award from IEEE in 2008. He was named a Beckman Fellow at the Center for Advanced Study, UIUC, in 2006, and received of a Xerox Award for Faculty Research from the College of Engineering, UIUC, in 2007. He was a member of the IEEE Signal Processing Theory and Methods Technical Committee, Image, Video, and Multidimensional Signal Processing Technical Committee, and an Associate Editor of the IEEE Transactions on Image Processing. He is a Fellow of the IEEE for contributions to image representation and computational imaging. He was a co-founder and CTO of Personify Inc., a spin-off from UIUC to commercialize depth-based visual communication.



Kwanghoon Sohn (M'92-SM'12) received the B.E. degree in electronic engineering from Yonsei University, Seoul, Korea, in 1983, the M.S.E.E. degree in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, in 1985, and the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, NC, USA, in 1992. He was a Senior Member of the Research engineer with the Satellite Communication Division, Electronics and Telecommunications Research Institute, Daejeon, Korea, from 1992 to 1993, and a Post-Doctoral Fellow with the MRI Center, Medical School of Georgetown University, Washington, DC, USA, in 1994. He was a Visiting Professor with Nanyang Technological University, Singapore, from 2002 to 2003. He is currently an Underwood Distinguished Professor with the School of Electrical and Electronic Engineering, Yonsei University. His research interests include 3D image processing and computer vision.