Non-Local Sparse and Low-Rank Regularization for Structure-Preserving Image Smoothing

Lei Zhu \(^1\), Chi-Wing Fu \(^1\), Yueming Jin \(^1\), Mingqiang Wei \(^2\), Jing Qin \(^3\) and Pheng-Ann Heng \(^1\)

\(^1\)The Chinese University of Hong Kong, \(^2\)Hefei University of Technology, \(^3\)The Hong Kong Polytechnic University

Abstract

This paper presents a new image smoothing method that better preserves prominent structures. Our method is inspired by the recent non-local image processing techniques on the patch grouping and filtering. Overall, it has three major contributions over previous works. First, we employ the diffusion map as the guidance image to improve the accuracy of patch similarity estimation using the region covariance descriptor. Second, we model structure-preserving image smoothing as a low-rank matrix recovery problem, aiming at effectively filtering the texture information in similar patches. Lastly, we devise an objective function, namely the weighted robust principle component analysis (WRPCA), by regularizing the low rank with the weighted nuclear norm and sparsity pursuit with \(L_1\) norm, and solve this non-convex WRPCA optimization problem by adopting the alternative direction method of multipliers (ADMM) technique. We experiment our method with a wide variety of images and compare it against several state-of-the-art methods. The results show that our method achieves better structure preservation and texture suppression as compared to other methods. We also show the applicability of our method on several image processing tasks such as edge detection, texture enhancement and seam carving.

1. Introduction

Structure-preserving image smoothing [XYXJ12, KEE13, CLKL14] aims at decomposing a single input image into prominent structures and fine-scale texture details. This often relies on measuring the size of local contrast, where structures are usually identified as pixels with relatively large contrast [CLKL14]. While it is a very challenging problem, it is highly beneficial to a wide variety of image manipulation applications, such as tone mapping, visual abstraction, detail enhancement, and seam carving.

Traditionally, image smoothing is often referred to as edge-preserving image smoothing [TM98, FFLs08, PHK11, XLXJ11, LZZG15, WCL15], where low-contrast details are simply removed while respecting high-contrast edges. Since this approach mainly examines pixel color and intensity difference [BSY14], it cannot effectively separate high-contrast textures from structures, and take them out [ZHJ14] for structure-aware image smoothing.

Recently, several methods have been proposed to preserve prominent structures while smoothing images. A pioneering approach is an optimization method [XYX12], which suppresses textures by regularization. However, due to the global optimization, oversmoothing often occurs [CLKL14]. Another approach is spatial filtering, which analyzes the spatial correlation between pixels and then filters each pixel by a weighted average over the associated candidate pixels. There are two kinds of spatial filters: local filters [SLD13, BSY14, SXJ14, CLKL14, ZDXZ15], which consider candidate pixels from a local window, and nonlocal mean (NLM) filter [KEE13], which expands the window to cover the entire image. Local filters are generally effective in terms of structure preservation and time complexity, but they cannot perform well when textures dominate the local regions. NLM filter, on the other hand, overcomes this issue by considering patches in the entire image, but structures may be blurred, since the averaging process may include nonlocal dissimilar pixels with large weights.

To improve the filtering quality, a variant of NLM filters, namely patch-group-based NLM filter, was developed with the following two key ideas. First, they group only the most similar patches for filtering each reference patch and completely ignore dissimilar patches. Second, they further enhance the analysis of the grouped patches by transforming them to wavelet domain [DFKE07] or SVD domain [GZZL16]. This approach has been successfully adopted in image processing tasks such as video denoising [JLSX10], image denoising [XZZ15], image inpainting [LZL15], and optical flow [DSHM14].

In this work, we focus on structure-preserving image smoothing and develop the following new techniques based on the patch-group-based NLM framework to improve image smoothing:

- First, we propose to use the diffusion map [FFL10] as a guidance image to improve the accuracy of patch similarities using the region covariance [TPM06, KEE13]. By mapping the image to the diffusion space, pixels within textured regions tend to be clustered. Hence, we can effectively suppress texture information.
- Second, we model structure-preserving image smoothing as a low-rank matrix recovery problem to effectively eliminate textures in similar patches. In short, once we identify the most similar patches of a reference patch, we construct the patch group (PG) matrix by arranging each similar patch along a matrix column. Therefore, for reference patches with rich structures, their PG matrices tend to possess the low-rank property due to mutu-
Several methods, see Fig. 1. Experiments show that it can produce the best image smoothing results that preserve structures while removing textures. Moreover, we adopt the alternative direction method of multipliers (ADMM) technique to solve the low-rank and sparse decomposition problem. To the best of our knowledge, we are not aware of any previous work that integrates the nonlocal low-rank and sparse model for structure-aware image smoothing.

To show the applicability of our method, we test it over a wide variety of images and compare it against several state-of-the-art methods, see Fig. 1. Experiments show that it can produce the best image smoothing results that preserve structures while removing textures. Moreover, we also show how our results can enhance the quality of edge detection, texture enhancement, and seam carving.

2. Related Work

In this section, we do not try to be exhaustive, and we focus mainly on works for image smoothing. The first group of related works is edge-preserving image smoothing, whose goal is to smooth out the low-contrast details without degrading the edges. Several methods have been proposed, e.g., bilateral filtering [TM98], weighted least squares [FPLS08], local histogram filtering [K10], $L_0$ gradient minimization [XLJ11], mixed domain manipulation [LGHM13], fast global smoother [MCL*14] and $L_1$ image transform [BHY15]. Since the approach relies mostly on pixel color/intensity difference, it cannot produce satisfactory results in preserving low-contrast structures when smoothing out high-contrast textures [XYYJ12].

Recently, some methods have been proposed to tackle the challenging problem of structure-aware image smoothing. A pioneering solution is a global optimization model by Xu et al. [XYYJ12], who proposed the relative total variation (RTV) measure to separate structures from textures. By involving a regularization term based on the RTV measure, they can penalize the structure and texture components distinctively, and smooth out textures accordingly. However, as a by-product of the global optimization, structures in the results tend to be oversmoothed [CLKL14].

Another approach for structure-aware image smoothing is spatial filtering, which estimates weights of pixel neighbors by similarities and filters each pixel by a weighted average. As described in the introduction, there are two kinds of spatial filters, local and nonlocal, according to where the candidates are selected from. Among the local filters, Bao et al. [BSY*14] constructed a minimum spanning tree (MST) over the input image and determined pixel weights as path lengths between nodes in the MST. Instead of building a MST over the entire image, Zhang et al. [ZDXZ15] decomposed the input image into superpixels and built a local MST for each superpixel region; hence, weights can be computed by combining internal weights from the local MST and external weights between superpixel regions. In addition, joint bilateral filter is also introduced to the texture removal problem, where the filtering weights are determined based on the structure information in the guidance image. Zhang et al. [ZSXJ14] adopted a rolling guidance scheme to iteratively refine the guidance image, which is initially the input image. Cho et al. [CLKL14] computed the guidance image based on the patch shift concept, which captures texture information of each
the alternative direction method of multipliers (ADMM) technique to solve the object function, we derive a new procedure based on


d window of reference patch (Section 3.1). Note that a patch is a small square to compute the patch similarity and to find more similar patches for each and use it as a guidance in the region covariance descriptor to com-

pared the two red boxes in Fig. 2(a&b)) while textures may still re-

amplitude in the region covariance descriptor (d) and using our descriptor (e). The histogram plot the distances between the obtained similar patches and the reference patch (the smaller the more similar) with the same set of bins, thereby showing that our descriptor enables us to find more similar patches.

Figure 2: Patch grouping analysis. We obtain forty most similar patches (blue) for the same reference patch (green) using the original region covariance descriptor (d) and using our descriptor (e). The histogram plot the distances between the obtained similar patches and the reference patch (the smaller the more similar) with the same set of bins, thereby showing that our descriptor enables us to find more similar patches.

3. Methods

This paper presents a new method for structure-preserving image smoothing based on the patch-group-based NLM approach. There are four major steps in our method. First, we build a diffusion map and use it as a guidance in the region covariance descriptor to com-
pute the patch similarity and to find more similar patches for each reference patch (Section 3.1). Note that a patch is a small square window of d pixels by d pixels. Second, we pack the similar patches into a patch-group (PG) matrix and formulate a low-rank matrix recovery problem for the structure component (Section 3.2). Third, to solve the object function, we derive a new procedure based on the alternative direction method of multipliers (ADMM) technique to decompose the low-rank and sparse components (Section 3.3).

Lastly, we produce the final structure image by aggregating the restored patches (Section 3.4).

3.1. Patch Matching and Grouping

3.1.1. Construct a diffusion map

The goal of the first step is to find a group of most similar patches for each reference patch in input image I. For this purpose, we employ [FFL10] to construct diffusion map Φ from input image I, see Fig. 2(a&b); since the diffusion map considers global distribution of pixels in feature space by computing dominant eigenvectors of a large affinity matrix that involves all pixel pairs, it can more effectively cluster pixels for suppressing texture information, see [FFL10] for detail. Here, we use σ²=0.1 (Gaussian kernel of the pairwise difference between two pixels), k=7 (number of eigenvectors), m=40 (number of samples), and empirically fix t (time sequence) as [1, 2, 4, 16] when producing the results shown in the paper; please refer to [FFL10] for details of these parameters.

Although the diffusion map can effectively cluster pixels, exploiting diffusion map alone is insufficient for structure-preserving image smoothing, since structures may disappear in the map (compare the two red boxes in Fig. 2(a&b)) while textures may still retain (see the green box in Fig. 2(b)). Therefore, we further adopt the diffusion map into the region covariance descriptor to improve the quality of patch matching and grouping.

3.1.2. Region covariance

The region covariance descriptor [TPM06] expresses an image region by the covariance of image features extracted from the pixels in the region. Given an input image I, Karacan et al. [KEE13] constructed a feature vector Fᵢ using seven simple image features: pixel intensity, 1st and 2nd derivatives, and pixel location. For a pixel at (p,q) in I, its feature vector Fᵢ( p, q) is given by:

\[
Fᵢ(p,q) = \left( I(p,q) , \frac{∂I}{∂x}(p,q) , \frac{∂I}{∂y}(p,q) , \frac{∂²I}{∂x²}(p,q) , \frac{∂²I}{∂y²}(p,q) , p , q \right)^T .
\]  

Motivated by the diffusion map (Φ)'s texture suppression capability, we propose to reformulate the feature vector by incorporating five additional terms using Φ to provide guidance for improving the accuracy of estimating patch similarity in our problem:

\[
\tilde{F}(p,q) = \left( Fᵢ^T(p,q) , \Phi(p,q) , \frac{∂Fᵢ}{∂x}(p,q) , \frac{∂Fᵢ}{∂y}(p,q) , \frac{∂²Fᵢ}{∂x²}(p,q) , \frac{∂²Fᵢ}{∂y²}(p,q) \right)^T .
\]
With this new formulation, we can incorporate the guidance information into the region covariance descriptor \( C_R \) for analyzing each \( d \times d \) (in pixel units) patch region \( R \):

\[
C_R = \frac{1}{d^2} \sum_{i=0}^{d^2} (\hat{F}_i - \mu)(\hat{F}_i - \mu)^T ,
\]

where \( \hat{F}_i \) is the feature vector of the \( i \)-th pixel in \( R \) and \( \mu \) the mean feature vector of all pixels in \( R \). Noted that the covariance matrices lie on the Riemannian manifold, so we need non-trivial measures to estimate their similarity \( [TPM06] \). For that purpose, we employ the model in \( [KEE13] \) to compute the distance between two covariance matrices: we represent each matrix as a vector by listing out the elements of Sigma points \( [KEE13] \), and compute the Euclidean distance between the two vectors as the similarity. Then, only the \( K \) most similar patches were retained. In our implementation, we consider nonlocal candidate patches within a large window \( ((2W_i + 1)^2 \text{ sq. pixel units}) \) centered at the pixel of the reference patch to trade-off between the computation time and quality of results: \( d \), \( K \), and \( W_i \) are set as 7, 40, and 20, respectively, in all the experiments.

Note that the original region covariance descriptor may misinterpret image structures as textures when the local patches have statistically similar appearance and scale, see \( [KEE13] \). Fortunately, the diffusion map can recover certain global structures by considering the affinity of pixel pairs over the entire image, so we can take such global information to improve the accuracy of patch similarity estimation. Fig. 2(c-e) shows a comparison result: (d) and (e) show patch grouping results produced with the original descriptor \( [KEE13] \) and our descriptor, respectively, for the same reference patch (green) shown in (c). The boxed histograms in (d) and (e) plot the distances from the reference patch to each group of similar patches (blue). From the plots, we can see that our method, which incorporates guidance information from the diffusion map, enables us to find patches that are more similar to the given reference patch.

### 3.2. Low-rank patch recovery

After finding the \( K \) most similar patches \( \{P_i\}_{i=1}^K \) for a given reference patch \( Q \), we next formulate the problem of decomposing the patch images into textures and structures as a low-rank recovery problem. To do so, we first construct a patch group (PG) matrix \( G_I \) by packing each of the patches, including \( Q \), as a matrix column:

\[
G_I = [Q, P_1, P_2, ..., P_K] .
\]

Hence, with respect to the output structure map, its PG matrix shall have a low-rank property due to the strong correlation among the similar patches, whereas corruption from textures would cause a high-rank value for the PG matrix of the input image.

Therefore, we propose to formulate a low-rank patch recovery \( [WGR^*09, CLMW11] \) to decompose \( G_I \) into \( G_S \) and \( G_E \), where \( G_S \) is a low-rank matrix and \( G_E \) is residual. Then, we can recover \( G_S \) from \( G_I \) by solving the following optimization problem:

\[
\min_{G_S, G_E} \|G_S\|_{w,*} + \lambda \|G_E\|_1 , \text{ s.t. } G_I = G_S + G_E ,
\]

where \( \text{rank}(G_S) \) denotes the rank of \( G_S \); \( \lambda \) is a positive weighting parameter; and \( \|G_E\|_1 \) denotes the sum of the absolute values of the matrix \( G_E \). The \( \|G_S\|_{w,*} \) term is introduced to improve the robustness of the recovery against outliers caused by the patch matching error or noise in input image. However, the above rank minimization is an NP-hard problem in general. A common way to make this non-convex optimization tractable is the Robust Principle Component Analysis (RPCA) \( [WGR^*09] \) method, which replaces the rank term \( \text{rank}(G_S) \) with the nuclear norm:

\[
\|G_S\|_* = \sum_{i=1}^{r} \sigma_i(G_S) ,
\]

where \( \sigma_i(G_S) \) is the \( i \)-th singular value of \( G_S \), and \( r \) is the number of singular values in \( G_S \).

For natural images, we have the general prior knowledge that larger singular values are typically more important than the smaller ones, since they represent the energy of the major components, which are typically related to important structures in the image. However, the standard nuclear norm \( (\text{Eq.6}) \) equally treats all the singular values. Hence, to better preserve structures in natural images, we propose to employ the weighted nuclear norm \( [GZZF14] \) to replace the nuclear norm as the surrogate of the rank in Eq.5, leading to a new optimization model, namely the weighted robust principle component analysis (WRPCA):

\[
\min_{G_S, G_E} \|G_S\|_{w,*} + \lambda \|G_E\|_1 , \text{ s.t. } G_I = G_S + G_E ,
\]

where \( \|G_S\|_{w,*} \) represents the weighted nuclear norm of the matrix \( G_S \), which is defined as the weighted sum of all singular values:

\[
\|G_S\|_{w,*} = \sum_{i=1}^{r} w_i \sigma_i(G_S) , \text{ where } w_i = \frac{\epsilon}{(\sigma_i(G_S) + \epsilon)} .
\]

Note that \( w_i \) is a non-negative weight; it should be inversely proportional to \( \sigma_i(G_S) \); \( \epsilon \) is set as \( 10^{-2} \); and \( \epsilon \) is introduced to avoid division by zero. The initial \( \sigma_i(G_S) \) is empirically estimated as:

\[
\sqrt{\max(\sigma_i^2(G_I) - \theta, 0)} ,
\]

where \( \sigma_i(G_I) \) is the \( i \)-th singular value of \( G_I \) and \( \theta \) is a parameter whose value depends on the textureness of the input image; we use a larger \( \theta \) for images with stronger textures. Throughout the experiments, we set \( \theta \) to be 0.0001, \( \lambda \) to be \( \frac{1}{\sqrt{n}} \) (where \( H \) is the height of the input image), and \( \theta \) in the range of \([30,60]\).

Fig.3 compares structure preservation in results produced with our method and with the original RPCA. Our new method models the low-rank recovery problem with the sparse term \( \|G_E\|_1 \) and the weighted nuclear norm, so we can better preserve the structure details than that with the original RPCA. In particular, we adaptively assign smaller weights for larger singular values.

© 2016 The Author(s)

Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.
Algorithm 1 Our ADMM procedure to decompose $G_I$

Input: a patch group matrix from input image: $G_I$, $\lambda$, $\gamma$, and $\beta$
1: Initialization: $Y^0 = 0; G_E = 0; k = 0$
2: while not converged do
3: Solve $G^{k+1}_S = \arg\min_{G_S} L(G_S, G_E, Y^k)$ by Eq. 14
4: Solve $G^{k+1}_E = \arg\min_{G_E} L(G^{k+1}_S, G_E, Y^k)$ by Eq. 17
5: Update $Y^{k+1}$ by the last equation in Eq. 11
6: $\beta = \gamma \beta$ and $k = k + 1$
7: end while
Output: $G^{k+1}_S$

3.3. ADMM Procedure for Low-rank Patch Recovery

To minimize Eq. 7, which is non-convex, we derive a new procedure based on the alternating direction method of multipliers (ADMM) technique. To do so, we first define the augmented Lagrangian function of Eq. 7 as:

$$L(G_S, G_E; Y) = ||G_S||_{w,*} + \lambda ||G_E||_1 + <Y, G_I - G_S - G_E > + \frac{\beta}{2} ||G_I - G_S - G_E||_F^2.$$  

where $||.||_F$ denotes the Frobenius norm; $Y$ is the Lagrange multiplier of the linear constraint; $\beta$ is the penalty parameter (in range $[0.4, 0.6]$) for the violation of the linear constraint; and $<., .>$ denotes the standard trace inner product. Note that we use a smaller $\beta$ for images with richer textures. The core idea of ADMM is to divide a problem into smaller subproblems, so that each has a closed-form solution, where the involved variables $G_S$ and $G_E$ can be minimized separately in alternating order. Specifically, we formulate the iterative procedure of ADMM as two subproblems:

$$G^{k+1}_S = \arg\min_{G_S} L(G^{k+1}_S, G_E, Y^k)$$  

$$G^{k+1}_E = \arg\min_{G_E} L(G^{k+1}_S, G_E, Y^k)$$

where $G^{k+1}_S$, $G^{k+1}_E$, and $Y^k$ are intermediate results in the $k$-th iteration. In each iteration, we solve for $G^{k+1}_S$ (subproblem 1) and then $G^{k+1}_E$ (subproblem 2). After that, we compute $Y^{k+1}$ from the results.

**Subproblem 1:** we solve for $G^{k+1}_S$ by the minimization below:

$$\min_{G_S} ||G_S||_{w,*} + <Y^k, G_I - G_S - G_E > + \frac{\beta}{2} ||G_I - G_S - G_E||_F^2.$$  

To obtain a closed-form solution, we propose to first add a new term $\frac{\beta}{2} ||\beta^{-1}Y^k||_F^2$ to Eq. 12, and transform it into:

$$\min_{G_S} ||G_S||_{w,*} + \frac{\beta}{2} ||G_S - (G_I - G_E + \frac{1}{\beta} Y^k)||_F^2.$$  

Since the new term does not contain $G_S$, minimizing Eq. 12 and minimizing Eq. 13 should give the same solution. However, the optimization in Eq. 13 differs from the traditional nuclear norm minimization, since $||G_S||_{w,*}$ assigns different weights to different singular values of $G_S$. Gu et al. [GZZF14] referred this as the weighted nuclear norm minimization (WNNM) and studied methods for solving it. In particular, since the weights $w_{i=0,...,n}$ are in a non-descending order, the closed-form solution of Eq. 13 according to [GZZF14] can be given by:

$$\left\{ \begin{array}{l}
(U, V) = \text{svd}(G_I - G_E + \frac{1}{\beta} Y^k) \\
G^{k+1}_S = USV^T
\end{array} \right.$$  

where the singular value shrinkage operator $S_{\frac{1}{\beta}}[\Sigma]_w$ is:

$$S_{\frac{1}{\beta}}[\Sigma]_w = \max(\Sigma_i - \frac{2w_i}{\beta}, 0)$$

where $\Sigma_i$ is the $i$-th singular values in the diagonal matrix $\Sigma$, and $w_i$ denotes the weight assigned to $\Sigma_i$. According to its definition in Eq. 8, the larger the singular values, the less they should be shrunk [GZZF14], leading to a better structure preservation.

**Subproblem 2:** Similarly, by adding the term $\frac{\beta}{2} ||\beta^{-1}Y^k||_F^2$ to Eq. 12, we can obtain $G^{k+1}_E$ by solving the following minimization:

$$\min_{G_E} \frac{\lambda}{2} ||G_E||_1 + \frac{\beta}{2} ||G_E - (G_I - G^{k+1}_S + \frac{1}{\beta} Y^k)||_F^2,$$

which has a closed-form solution [LCM10]:

$$G^{k+1}_E = S_{\frac{1}{\beta}}[G_I - G^{k+1}_S + \frac{1}{\beta} Y^k],$$

where $S_{\frac{1}{\beta}}[]$ is the soft-thresholding operator with the threshold $\frac{1}{\beta}$, and its definition is formulated as: $S_{\frac{1}{\beta}}[x] = \max(\frac{x}{\beta} - 1, 0)$.

By iteratively calculating $G^{k+1}_S$, $G^{k+1}_E$, and $Y^k$, we can compute the low-rank matrix $G_S$ and sparse matrix $G_E$, and then solve for the WRPCA problem, see the overall procedure in Algorithm 1. Note that our optimization has the same convergence rate as the ADMM model in [LCM10]; it usually converges in 10-20 iterations. In all experiments, we set $\gamma$ as 6 and set the initial value $Y^0$ as:

$$Y^0 = \text{sgn}(Y)^\top / \max(||\text{sgn}(Y)||_2, \lambda^{-1}||Y||_\infty),$$

Figure 4: Intermediate results during the iterative regularization process (see Section 3.4).
where \( \text{sgn} \) denotes the sign function, and \( ||.||_\infty \) is the maximum absolute value of all the matrix elements. For details about this initialization, readers may refer to [LGW'09, LCM10].

### 3.4. Patch Aggregation and Iterative Regularization

By applying the above procedure to every reference patch in input image \( I \), we can reconstruct the structure image \( S \) by aggregating all the restored patches \( (G_i) \) from Algorithm 1. Then, for each pixel at \((i, j)\), we compute its value in the resulting structure image by:

\[
(S^{(l)})_{i,j} = \Omega_{i,j} - 0.25 \text{sgn}(L \ast S^{(l-1)})_{i,j} ||(\nabla S^{(l-1)})_{i,j}||,
\]

where \( \Omega_{i,j} \) computes the average value of all the patches associated with the pixel; \( \text{sgn} \) is the sign function; \( L \) is the discrete 2D Laplacian filter, and \( \nabla S^{(l-1)} \) is the image gradient. The first term \( \Omega_{i,j} \) gives the average value, while the second term avoids potential blurring caused by the averaging process. Moreover, we adopt the iterative regularization method [DSL13] and progressively smooth the textures by running the procedure for a few rounds:

\[
I^{(l)} = S^{(l-1)} + \eta(I - S^{(l-1)}),
\]

where \( I^{(l)} \) and \( S^{(l)} \) are the (intermediate) image and corresponding structure image, respectively, at round \( l \), and \( \eta \) is the relaxation parameter (set as 0.65 in all our experiments) to control the proportion of textures to be added back to \( I^{(l)} \) in Eq. 20. Fig. 4 shows example intermediate results for \( S^{(l)} \). Algorithm 2 summarizes the image smoothing process, where \( T \) denotes the number of iterations.

### 4. Experiments

#### Parameters

Like existing image smoothing methods, our method also involves some parameters to control the smoothing, see Table 1. This includes \( \theta \) in Eq. 9, \( \beta \) in Eq. 10, and the number of rounds \( T \) (Algorithm 2) in the iterative regularization. In general, we use a larger \( \theta \) for images with strong or complicated textures, and we use a slightly larger \( \beta \) to control the image sharpness.

#### Time complexity

Let \( N \) denote the total number of pixels in input image, and we select reference patches by skipping \( (\mu-1) \) pixels vertically and horizontally over the image space, so the total number of reference patches is roughly \( \frac{N}{\mu} \) (\( \mu = 7 \)). To analyze the time complexity, we can look at the major steps in Algorithm 2:

1. **Step 4:** Computing a diffusion map costs \( O(m^2N) \), see [FFL10];
2. **Step 6:** The cost is \( O((2W_0+1)^2 \cdot f^2) \), since we can efficiently compare two \( f \times f \) covariance matrices in \( O(f^2) \) [Kee13], where \( f = 12 \) is the dimension of our feature vectors, see Eq. 2.
3. **Step 8:** In Algorithm 1, the SVD operation (see Eq. 14) dominates the optimization time in our ADMM-based solver, so

![Image smoothing results](image_url)
4.1. Comparison with State-of-the-Art Methods

Figs. 1, 6 and 7 show results that compare our method with several state-of-the-art methods designed for texture removal [SSD09, XYXJ12, KEE13, BSY∗14, ZSXJ14, CLKL14], and for edge preserving image smoothing [BHY15]. To produce their results, we corrupt an input image with different types of noise (top row), and the results (bottom row) show that our method can still recover plausible structure images in the presence of different amounts of Gaussian noise and salt-and-pepper noise, see Fig. 5 (b-e).

Robustness to noise. Another advantage of our method is its robustness against noise. Thanks to the low-rank recovery formulation with $L_1$ norm, our method can also recover the prominent structures in the presence of noise. Fig. 5 shows an example, where we corrupt an input image with different types of noise (top row), and the results (bottom row) show that our method can still recover plausible structure images in the presence of different amounts of Gaussian noise and salt-and-pepper noise, see Fig. 5 (b-e).

Robustness to noise. Another advantage of our method is its robustness against noise. Thanks to the low-rank recovery formulation with $L_1$ norm, our method can also recover the prominent structures in the presence of noise. Fig. 5 shows an example, where we corrupt an input image with different types of noise (top row), and the results (bottom row) show that our method can still recover plausible structure images in the presence of different amounts of Gaussian noise and salt-and-pepper noise, see Fig. 5 (b-e).
From the results, we observe that [BHY15] tends to degrade or oversmooth the main structures in the input images while filtering out the textures. Hence, some textures with high gradients cannot be effectively suppressed, since the method, as an edge-preserving image smoothing operator, has a limited capability to separate high contrast structures and textures. Also, the main structures of [SSD09] (Fig. 1(b)) are blurred in the texture smoothing procedure, since its capability to locate extremas fail in regions with mixed textures and structures. Later, as pointed out in [KEE13] and [CLKL14], [XYXJ12] tends to oversmooth some details during the removal of mosaic textures, thus obscuring the surface shading and making the restored result look somewhat flat. Similarly, [KEE13] also oversmoothes (blurs) the structures due to many non-zero weights from the extracted similar patches that carry textures to the filtering process. On the other hand, results from local filters such as [BSY14, ZSXJ14, CLKL14] are also unsatisfactory in terms of maintaining the structures. On the contrary, our method can effectively smooth out textures in various input images and better preserve structures as shown in the various results presented in the comparison figures, e.g., see the hairs and wrinkles in Fig. 1, see the eye lid and chin regions in the blown-up views of Fig. 6, and see the door frame shown in Fig. 7. More comparison results can be found in the supplementary material.

Fig. 8 presents more results produced by our method. Although these input images have different kinds of texture pattern, our method can consistently preserve the structures in these input images and effectively smooth out the textures. Again, more image smoothing results can be found in the supplementary material.
4.2. Applications

Being able to separate meaningful structures from textures benefits many applications. Below, we show how our results can be applied to various image processing applications.

**Edge detection.** This application aims to extract salient edges in the input image even in the presence of details such as textures. Fig. 9 shows an input image in cloth texture with a salient foreground. Since the texture itself has high contrast, directly applying the Canny operator cannot produce reasonable results, see Fig. 9(b). For the other three state-of-the-art methods, while they can detect salient edges, we can see that some of the resulting edges actually come from the textures and some important edges may be missed in the results, see areas marked with red and blue arrows in Fig. 9(c-f), respectively. In contrast, our method, which first produces a structure image, enables us to more effectively extract salient edges from the input image, see Fig. 9(h). Another advantage of our approach is that it does not require a pre-training of a large data set as in the above state-of-the-art edge detection methods; particularly, such pre-training is usually very time-consuming.

**Texture enhancement.** Our method can also allow us to enhance the underlying textures by manipulating the texture contrast. To do so, after we decompose the input image into textures and structures, we can increase the contrast in the texture component and then add it back to the structure component. Fig. 10 shows two examples. Obviously, our method can effectively enhance the underlying cloth texture without blurring the main structure, since our method can effectively capture the low-frequency structure.

**Seam carving.** Our method also benefits the seam carving application [AS07], whose goal is to resize an image by trying to preserve its salient contents. Typically, a gradient-based energy function is employed to compute and guide the resizing process.

However, high-gradient pixels in natural images not only locate at salient edges, but also in textured regions. Hence, we may not always be able to well preserve the salient objects in the input image. We show one such example in Fig. 11(a), where waves and clouds in the image have large gradient magnitude compared to the salient boat object. The left column shows the seam carving result from the original method [AS07]. As shown in Fig. 11(c), many vertical seams go across the boat, thus distorting and shrinking the boat object and making the result (see Fig. 11(e)) less satisfactory. In the right column, we illustrate the seam carving result produced by incorporating our structure image result (see Fig. 11(b)) into the energy function. Since high-frequency underlying textures from clouds and waves are mostly removed in our structure image, most of the seams no longer pass through the boat (see Fig. 11(d)), thus enabling us to better preserve the boat object in the final result (see Fig. 11(f)). Note that size of the boats in Fig. 11(e&f) as compared to the boat in the original image, i.e., Fig. 11(a).

**Limitations.** First, our method may not effectively handle images with sparse structures since such images may not provide sufficient similar patches in patch grouping. Second, our current implementation based on CPU is not fast for real-time applications; we leave it as future work to improve its performance by GPU computation.

5. Conclusion

We present a new structure-aware image smoothing method based on patch-group-based nonlocal means. In summary, there are three key contributions in our method. First, we introduce the diffusion map to provide guidance in the region covariance descriptor and improve the accuracy of finding similar patches. Second, we devise a low-rank matrix recovery formulation with the weighted robust principle component analysis (WRPCA) model and weighted nuclear norm to decompose the patch groups into structure (low rank)
and texture (sparse) components. Third, we formulate a new procedure to effectively solve this non-convex low-rank optimization problem based on the alternating direction method of multipliers (ADMM) technique. In the end, we demonstrate the superiority of our method by showing how it outperforms several state-of-the-art methods in terms of structure preservation and texture removal, its capability to smooth a wide variety of images, and its applicability to enhance the results in several other applications such as edge detection, texture enhancement, and seam carving.

Acknowledgments

The authors thank the reviewers for their worthwhile comments. Images are almost from a public dataset of [XYZ12]. The work described in this paper was supported by the Hong Kong Research Grants Council General Research Fund (Project No. 412513), the National Natural Science Foundation of China (Project No. 61233012), the National Natural Foundation of China with No. 61502137, and the Chinese University of Hong Kong strategic research fund and direct grant (4055061).

References

[XYYY12] Xu L., Yan Q., Xia Y., Jia J.: Structure extraction from texture via relative total variation. ACM Trans. Graph. (SIGGRAPH Asia) 31, 6 (2012), 139:1–10, 1, 2, 7, 3, 10.