



Full length article

Cartoon-Texture decomposition with patch-wise decorrelation[☆]

Xiaofang Li, Weiwei Wang*, Xiangchu Feng, Tingting Qi

School of Mathematics and Statistics, Xidian University, Xi'an 710071, China

ARTICLE INFO

Keywords:

Cartoon-Texture decomposition
Patch-wise cosine similarity
Total variation

ABSTRACT

Cartoon-Texture decomposition (CTD) is a fundamental task and has wide applications in image processing and computer vision. To enhance separation of the cartoon and texture, existing models explicitly introduce correlation terms to decorrelate the two components. However, existing correlations usually ignore the local geometric structure information, thus insufficient to decorrelate cartoon and texture. In this work, we propose the patch-wise cosine similarity to decorrelate the cartoon and texture. The proposed decorrelation term takes the local geometric information into account and is more effective in separating cartoon and texture. Combining our decorrelation term with the regularities for cartoon (Relative Total Variation (RTV)) and texture ($\text{div}(L^1)$ -norm), we propose a new CTD model. Extended experiments show that the proposed model outperforms existing methods in CTD, especially in preserving edges of the cartoon.

1. Introduction

Decomposing an image into distinct morphological components is a fundamental task in image processing [1], and has wide applications in computer vision [2–6]. Cartoon-Texture decomposition (CTD) is such a method, aiming to decompose an image into the cartoon component and the texture component. The cartoon part contains the main structure, including the homogeneous regions, contours and sharp edges; while the texture part contains the oscillating patterns, like fine details and locally repeated features. CTD has been widely used in edge detection [2,3], image restoration [4,5] and among others [6].

In general, CTD can be formulated in the following form:

$$\mathbf{f} = \mathbf{u} + \mathbf{v}, \quad (1)$$

where $\mathbf{f} \in \mathbb{R}^N$, $\mathbf{u} \in \mathbb{R}^N$ and $\mathbf{v} \in \mathbb{R}^N$ respectively denote the input image, cartoon and texture. To solve this ill-posed problem, various models have been developed [7]. The general form is

$$\hat{\mathbf{u}}, \hat{\mathbf{v}} = \underset{\mathbf{u}, \mathbf{v}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{f} - \mathbf{u} - \mathbf{v}\|_2^2 + \lambda_1 R_1(\mathbf{u}) + \lambda_2 R_2(\mathbf{v}), \quad (2)$$

where $R_1(\mathbf{u})$ and $R_2(\mathbf{v})$ are regularities of the cartoon and the texture, respectively. λ_1 and λ_2 are tuning parameters.

Existing methods roughly fall into four categories. The first category takes the cartoon and the texture as functions in certain functional spaces, and regularizes them by norms of the corresponding functional spaces [7–18]; the second category regards the cartoon and the texture

as vectors in the Euclidean space and regularizes them accordingly [19–23]; the third category considers patch-wise regularity from the matrix theory [24–27]; while some recent methods use unsupervised learning neural network [28,29].

The representatives of the first category are the models proposed by Meyer [7]. In these models, the cartoon is regularized by the Total Variation (TV) [8] and the texture is penalized with the G-norm, F-norm and E-norm in particular functional spaces [7]. Following this track, many variants are proposed by modifying the regularity on the cartoon and texture. As for the regularity on texture, the $\text{div}(L^p)$ -norm is adopted by Vese and Osher (VO model) to approximate the G-norm [9], where div represents the first order divergence operator. In case of $p = 2$, it reduces to the H^{-1} -norm and is used in the Osher-Solé-Vese (OSV) model [10]. Aujol and Gilboa further extend the H^{-1} -norm into the H^{-s} -norm [11]. Aujol et al. also utilize the dual method to approximate Meyer's G-model (AABC model) [12]. Yin et al. empirically show that the simple L^1 norm is effective in penalizing the texture of human face images [13]. As for the regularity on cartoon, TV guarantees effectiveness theoretically and empirically, though, it causes the annoying staircasing effect [14]. To overcome this drawback, a number of variants are proposed. The second-order TV [15], Total Generalized Variation (TGV) [16], and Nonlocal Total Variation (NLTV) [17] are among the examples. Nevertheless, these modifications may cause over-smoothing of the cartoon. The Relative Total Variation (RTV) shows better effect in preserving edges of the cartoon [18].

[☆] This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.

E-mail addresses: xli_96@stu.xidian.edu.cn (X. Li), wwwang@mail.xidian.edu.cn (W. Wang), xcfeng@mail.xidian.edu.cn (X. Feng), ttqi@stu.xidian.edu.cn (T. Qi).

In the second category, the most well-known method is the Morphological Component Analysis (MCA) [19]. It is based on the observation: The texture and the cartoon can be sparsely represented in appropriate dictionaries. For example, the Discrete Cosine Transform (DCT) is a good choice for texture and the Wavelet Transform is appropriate to represent the cartoon [20]. Consequently, sparsity inducing norms [30] are used to regularize the representation coefficients. To better express complicated textures, Starck et al. also used learned dictionaries [21] while Zhang et al. used Convolutional Sparse Coding (CSC) [22]. To better distinguish texture from large scale edges in the cartoon component, Xu et al. proposed the isotropic non-local transform [23].

The third category generally uses proper matrix norms to regularize cartoon or texture patches. Schaeffer and Osher regularized aligned texture patches by the low-rank regularization, or Low Patch-Rank in brief (LPR) [24]. Ma et al. further grouped similar patches and applied the low-rank regularization to each patch-group [25]. Zhang et al. used global nuclear norm to characterize the texture [26]. Such regularization shows good effects for globally well-patterned texture. For textures globally dissimilar but locally well-patterned, Ono et al. proposed the Block Nuclear Norm (BNN) [27].

The recent popularity of neural network also inspires a few exploration. Kim et al. used a neural network denoiser to regularize the cartoon component [28]. Zhou et al. utilized a modified Unet to generate the cartoon [29].

In addition to the regularities specific to the cartoon or the texture, intermediary regularities are used to decorrelate the cartoon and the texture [31–33]. Shahidi et al. considered the Pearson correlation of the texture and the cartoon [31]; Liu et al. used correlation of the texture and the gradient of the cartoon [32]. Xu et al. introduced the correlation between the gradient of the cartoon and the vector whose divergence represents the texture [33].

The correlation regularities in [32,33] add up the correlation of the two components at each pixel, and such pixel-wise correlation ignores the local geometric structure information of each pixel. The correlation in [31] is patch-wise, however it could cause problem since the texture part has an inherent zero mean while the cartoon does not [32]. Thus it is ineffective in separating cartoon and texture. In this work, we take the local geometric structure information into account. Specifically, we compute the cosine similarity of the two components in a neighbor/patch around each pixel. Such patch-wise cosine similarity (PACS) essentially represents the correlation of two pixels in a 2-dimensional local manifold. Besides, owing to the effectiveness of the RTV [18] and $\text{div}(L^1)$ -norm [9], we use them as the regularities for the cartoon and texture respectively. Then we formulate a new CTD model with these three terms.

In summary, the contributions of this work are three-fold.

- We propose a novel patch-wise decorrelation term PACS, which is more effective in distinguishing cartoon and texture than existing decorrelation terms.
- Combining our PACS with the RTV and $\text{div}(L^1)$ -norm, we present a new model for cartoon and texture decomposition.
- Extended experiments demonstrate that the proposed PACS obtains better performance than previous methods on CTD. Specifically, compared with the decomposition results of other methods, our model can better preserve edges in the cartoon, and the cartoon and texture have much less mixture.

The rest of this paper is organized as follows. We first review some related works in Section 2. Then we present our model and its solution algorithm in Section 3. Experimental results and analysis are provided in Section 4. Conclusions are finally drawn in Section 5.

2. Related work

The following VO model is proposed to approximate Meyer's G-model [7] by using $\text{div}(L^p)$ -norm to approximate the G-norm [9]:

$$\min_{u,g} \left\{ \|f - u - \text{div}g\|_2^2 + \lambda \sum_{i=1}^N \|(\nabla u)_i\|_2 + \mu \|g\|_p \right\}, \quad (3)$$

where u is the cartoon, and $v = \text{div}g$ denotes the texture. The second term is the well known TV [8] of u , and $\nabla : \mathbb{R}^N \rightarrow \mathbb{R}^N \times \mathbb{R}^N$ stands for the gradient operator. $g = (g_1, g_2) \in \mathbb{R}^N \times \mathbb{R}^N$ and $\|g\|_p = \left(\sum_{i=1}^N \left(\sqrt{(g_1)_i^2 + (g_2)_i^2} \right)^p \right)^{\frac{1}{p}}$. λ and μ are positive tuning parameters. The $\text{div}g$ has shown effectiveness in representing various textures.

The RTV model [18] is

$$\min_u \left\{ \|u - f\|_2^2 + \lambda \sum_{i=1}^N \left(\frac{\sum_{j \in n_i} w_{i,j} |(\nabla_x u)_j|}{|\sum_{j \in n_i} w_{i,j} (\nabla_x u)_j| + \epsilon} \right. \right. \\ \left. \left. + \frac{\sum_{j \in n_i} w_{i,j} |(\nabla_y u)_j|}{|\sum_{j \in n_i} w_{i,j} (\nabla_y u)_j| + \epsilon} \right) \right\}, \quad (4)$$

where n_i denotes the neighborhood of the pixel i , $w_{i,j} \propto \exp\left(-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2\sigma^2}\right)$ is a weighting function defined according to spatial affinity, and σ controls the spatial scale of the patch. ∇_x and ∇_y are the partial derivatives, and ϵ is a small positive number to avoid division by zero. The second term is the RTV of u . Compared to TV, the RTV is much better in preserving edges of the cartoon.

To decorrelate the texture and the cartoon, Liu et al. proposed the following model [32]:

$$\min_u \left\{ \|f - u\|_2^2 + \lambda \sum_{i=1}^N \|(\nabla u)_i\|_2 + \mu \sum_{i=1}^N |f_i - u_i| \|(\nabla u)_i\|_2 \right\}. \quad (5)$$

The third term tends to minimize the pixel-wise correlation of the texture $f - u$ and the gradient of the cartoon u . Xu et al. introduced in the VO model [9] the following term [33]:

$$\sum_{i=1}^N (\nabla u)_i g_i, \quad (6)$$

which can be regarded as the correlation of the gradient field of the cartoon and g . Shahidi et al. [31] explicitly added a decorrelation term into the OSV model [10]:

$$\min_u \left\{ \|\nabla(\Delta^{-1})(f - u)\|_2^2 + \lambda \sum_{i=1}^N \|(\nabla u)_i\|_2 + \mu \sum_{i=1}^N |\rho_{u,v}^{n_i}| \right\}, \quad (7)$$

where $\Delta^{-1}(f - u)$ represents the inverse Laplacian of $(f - u)$, or the functions whose Laplacian is equal to $(f - u)$, $\rho_{u,v}^{n_i}$ denotes the Pearson Correlation Coefficient (PCC) of u and v computed in a patch n_i centered at the pixel i .

3. Proposed method

In this section, we propose a new model for CTD. Owing to the effectiveness of the RTV [18] and $\text{div}(L^1)$ -norm [9], we use the RTV and $\text{div}(L^1)$ -norm to regularize the cartoon and texture respectively. The novelty of our model is the newly defined decorrelation term PACS.

3.1. Motivation

The decorrelation terms in Eqs. (5) and (6) consider pixel-wise correlation, which ignore the local geometric structure information. The patch-wise PCC in Eq. (7) is based on the assumption: The distribution of the pixel values in the patch is the same as that of the center pixel. However, for the center pixel on edges or in texture, this assumption is untrue.

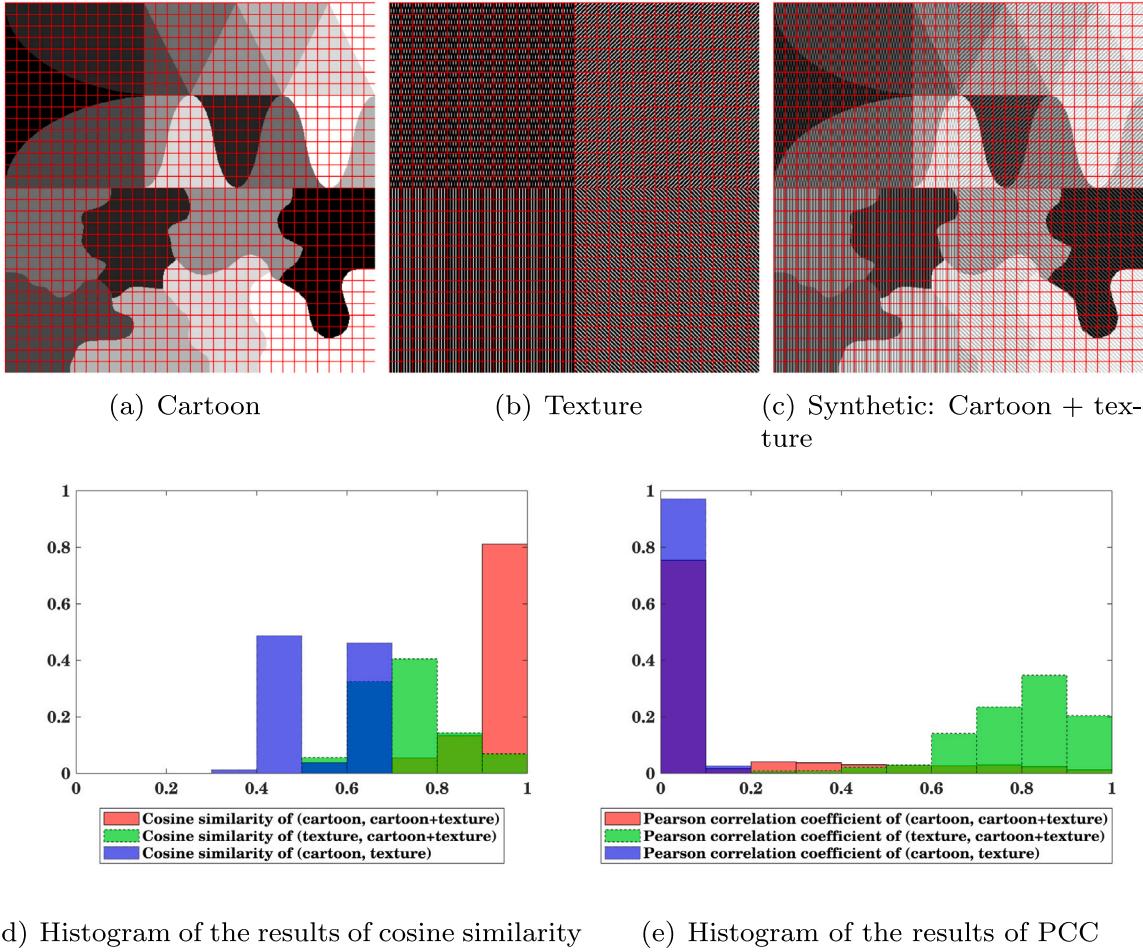


Fig. 1. PACS vs. patch-wise PCC.

In this work, we propose to use the following PACS to explicitly enhance CTD. Let b_{u_i} and b_{v_i} denote the patches (a square of size $r \times r$), centered at pixel i , in the cartoon and texture respectively, we define the patch-wise correlation as their cosine similarity:

$$\cos(b_{u_i}, b_{v_i}) = \frac{b_{u_i} \cdot b_{v_i}}{\|b_{u_i}\|_2 \|b_{v_i}\|_2}. \quad (8)$$

To intuitively compare the PCC with our PACS, we compute the two terms for patch pairs: (cartoon, cartoon+texture), (texture, cartoon+texture), and (cartoon, texture) on the synthetic image (with patch partition) shown in Fig. 1. The histograms of absolute values of the two terms are shown in Fig. 1(d) and (e). We can make the following observations: For patch pairs (cartoon, cartoon+texture) and (texture, cartoon+texture), PACS of vast majority patches (100%, 94%) is larger than 0.6; for patch pair (cartoon, texture), PACS of vast majority patches (53%) is less than 0.6. This means that, if the cartoon is mixed with texture, then PACS is large; and PACS is small for pure cartoon and texture patches. Therefore, minimizing PACS facilitates the decomposition. On the contrary, PCC of 97% patch pairs (cartoon, texture) is less than 0.1, while PCC of 75% patch pairs (cartoon, cartoon+texture) is less than 0.1. Though PCC is small for pure cartoon and texture patches, it is also small even if the cartoon is mixed with texture. Therefore minimizing PCC is insufficient to facilitate the decomposition.

3.2. Proposed model

First, we rewrite the RTV [18] in the following form:

$$RTV(\mathbf{u}) = \sum_{j=1}^N \left((\mathbf{w}_x)_j |(\nabla_x \mathbf{u})_j| + (\mathbf{w}_y)_j |(\nabla_y \mathbf{u})_j| \right) = \|\mathbf{w}_x \odot \nabla_x \mathbf{u}\|_1 + \|\mathbf{w}_y \odot \nabla_y \mathbf{u}\|_1, \quad (9)$$

where \mathbf{w}_x and \mathbf{w}_y have the same size with \mathbf{u} , \odot denotes the element-wise product. For pixel j , $(\mathbf{w}_x)_j$ and $(\mathbf{w}_y)_j$ are defined as

$$\begin{aligned} (\mathbf{w}_x)_j &= \sum_{i \in n_j} \frac{w_{i,j}}{|\sum_{j \in n_i} w_{i,j} (\nabla_x \mathbf{u})_j| + \epsilon}, \text{ and} \\ (\mathbf{w}_y)_j &= \sum_{i \in n_j} \frac{w_{i,j}}{|\sum_{j \in n_i} w_{i,j} (\nabla_y \mathbf{u})_j| + \epsilon}. \end{aligned} \quad (10)$$

In this form, RTV can be interpreted as an adaptive weighted anisotropic TV [8] and it performs better than TV in preserving edges in the cartoon.

By using the RTV and $\text{div}(L^1)$ -norm [9] to characterize the cartoon and texture respectively, and combining our PACS for decorrelation, we give our model, also called PACS, as follows:

$$\min_{\mathbf{u}, \mathbf{g}} \left\{ \frac{1}{2} \|\mathbf{f} - \mathbf{u} - \text{div} \mathbf{g}\|_2^2 + \alpha (\|\mathbf{w}_x \odot \nabla_x \mathbf{u}\|_1 + \|\mathbf{w}_y \odot \nabla_y \mathbf{u}\|_1) + \beta \|\mathbf{g}\|_1 + \frac{\gamma}{2} \|\cos(\mathbf{b}_u, \mathbf{b}_{\text{div} \mathbf{g}})\|_2^2 \right\}, \quad (11)$$

where

$$\|\cos(\mathbf{b}_u, \mathbf{b}_{\text{div} \mathbf{g}})\|_2^2 = \sum_{i=1}^N \left(\cos(b_{u_i}, b_{\text{div} g_i}) \right)^2, \quad (12)$$

α, β and γ are positive tuning parameters. Note that the third term is the special case of $\text{div}(L^p)$ -norm with $p = 1$.

3.3. Numerical algorithm

We use the Alternating Direction Method of Multipliers (ADMM) [34] framework to solve problem (11). Introducing four auxiliary variables $\mathbf{d}_1 = \nabla_x \mathbf{u}$, $\mathbf{d}_2 = \nabla_y \mathbf{u}$, $\mathbf{z} = \mathbf{g}$ and $\mathbf{v} = \text{div} \mathbf{g}$, the corresponding augmented Lagrangian function is

$$\begin{aligned} \mathcal{L}(\mathbf{u}, \mathbf{d}_1, \mathbf{d}_2, \mathbf{g}, \mathbf{z}, \mathbf{v}; \mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4) &= \frac{1}{2} \|\mathbf{f} - \mathbf{u} - \mathbf{v}\|_2^2 \\ &+ \alpha (\|\mathbf{w}_x \odot \mathbf{d}_1\|_1 + \|\mathbf{w}_y \odot \mathbf{d}_2\|_1) + \beta \|\mathbf{z}\|_1 + \frac{\gamma}{2} \|\cos(b_{\mathbf{u}}, b_{\mathbf{v}})\|_2^2 \\ &+ \frac{\mu}{2} \|\nabla_x \mathbf{u} - \mathbf{d}_1 + \mathbf{m}_1\|_2^2 + \frac{\mu}{2} \|\nabla_y \mathbf{u} - \mathbf{d}_2 + \mathbf{m}_2\|_2^2 \\ &+ \frac{\mu}{2} \|\mathbf{g} - \mathbf{z} + \mathbf{m}_3\|_2^2 + \frac{\mu}{2} \|\text{div} \mathbf{g} - \mathbf{v} + \mathbf{m}_4\|_2^2, \end{aligned} \quad (13)$$

where $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$ and \mathbf{m}_4 are Lagrangian multipliers, and μ is a penalty parameter. Then we solve for one variable by fixing others at each iteration step. The corresponding sub-problems for $\mathbf{u}, \mathbf{d}_1, \mathbf{d}_2, \mathbf{g}, \mathbf{z}$ and \mathbf{v} are as follow:

$$\begin{aligned} \mathbf{u}^{k+1} &= \underset{\mathbf{u}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{f} - \mathbf{u} - \mathbf{v}^k\|_2^2 + \frac{\gamma}{2} \|\cos(b_{\mathbf{u}}, b_{\mathbf{v}^k})\|_2^2 \\ &+ \frac{\mu}{2} \|\nabla_x \mathbf{u} - \mathbf{d}_1^k + \mathbf{m}_1^k\|_2^2 + \frac{\mu}{2} \|\nabla_y \mathbf{u} - \mathbf{d}_2^k + \mathbf{m}_2^k\|_2^2, \end{aligned} \quad (14)$$

$$\mathbf{d}_1^{k+1} = \underset{\mathbf{d}_1}{\operatorname{argmin}} \alpha \|\mathbf{w}_x \odot \mathbf{d}_1\|_1 + \frac{\mu}{2} \|\nabla_x \mathbf{u}^{k+1} - \mathbf{d}_1 + \mathbf{m}_1^k\|_2^2, \quad (15)$$

$$\mathbf{d}_2^{k+1} = \underset{\mathbf{d}_2}{\operatorname{argmin}} \alpha \|\mathbf{w}_y \odot \mathbf{d}_2\|_1 + \frac{\mu}{2} \|\nabla_y \mathbf{u}^{k+1} - \mathbf{d}_2 + \mathbf{m}_2^k\|_2^2, \quad (16)$$

$$\mathbf{g}^{k+1} = \underset{\mathbf{g}}{\operatorname{argmin}} \frac{\mu}{2} \|\mathbf{g} - \mathbf{z}^k + \mathbf{m}_3^k\|_2^2 + \frac{\mu}{2} \|\text{div} \mathbf{g} - \mathbf{v}^k + \mathbf{m}_4^k\|_2^2, \quad (17)$$

$$\mathbf{z}^{k+1} = \underset{\mathbf{z}}{\operatorname{argmin}} \beta \|\mathbf{z}\|_1 + \frac{\mu}{2} \|\mathbf{g}^{k+1} - \mathbf{z} + \mathbf{m}_3^k\|_2^2, \quad (18)$$

$$\begin{aligned} \mathbf{v}^{k+1} &= \underset{\mathbf{v}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{f} - \mathbf{u}^{k+1} - \mathbf{v}\|_2^2 + \frac{\gamma}{2} \|\cos(b_{\mathbf{u}^{k+1}}, b_{\mathbf{v}})\|_2^2 \\ &+ \frac{\mu}{2} \|\text{div} \mathbf{g}^{k+1} - \mathbf{v} + \mathbf{m}_4^k\|_2^2. \end{aligned} \quad (19)$$

The multipliers are updated by using the gradient ascending scheme:

$$\mathbf{m}_1^{k+1} = \mathbf{m}_1^k + \nabla_x \mathbf{u}^{k+1} - \mathbf{d}_1^{k+1}, \quad (20)$$

$$\mathbf{m}_2^{k+1} = \mathbf{m}_2^k + \nabla_y \mathbf{u}^{k+1} - \mathbf{d}_2^{k+1}, \quad (21)$$

$$\mathbf{m}_3^{k+1} = \mathbf{m}_3^k + \mathbf{g}^{k+1} - \mathbf{z}^{k+1}, \quad (22)$$

$$\mathbf{m}_4^{k+1} = \mathbf{m}_4^k + \text{div} \mathbf{g}^{k+1} - \mathbf{v}^{k+1}. \quad (23)$$

As discussed in Appendix A, the solution of problem (14) can be approximated by the solution of the following system:

$$\begin{aligned} &\left(1 + \gamma \mathbf{A}^k + \mu \left(\nabla_x^T \nabla_x + \nabla_y^T \nabla_y\right)\right) \mathbf{u} \\ &= (\mathbf{f} - \mathbf{v}^k - \gamma \mathbf{B}^k) + \mu \left(\nabla_x^T (\mathbf{d}_1^k - \mathbf{m}_1^k) + \nabla_y^T (\mathbf{d}_2^k - \mathbf{m}_2^k)\right), \end{aligned} \quad (24)$$

which can be easily handled by the Fast Fourier Transform (FFT) if periodic boundary condition is exploited to the first-order derivative operators in horizontal and vertical directions [4,33].

Then we update the weights \mathbf{w}_x^{k+1} and \mathbf{w}_y^{k+1} by using Eq. (10) with \mathbf{u}^{k+1} .

Problems (15) and (16) have the following closed-form solution, respectively:

$$\mathbf{d}_1^{k+1} = \mathcal{S}_{\frac{\alpha \mathbf{w}_x^{k+1}}{\mu}} (\nabla_x \mathbf{u}^{k+1} + \mathbf{m}_1^k), \quad (25)$$

$$\mathbf{d}_2^{k+1} = \mathcal{S}_{\frac{\alpha \mathbf{w}_y^{k+1}}{\mu}} (\nabla_y \mathbf{u}^{k+1} + \mathbf{m}_2^k), \quad (26)$$

where $\mathcal{S}_\lambda(\cdot)$ is the soft-thresholding operator [35]:

$$\mathcal{S}_\lambda(x) = \max\{|x| - \lambda, 0\} \odot \text{sgn}(x), x \in \mathbb{R}^N. \quad (27)$$

Algorithm 1 The iterative algorithm for the proposed model

Input: Original image f .

- 1: Initialize: $\mathbf{u}^0, \mathbf{d}_1^0, \mathbf{d}_2^0, \mathbf{g}^0, \mathbf{z}^0, \mathbf{v}^0, \mathbf{m}_1^0, \mathbf{m}_2^0, \mathbf{m}_3^0, \mathbf{m}_4^0, \alpha, \beta, \gamma, \mu, k = 0, \epsilon = 10^{-3}$.
- 2: Output: \mathbf{u}, \mathbf{v} .
- 3: **while** not converge **do**
- 4: compute \mathbf{A}^k and \mathbf{B}^k for all pixels by using Eq. (A.6) with \mathbf{u}^k and \mathbf{v}^k ;
- 5: update \mathbf{u}^{k+1} by using Eq. (24);
- 6: update \mathbf{w}_x^{k+1} and \mathbf{w}_y^{k+1} by using Eq. (10) with \mathbf{u}^{k+1} ;
- 7: update \mathbf{d}_1^{k+1} by using Eq. (25);
- 8: update \mathbf{d}_2^{k+1} by using Eq. (26);
- 9: update \mathbf{g}^{k+1} by using Eq. (28);
- 10: update \mathbf{z}^{k+1} by using Eq. (29);
- 11: compute \mathbf{C}^k and \mathbf{D}^k for all pixels by using Eq. (B.3) with \mathbf{u}^{k+1} and \mathbf{v}^k ;
- 12: update \mathbf{v}^{k+1} by using Eq. (30);
- 13: update $\mathbf{m}_1^{k+1}, \mathbf{m}_2^{k+1}, \mathbf{m}_3^{k+1}, \mathbf{m}_4^{k+1}$ by using Eq. (20), (21), (22) and (23) respectively;
- 14: check the convergence condition:
- 15: $\max \left(\frac{\|\mathbf{u}^{k+1} - \mathbf{u}^k\|_2^2}{\max \left\{ 1, \|\mathbf{u}^k\|_2^2 \right\}}, \frac{\|\mathbf{v}^{k+1} - \mathbf{v}^k\|_2^2}{\max \left\{ 1, \|\mathbf{v}^k\|_2^2 \right\}} \right) \leq \epsilon$;
- 16: $k = k + 1$;
- 17: **end while**

The optimal solution of problem (17) satisfies the linear system:

$$(1 + \text{div}^T \text{div}) \mathbf{g} = (\mathbf{z}^k - \mathbf{m}_3^k) + \text{div}^T (\mathbf{v}^k - \mathbf{m}_4^k), \quad (28)$$

which can be easily handled by FFT if periodic boundary condition is imposed to the first-order divergence operator [4,33].

Problem (18) also has the closed-form solution:

$$\mathbf{z}^{k+1} = \mathcal{S}_{\frac{\mu}{\mu}} (\mathbf{g}^{k+1} + \mathbf{m}_3^k). \quad (29)$$

As discussed in Appendix B, the solution of problem (19) can be approximated by

$$\mathbf{v}^{k+1} = \frac{\mathbf{f} - \mathbf{u}^{k+1} - \gamma \mathbf{D}^k + \mu (\text{div} \mathbf{g}^{k+1} + \mathbf{m}_4^k)}{1 + \gamma \mathbf{C}^k + \mu}. \quad (30)$$

The whole procedure is summarized in Algorithm 1. At each iteration, the computational complexity of Eq. (25), (26) and (29) is $O(N)$, N is the total number of pixels in the image. The cost for Eq. (10) is $O(\sigma N)$ [18]. The main cost for solving Eq. (28) is two FFTs (include one inverse FFT), each at a cost of $O(N \log(N))$. Moreover, the cost for solving $\mathbf{A}^k, \mathbf{B}^k, \mathbf{C}^k$ and \mathbf{D}^k is $O(r^2 N)$. So the cost for solving Eq. (24) is $O(r^2 N \log(r^2 N))$. The cost for solving Eq. (30) is $O(r^2 N)$. Thus, the computation complexity of Algorithm 1 is $O(r^2 N \log(r^2 N))$.

4. Experiment and discussion

In this section, we empirically analyze our model and verify its performance. In Section 4.1, we discuss some aspects of our model. Firstly, we study how the parameters affect the decomposition results. Secondly, we conduct three ablation studies to show the merits of the proposed model, and the new decorrelation term in particular. Finally, we provide the numerical convergence analysis of Algorithm 1. In Section 4.2, we evaluate our method on several commonly used test images: two synthetic image and four natural images, as shown in Fig. 2. The synthetic image ‘Texmos3_syn’ is the sum of ‘Texmos3’ and ‘Syn_texture’ cited in [36]. The synthetic image ‘Our_syn’ is superposed by a cartoon image (from website) and a texture image (from Brodatz database [37]) with the ratio of 8:2. The four natural images include two gray images and two color images [18]. Note that color images are processed channel by channel. The proposed PACS is compared with

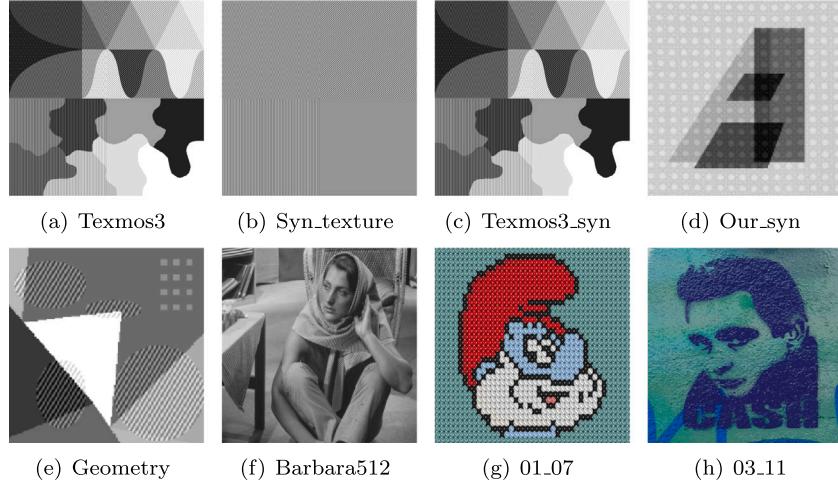


Fig. 2. Test images.

three related methods: VO [9], Xu [33] and RTV [18].¹ Besides, we also evaluate it with two state-of-art methods: ISO [23]² and CLRP [26].³ The optimal parameters of existing methods are set according to the corresponding literature. When the ground-truth images are available, the quality of the decomposition results are measured by two commonly used metrics: Peak Signal Noise Ratio (PSNR) and Structural Similarity Index Metric (SSIM) [38]; otherwise, we just present decomposition results for visual comparison. The definition of PSNR is

$$PSNR = 10 \log_{10} \left(\frac{MAX_O^2}{MSE} \right), MSE = \frac{\sum_{i=1}^N (O_i - R_i)^2}{N}, \quad (31)$$

where $O \in \mathbb{R}^N$ is the output image, $R \in \mathbb{R}^N$ is the reference image, and MAX_O is the maximum fluctuation in the output image data type. MSE is the mean-square error between the output image O and reference image R .

4.1. Model analysis and ablation study

(1) Parameter Setting. The proposed model involves four parameters α, β, γ and r , where α, β and γ are used to tune the regularities, r is the patch size. The parameters α and β are set according to the referred literature. Our experiments show that our model is insensitive to the parameter γ . Fig. 3(a)–(d) shows the decomposition results and the corresponding zoomed-in patches of image ‘Our_syn’ for different values of $\gamma = \{1, 0.1, 0.01, 0\}$, with $\alpha = 0.0004, \beta = 1$ and $r = 11$. With the truth of each component, we calculate the PSNR/SSIM on the cartoon and texture respectively. The PSNR/SSIM values are listed in Table 1. With the help of PACS, the cartoon and texture are perfectly separated. While $\gamma = 0$, i.e. without our PACS, the edges of cartoon are squeezed into the texture. Besides, both qualitative and quantitative results show a little difference for different values of γ , suggesting that the results are not sensitive to the value of γ . So for almost all test images used in our experiment, we fix $\gamma = 1$. Fig. 3(e)–(h) show the decomposition results and the corresponding zoomed-in patches of image ‘Our_syn’ for different values of $r = \{3, 7, 15, 19\}$, with $\alpha = 0.0004, \beta = 1$ and $\gamma = 1$. The corresponding PSNR/SSIM values are also listed in Table 1. Qualitative results show a little difference for different values of r . Quantitative results suggest $r = 11$ is a good choice for the patch size.

(2) Ablation Study. The ablation study aims to show the effectiveness of our decorrelation term in CTD and its advantage over

Table 1
PSNRs (dB) and SSIMs of the synthetic image ‘Our_syn’ with different γ and r .

γ, r	Cartoon		Texture	
	PSNR	SSIM	PSNR	SSIM
$\gamma = 1, r = 11$	31.8017	0.9878	34.3234	0.7679
$\gamma = 0.1, r = 11$	31.7945	0.9877	34.3294	0.7686
$\gamma = 0.01, r = 11$	31.7726	0.9874	34.3243	0.7687
$\gamma = 0, r = 11$	30.9431	0.9807	33.8817	0.7640
$\gamma = 1, r = 3$	31.7774	0.9863	33.8036	0.7467
$\gamma = 1, r = 7$	31.7975	0.9871	34.1248	0.7592
$\gamma = 1, r = 15$	31.7888	0.9868	34.1451	0.7608
$\gamma = 1, r = 19$	31.7228	0.9870	34.1810	0.7626

existing decorrelation terms. Fig. 4 shows the cartoons (on the top row) and the textures (on the third row) obtained by some baselines and the proposed method. Fig. 4(a)–(e) respectively correspond to Xu’s model [33], VO [9]+PCC [31], VO [9]+PACS, PACS(wo) and PACS(w). The former three models are all based on the VO model [9], and differ by the decorrelation terms. Xu’s model [33] uses the pixel-wise correlation (Eq. (6)). To compare our PACS decorrelation term with the PCC in [31], we combine them with the VO model [9], and call them VO+PCC and VO+PACS, respectively. PACS(w) means our PACS model (Eq. (11)) and PACS(wo) means our model without the decorrelation term (or the special case when $\gamma = 0$). For the first three models, we set $\alpha = 1.5, \beta = 1$ and $\gamma = 0.01$; for the last two models, we set $\alpha = 0.00025, \beta = 3$.

First, we can note that, in the framework of VO model [9], our decorrelation term performs better than that given by Xu [33] and the PCC [31] in separating the large scale edges from the texture. In view of the fact that the RTV [18] is better than the standard TV [8] in preserving large scale edges, we use the RTV [18] in our PACS model. Truly, our PACS model obtains good separation: no visible mixture of the cartoon and the texture. In contrast, we can see leakage of large scale edges in the texture obtained by PACS(wo) without our decorrelation term. This confirms the effectiveness of our decorrelation regularity in separating edges of the cartoon and the texture.

(3) Numerical Convergence. On all test images, our algorithm shows numerical convergence. For the test image ‘Our_syn’, we plot the residuals of $R_1 = d_1 - \nabla_x u$, $R_2 = d_2 - \nabla_y u$, $R_3 = z - g$ and $R_4 = v - \text{div}g$, the relative errors of the Lagrangian multipliers $m_i, i \in \{1, 2, 3, 4\}$, the relative errors of u, g , and the value of the objective function (Eq. (11)) in the iteration, as shown in Fig. 5. The residuals are defined as

$$(R_1^k, R_2^k, R_3^k, R_4^k)$$

¹ <http://www.cse.cuhk.edu.hk/~leojia/projects/texturesep/>.

² <https://github.com/RuotaoXu/IsoDecompose>.

³ <https://github.com/Zhiyuan-Zhang510zg/CLRP>.

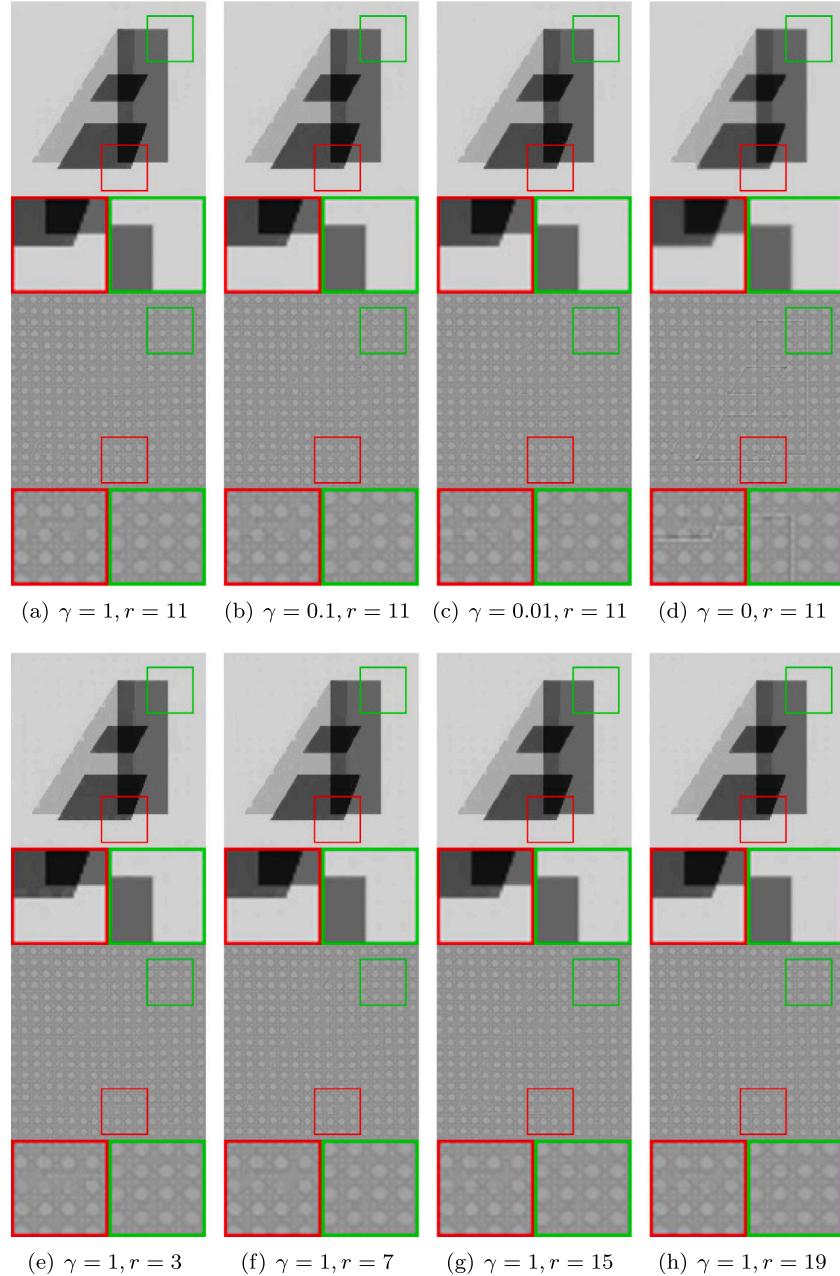


Fig. 3. Results of PACS with different values of γ and r (on ‘Our_syn’).

$$= \left(\frac{\|d_1^k - \nabla_x u^k\|_1}{N}, \frac{\|d_2^k - \nabla_y u^k\|_1}{N}, \frac{\|z^k - g^k\|_1}{N}, \frac{\|\nu^k - \text{div}g^k\|_1}{N} \right). \quad (32)$$

The relative errors $M_i, i \in \{1, 2, 3, 4\}$ of the Lagrangian multipliers $m_i, i \in \{1, 2, 3, 4\}$ are defined as

$$\begin{aligned} & (M_1^k, M_2^k, M_3^k, M_4^k) \\ &= \left(\frac{\|m_1^k - m_1^{k-1}\|_1}{\|m_1^{k-1}\|_1}, \frac{\|m_2^k - m_2^{k-1}\|_1}{\|m_2^{k-1}\|_1}, \frac{\|m_3^k - m_3^{k-1}\|_1}{\|m_3^{k-1}\|_1}, \frac{\|m_4^k - m_4^{k-1}\|_1}{\|m_4^{k-1}\|_1} \right). \end{aligned} \quad (33)$$

And the relative errors of u and g are defined as

$$(R_u^k, R_g^k) = \left(\frac{\|u^k - u^{k-1}\|_1}{\|u^{k-1}\|_1}, \frac{\|g^k - g^{k-1}\|_1}{\|g^{k-1}\|_1} \right). \quad (34)$$

It can be seen that, the residuals in Eq. (32), the relative errors in Eqs. (33) and (34) all decrease eventually. And the value of the objective function (Eq. (11)) quickly decreases in the first few iterations, then slows down and finally tends to be stationary. This demonstrates that the iteration is Algorithm 1 has numerical convergence.

4.2. Assessment and comparison

We first compare the proposed model with related methods on ‘Texmos3_syn’. The decomposition results are shown in Fig. 6. For better observing the differences among the results, we also show some zoomed-in patches below the corresponding images. We can see that, the proposed model has at least two advantages over other methods. First, the proposed model can better protect edges of the cartoon. For

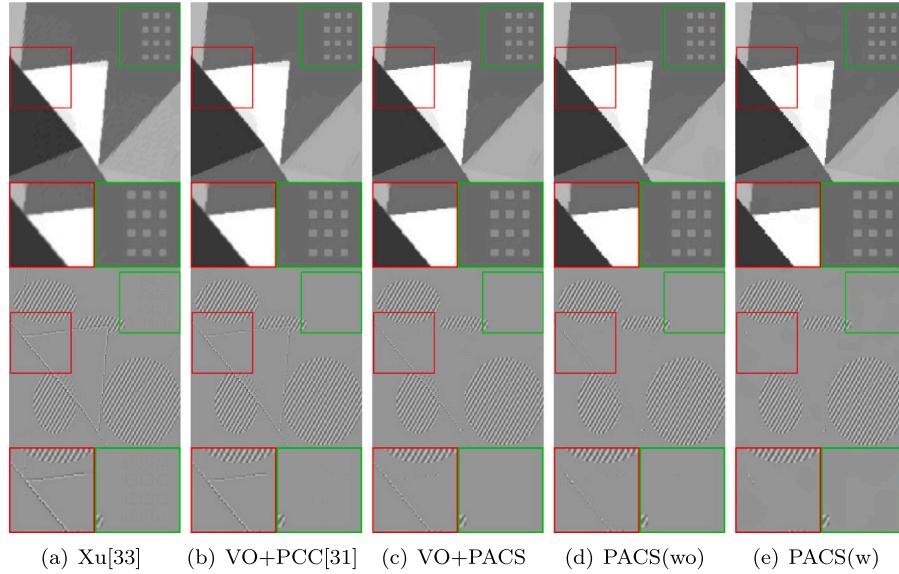


Fig. 4. Ablation study. From top to bottom: cartoon components, corresponding zoomed-in patches of the cartoon, texture components, corresponding zoomed-in patches of the texture.

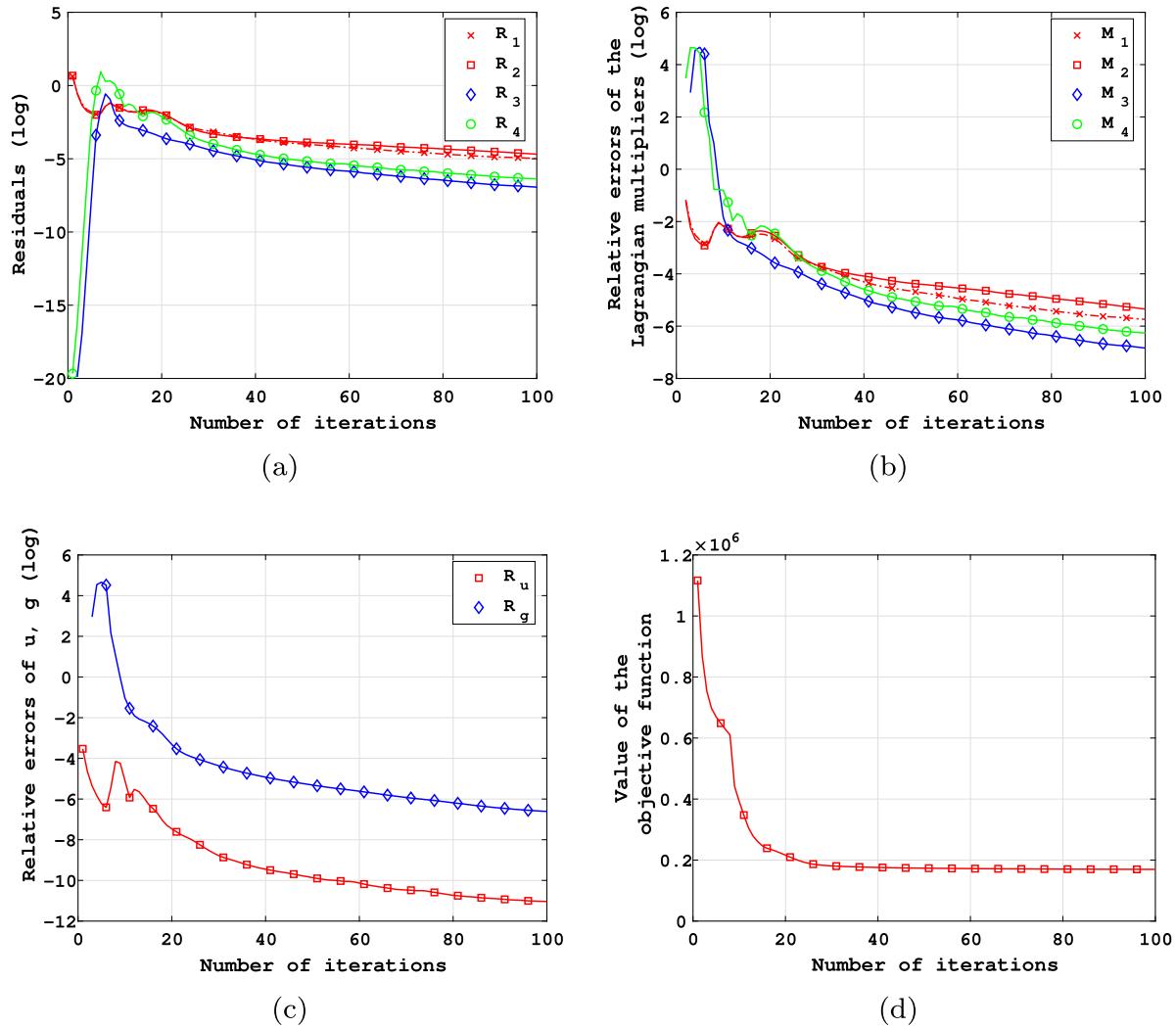


Fig. 5. Plots of the residuals in Eq. (32), the relative errors of the Lagrangian multipliers $m_i, i \in \{1, 2, 3, 4\}$ in Eq. (33), the relative errors of u, g in Eq. (34), and the value of the objective function (Eq. (11)) against the number of iterations.

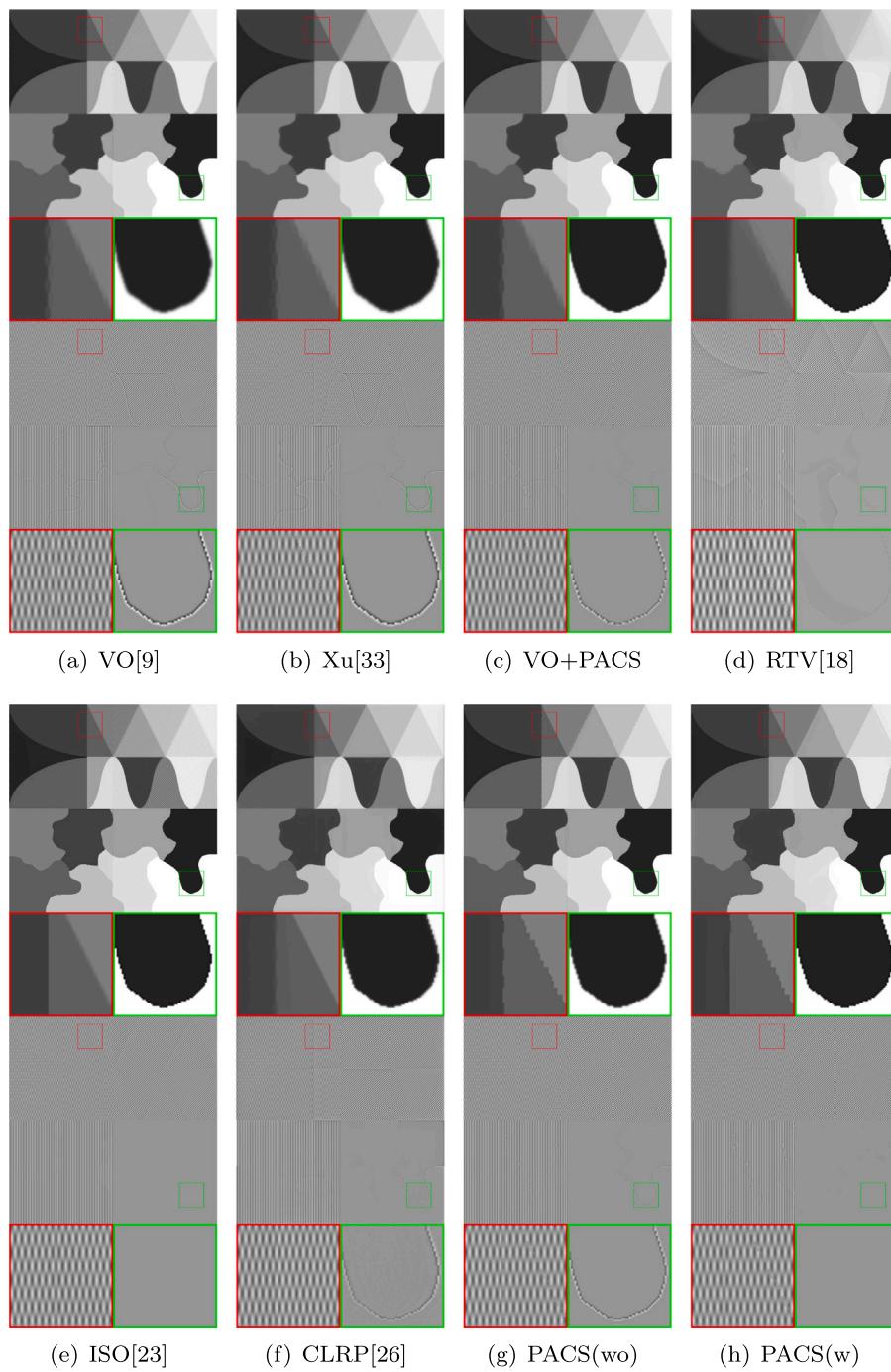


Fig. 6. Visual comparison on the synthetic image 'Texmos3_syn'.

Table 2

PSNRs (dB) and SSIMs of the synthetic image 'Texmos3_syn'.

Criterion	Cartoon		Texture	
	PSNR	SSIM	PSNR	SSIM
VO [9]	31.0489	0.9609	32.2504	0.9396
Xu [33]	31.3310	0.9742	32.4772	0.9424
VO + PACS	32.4093	0.9799	33.0778	0.9490
RTV [18]	32.2547	0.9781	33.6684	0.9443
ISO [23]	30.8437	0.8878	32.2447	0.9448
CLRP [26]	31.3751	0.9706	33.5219	0.9350
PACS(wo)	32.5179	0.9843	32.9745	0.9494
PACS(w)	33.1555	0.9859	33.3298	0.9573

Table 3

Average PSNRs (dB) and SSIMs of 16 synthetic images.

Criterion	Cartoon		Texture	
	PSNR	SSIM	PSNR	SSIM
VO [9]	29.7014	0.9678	32.5129	0.7193
Xu [33]	29.9545	0.9694	32.5664	0.7155
VO + PACS	30.7806	0.9805	33.6126	0.7663
RTV [18]	31.1590	0.9795	33.6920	0.7605
ISO [23]	31.3027	0.9820	32.6668	0.6768
CLRP [26]	30.7719	0.9816	31.6398	0.5876
PACS(wo)	31.1491	0.9851	34.3097	0.7893
PACS(w)	31.2488	0.9880	34.4457	0.7903

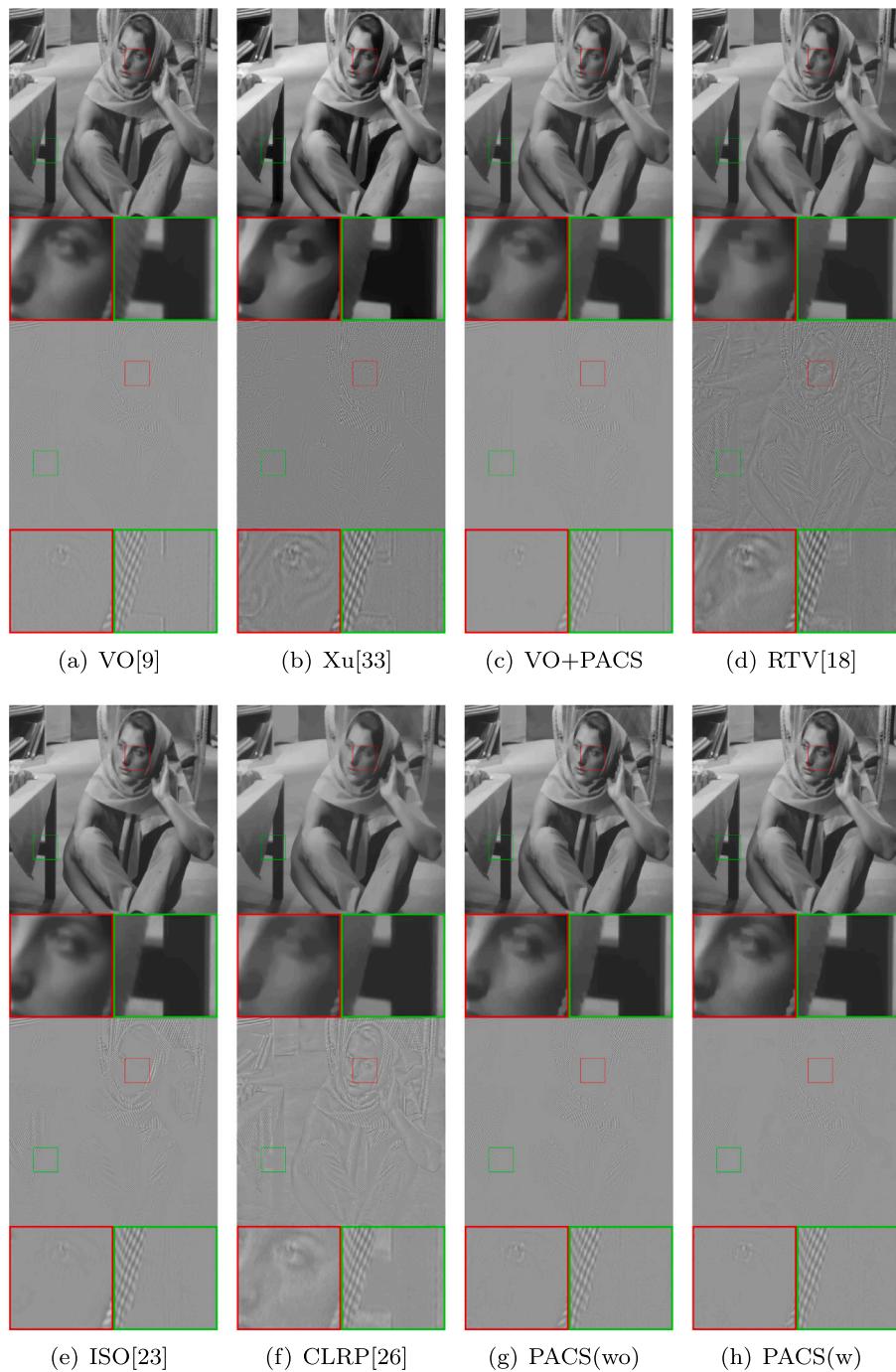


Fig. 7. Visual comparison on ‘Barbara512’.

example, edges of the cartoon obtained by our method (Fig. 6(h)) are sharper than that obtained by other methods (it can be seen clearly in the zoomed-in images of the cartoon). Second, the proposed model can better separate texture and cartoon so that they have less mixture. For example, in the texture components obtained by other methods, we can notice (strong or weak) edges of the cartoon; while it is invisible in the texture obtained by our method. With the ground truth components, we can assessment all results by the PSNR and SSIM metric, reported in Table 2. The cartoon obtained by our method has the best PSNR and SSIM index; as for the texture, RTV [18] and CLRP [26] obtain higher PSNR, while our method obtains higher SSIM. Actually, the texture obtained by RTV [18] (Fig. 6(d)) and CLRP [26] (Fig. 6(f)) has visible

leaked edges of the cartoon, while the texture obtained by our method (Fig. 6(h)) has little mixture.

We also test the proposed method on 16 synthetic images, which are acquired in the same manner as the image ‘Our_syn’. The average PSNR/SSIM values are listed in Table 3. The texture obtained by our method has the best PSNR and SSIM index; as for the cartoon, ISO [23] obtains higher PSNR, while our method obtains higher SSIM. More details are provided in the supplementary material.

Then we compare all methods on test images (f)–(h) shown in Fig. 2. The decomposition results are respectively shown in Figs. 7–9. We can draw similar observations: Our method obtains better separation of the components, and the edges of the cartoon are sharper. For color

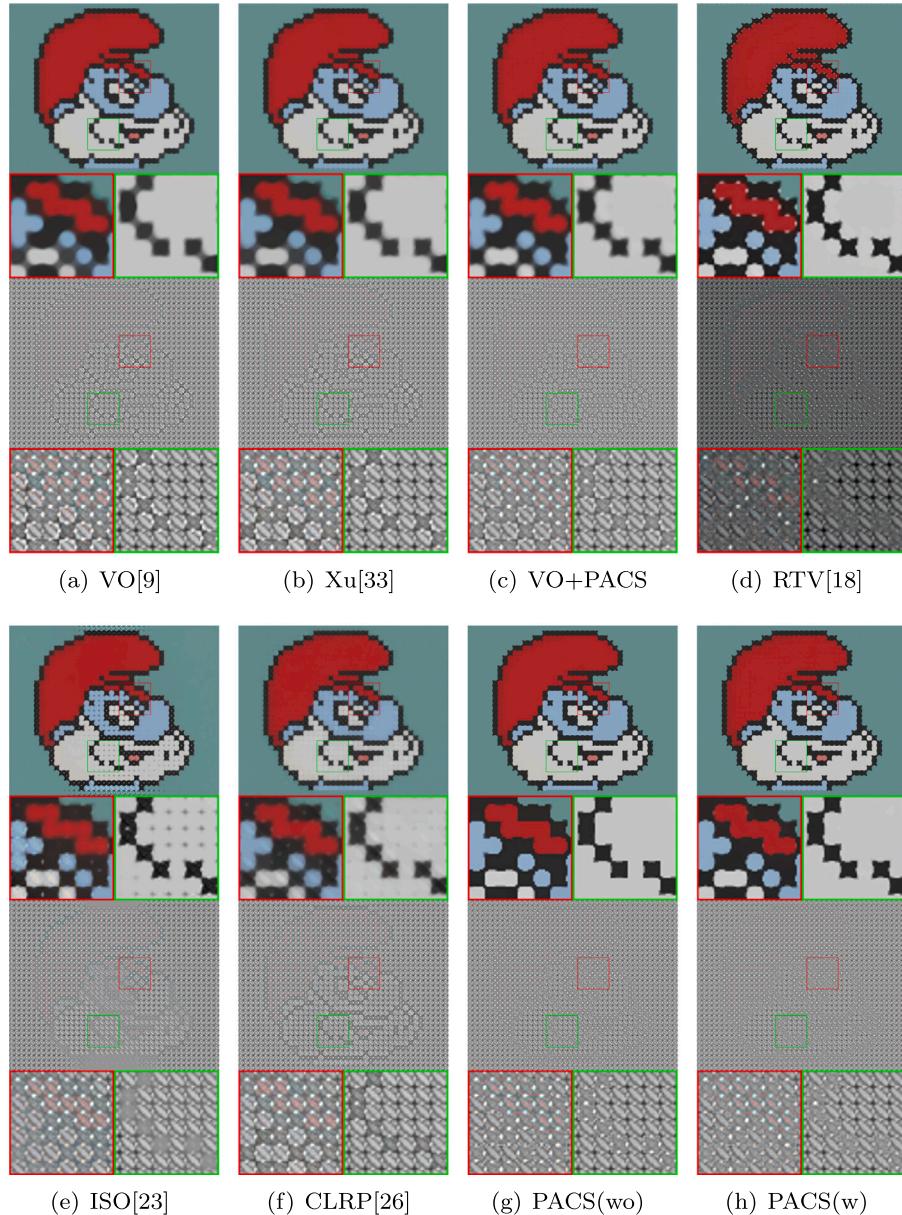


Fig. 8. Visual comparison on '01_07'.

images, our method has an additional advantage: It can well preserve the colors in the cartoon component. In general, both the cartoon and the colors are piece-wise smooth, so the colors should be well preserved in the cartoon component rather than in the texture. From the results shown in Figs. 8 and 9, we can observe that, the texture obtained by all methods have much or less colors, while the textures obtained by our method have less colors. More experimental results are provided in the supplementary material.

Finally, we study application of CTD in edge detection. Fig. 10 shows the results: The first row shows the original test image '13_25' [18] and the cartoons obtained by different methods, and the second row shows edges extracted from the original image and the corresponding cartoons by using the Canny edge detector. We can see that, there is little texture mixed into the cartoon obtained by our method, so the edges detected from the cartoon are more regular. In contrast, the cartoons obtained by other methods have much or less cluttered textures, which causes strong or weak jittering of the edges.

In summary, all the visual results demonstrate that, the proposed decorrelation term is effective in enhancing separation of texture and cartoon, and our model is better in preserving the edges of the cartoon.

5. Conclusion

We introduced a patch-wise decorrelation (PACS) term, in which each pixel in the image is embedded into a local 2-dimensional manifold (a patch), rather than an isolated point. So our PACS can separate cartoon and texture more effectively. Combining with RTV and $\text{div}(L^1)$ -norm, we proposed a new CTD model, also called PACS. Extended experiments show that the proposed model outperforms related existing methods in CTD. Specifically, our model can better preserve edges in the cartoon, and the cartoon and texture have much less mixture. In our future research, we will explore more applications of our model in image processing.

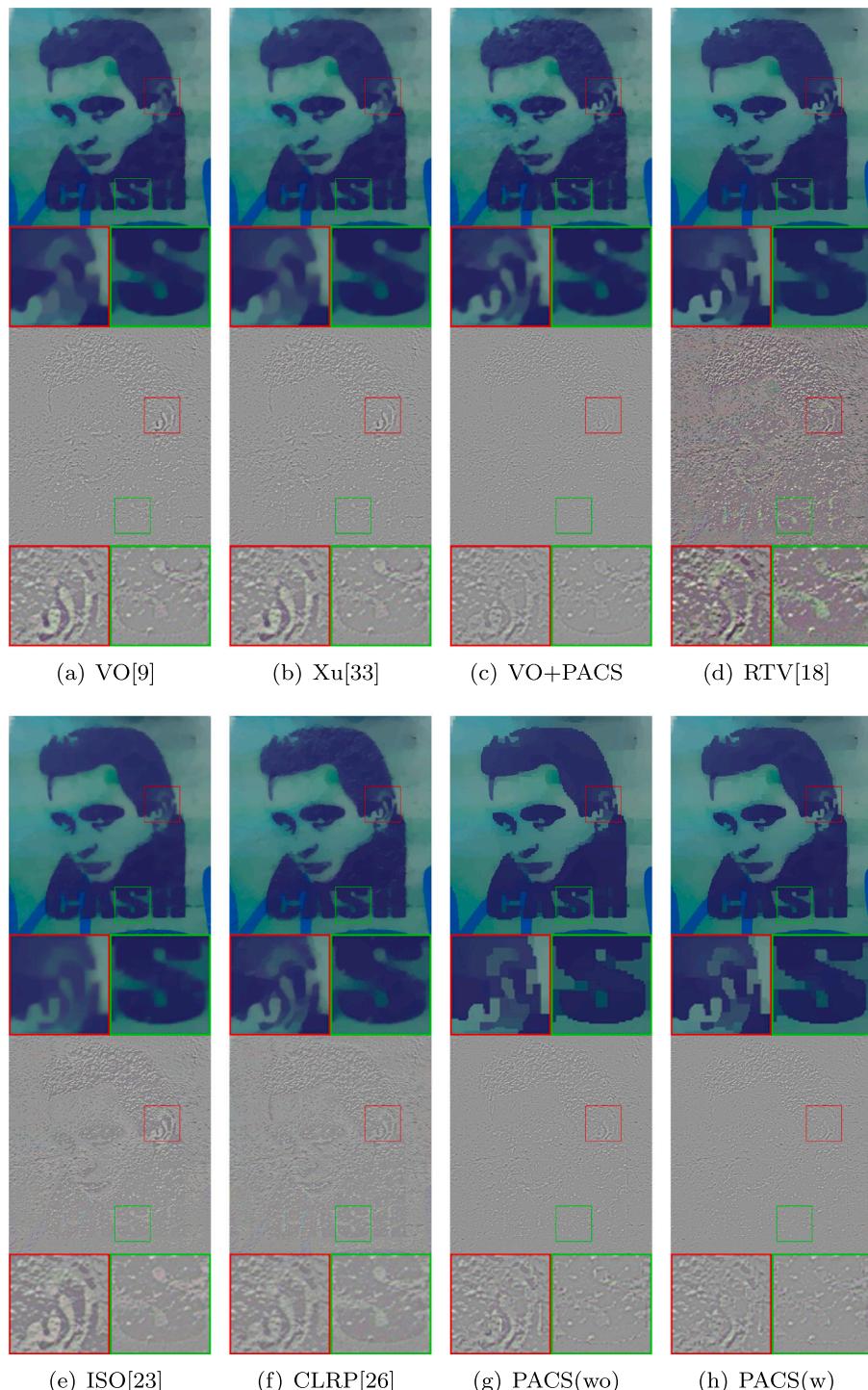


Fig. 9. Visual comparison on '03_11'.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank the editors and the anonymous reviewers for their constructive comments and suggestions. This paper is supported by the National Natural Science Foundation of China (Grant Nos. 61972264, 62072312, 61971005), Natural Science Foundation of Guangdong Province (Grant No. 2019A1515010894) and Natural Science Foundation of Shenzhen (20200807165235002).

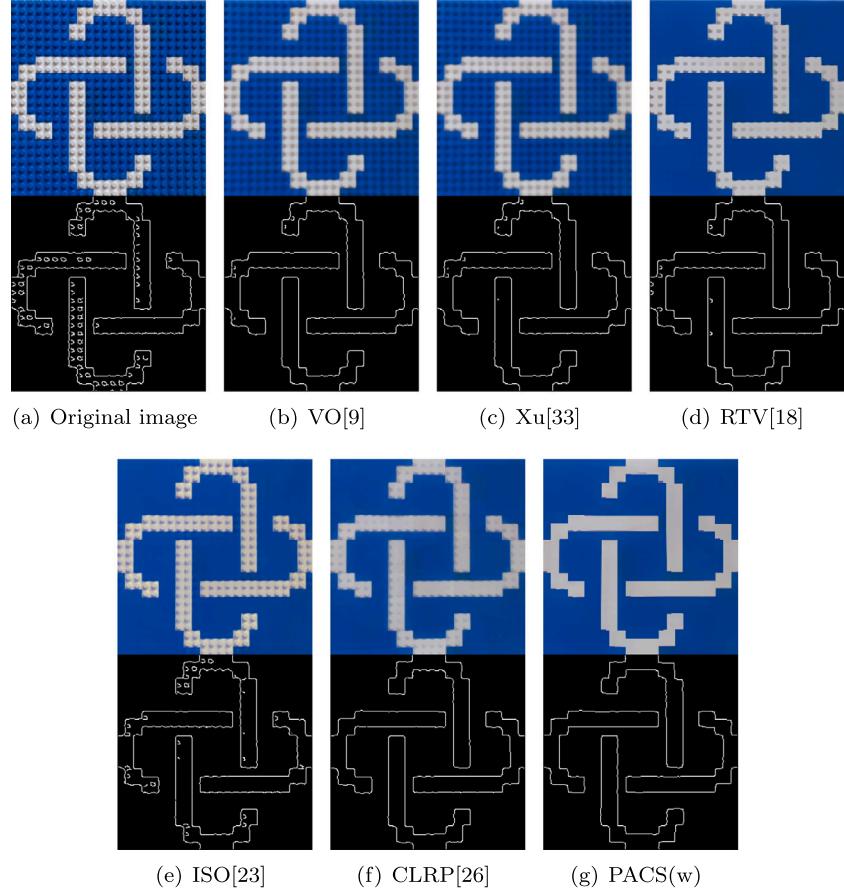


Fig. 10. Detected edges on '13_25'.

Appendix A. Minimization for the sub-problem (14)

The objective function in Eq. (14) is differentiable with respect to \mathbf{u} and the optimal solution can be obtained by finding the stationary point. However, the derivative of $\frac{\gamma}{2} \|\cos(\mathbf{b}_\mathbf{u}, \mathbf{b}_\mathbf{v}^k)\|_2^2 = \frac{\gamma}{2} \sum_{i=1}^N \left(\frac{(\mathbf{b}_{\mathbf{u}_i} \cdot \mathbf{b}_{\mathbf{v}_i^k})^2}{\|\mathbf{b}_{\mathbf{u}_i}\|_2^2 \|\mathbf{b}_{\mathbf{v}_i^k}\|_2^2} \right)$ with respect to \mathbf{u} is complicated. To make it easy, we fix $\|\mathbf{b}_{\mathbf{u}_i}\|_2^2$ in the denominator as $\|\mathbf{b}_{\mathbf{u}_i^k}\|_2^2$ and get the approximate $\frac{\gamma}{2} \|\cos(\mathbf{b}_\mathbf{u}, \mathbf{b}_\mathbf{v}^k)\|_2^2 \approx \frac{\gamma}{2} \sum_{i=1}^N \left(\frac{(\mathbf{b}_{\mathbf{u}_i} \cdot \mathbf{b}_{\mathbf{v}_i^k})^2}{\|\mathbf{b}_{\mathbf{u}_i^k}\|_2^2 \|\mathbf{b}_{\mathbf{v}_i^k}\|_2^2} \right)$. Then we approximate the whole objective function in Eq. (14) thereby and get the following approximate problem:

$$\begin{aligned} & \min_{\mathbf{u}} \frac{1}{2} \|\mathbf{f} - \mathbf{u} - \mathbf{v}^k\|_2^2 + \frac{\gamma}{2} \sum_{i=1}^N \left(\frac{(\mathbf{b}_{\mathbf{u}_i} \cdot \mathbf{b}_{\mathbf{v}_i^k})^2}{\|\mathbf{b}_{\mathbf{u}_i^k}\|_2^2 \|\mathbf{b}_{\mathbf{v}_i^k}\|_2^2} \right) \\ & + \frac{\mu}{2} \|\nabla_x \mathbf{u} - \mathbf{d}_1^k + \mathbf{m}_1^k\|_2^2 + \frac{\mu}{2} \|\nabla_y \mathbf{u} - \mathbf{d}_2^k + \mathbf{m}_2^k\|_2^2. \end{aligned} \quad (\text{A.1})$$

The derivative of the first, the third and the last term with respect to \mathbf{u} is $(\mathbf{u} + \mathbf{v}^k - \mathbf{f}), \mu \nabla_x^T (\nabla_x \mathbf{u} - \mathbf{d}_1^k + \mathbf{m}_1^k)$ and $\mu \nabla_y^T (\nabla_y \mathbf{u} - \mathbf{d}_2^k + \mathbf{m}_2^k)$, respectively. Now we consider the derivative of the second term with respect to each \mathbf{u}_i . Note that, not only $\mathbf{b}_{\mathbf{u}_i}$ contains \mathbf{u}_i , the patches $\mathbf{b}_{\mathbf{u}_j}$ ($j \in n_i$, the neighborhood of pixel i) also contain \mathbf{u}_i . Putting them together, we have

$$\frac{\gamma}{2} \sum_{j \in n_i} \left(\frac{(\mathbf{b}_{\mathbf{u}_j} \cdot \mathbf{b}_{\mathbf{v}_j^k})^2}{\|\mathbf{b}_{\mathbf{u}_j^k}\|_2^2 \|\mathbf{b}_{\mathbf{v}_j^k}\|_2^2} \right). \quad (\text{A.2})$$

So the derivative of the second term with respect to \mathbf{u}_i is

$$\gamma \sum_{j \in n_i} \left(\frac{\mathbf{b}_{\mathbf{u}_j} \cdot \mathbf{b}_{\mathbf{v}_j^k}}{\|\mathbf{b}_{\mathbf{u}_j^k}\|_2^2 \|\mathbf{b}_{\mathbf{v}_j^k}\|_2^2} \right) \mathbf{v}_i^k. \quad (\text{A.3})$$

Rewrite Eq. (A.3) as

$$\begin{aligned} & \gamma \sum_{j \in n_i} \left(\frac{\mathbf{b}_{\mathbf{u}_j} \cdot \mathbf{b}_{\mathbf{v}_j^k}}{\|\mathbf{b}_{\mathbf{u}_j^k}\|_2^2 \|\mathbf{b}_{\mathbf{v}_j^k}\|_2^2} \right) \mathbf{v}_i^k \\ & = \gamma \left(\sum_{j \in n_i} \frac{(\mathbf{v}_i^k)^2 \mathbf{u}_i}{\|\mathbf{b}_{\mathbf{u}_j^k}\|_2^2 \|\mathbf{b}_{\mathbf{v}_j^k}\|_2^2} + \sum_{j \in n_i} \frac{\mathbf{v}_i^k \sum_{l \in n_j, l \neq i} (\mathbf{u}_l \mathbf{v}_l^k)}{\|\mathbf{b}_{\mathbf{u}_j^k}\|_2^2 \|\mathbf{b}_{\mathbf{v}_j^k}\|_2^2} \right). \end{aligned} \quad (\text{A.4})$$

In the second term on the right hand side, we fix \mathbf{u}_l as \mathbf{u}_l^k for pixel $l \in n_j$ and $l \neq i$, we get the following approximation of Eq. (A.4):

$$\gamma \sum_{j \in n_i} \left(\frac{\mathbf{b}_{\mathbf{u}_j} \cdot \mathbf{b}_{\mathbf{v}_j^k}}{\|\mathbf{b}_{\mathbf{u}_j^k}\|_2^2 \|\mathbf{b}_{\mathbf{v}_j^k}\|_2^2} \right) \mathbf{v}_i^k \approx \gamma \left(\sum_{j \in n_i} \frac{(\mathbf{v}_i^k)^2 \mathbf{u}_i}{\|\mathbf{b}_{\mathbf{u}_j^k}\|_2^2 \|\mathbf{b}_{\mathbf{v}_j^k}\|_2^2} + \sum_{j \in n_i} \frac{\mathbf{v}_i^k \sum_{l \in n_j, l \neq i} (\mathbf{u}_l^k \mathbf{v}_l^k)}{\|\mathbf{b}_{\mathbf{u}_j^k}\|_2^2 \|\mathbf{b}_{\mathbf{v}_j^k}\|_2^2} \right) = \gamma (\mathbf{A}_i^k \mathbf{u}_i + \mathbf{B}_i^k), \quad (\text{A.5})$$

where

$$\begin{aligned} \mathbf{A}_i^k &= \sum_{j \in n_i} \frac{(\mathbf{v}_i^k)^2}{\|\mathbf{b}_{\mathbf{u}_j^k}\|_2^2 \|\mathbf{b}_{\mathbf{v}_j^k}\|_2^2}, \text{ and} \\ \mathbf{B}_i^k &= \sum_{j \in n_i} \frac{\mathbf{v}_i^k \sum_{l \in n_j, l \neq i} (\mathbf{u}_l^k \mathbf{v}_l^k)}{\|\mathbf{b}_{\mathbf{u}_j^k}\|_2^2 \|\mathbf{b}_{\mathbf{v}_j^k}\|_2^2}. \end{aligned} \quad (\text{A.6})$$

So the derivative of the second term with respect to \mathbf{u} is

$$\gamma (\mathbf{A}^k \mathbf{u} + \mathbf{B}^k), \quad (\text{A.7})$$

where \mathbf{A}^k and \mathbf{B}^k have the same size with \mathbf{u} , with \mathbf{A}_i^k and \mathbf{B}_i^k being the i th entry. Putting the derivatives of the first, the third, and the last term in Eq. (A.1) with respect to \mathbf{u} together with the approximate derivative (Eq. (A.7)) of the second term in Eq. (A.1), and letting the sum be zero, we get the following equation:

$$\begin{aligned} & \left(1 + \gamma \mathbf{A}^k + \mu \left(\nabla_x^T \nabla_x + \nabla_y^T \nabla_y\right)\right) \mathbf{u} \\ &= (\mathbf{f} - \mathbf{v}^k - \gamma \mathbf{B}^k) + \mu \left(\nabla_x^T (\mathbf{d}_1^k - \mathbf{m}_1^k) + \nabla_y^T (\mathbf{d}_2^k - \mathbf{m}_2^k)\right). \end{aligned} \quad (\text{A.8})$$

This equation can be easily solved and we take the solution of this equation as the approximate solution of the sub-problem (14).

Appendix B. Minimization for the sub-problem (19)

For the sub-problem (19), we adopt the same technique to approximate $\frac{\gamma}{2} \|\cos(b_{\mathbf{u}}^{k+1}, b_{\mathbf{v}})\|_2^2$ and the derivative of its approximation. Specifically, we first approximate $\frac{\gamma}{2} \|\cos(b_{\mathbf{u}}^{k+1}, b_{\mathbf{v}})\|_2^2$ as $\frac{\gamma}{2} \sum_{i=1}^N \left(\frac{(b_{\mathbf{u}_i^{k+1}} \cdot b_{\mathbf{v}_i})^2}{\|b_{\mathbf{u}_i^{k+1}}\|_2^2 \|b_{\mathbf{v}_i}\|_2^2} \right)$

by fixing $\|b_{\mathbf{v}_i}\|_2^2$ in the denominator as $\|b_{\mathbf{v}_i^k}\|_2^2$. Then the objective function in Eq. (19) can be approximated as

$$\min_{\mathbf{v}} \frac{1}{2} \|\mathbf{f} - \mathbf{u}^{k+1} - \mathbf{v}\|_2^2 + \frac{\gamma}{2} \sum_{i=1}^N \left(\frac{(b_{\mathbf{u}_i^{k+1}} \cdot b_{\mathbf{v}_i})^2}{\|b_{\mathbf{u}_i^{k+1}}\|_2^2 \|b_{\mathbf{v}_i^k}\|_2^2} \right) + \frac{\mu}{2} \|\operatorname{div} \mathbf{g}^{k+1} - \mathbf{v} + \mathbf{m}_4^k\|_2^2. \quad (\text{B.1})$$

Similar to the deduction of Eq. (A.8), the derivative of the second term with respect to \mathbf{v} can be approximated as

$$\gamma (\mathbf{C}^k \mathbf{v} + \mathbf{D}^k), \quad (\text{B.2})$$

where \mathbf{C}^k and \mathbf{D}^k are similar to \mathbf{A}^k and \mathbf{B}^k , with

$$\begin{aligned} \mathbf{C}_i^k &= \sum_{j \in n_i} \frac{(\mathbf{u}_i^{k+1})^2}{\|b_{\mathbf{u}_i^{k+1}}\|_2^2 \|b_{\mathbf{v}_j^k}\|_2^2}, \text{ and} \\ \mathbf{D}_i^k &= \sum_{j \in n_i} \frac{\mathbf{u}_i^{k+1} \sum_{l \in n_j, l \neq i} (\mathbf{u}_l^{k+1} \mathbf{v}_l^k)}{\|b_{\mathbf{u}_i^{k+1}}\|_2^2 \|b_{\mathbf{v}_j^k}\|_2^2}. \end{aligned} \quad (\text{B.3})$$

The derivatives of the first and the last term in Eq. (B.1) with respect to \mathbf{v} are $(\mathbf{v} + \mathbf{u}^{k+1} - \mathbf{f})$ and $\mu (\mathbf{v} - \operatorname{div} \mathbf{g}^{k+1} - \mathbf{m}_4^k)$, respectively. Adding these derivatives together and letting the sum be zero, we get the following equation:

$$(\mathbf{v} + \mathbf{u}^{k+1} - \mathbf{f}) + \mu (\mathbf{v} - \operatorname{div} \mathbf{g}^{k+1} - \mathbf{m}_4^k) + \gamma (\mathbf{C}^k \mathbf{v} + \mathbf{D}^k) = 0. \quad (\text{B.4})$$

This equation has the following closed-form solution

$$\mathbf{v}^{k+1} = \frac{\mathbf{f} - \mathbf{u}^{k+1} - \gamma \mathbf{D}(\mathbf{v}^k) + \mu (\operatorname{div} \mathbf{g}^{k+1} + \mathbf{m}_4^k)}{1 + \gamma C(\mathbf{v}^k) + \mu}, \quad (\text{B.5})$$

which we take as the approximate solution of the sub-problem (19).

Appendix C. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.jvcir.2022.103726>.

References

- [1] R. Xu, Y. Xu, Y. Quan, H. Ji, Cartoon-texture image decomposition using orientation characteristics in patch recurrence, SIAM J. Imaging Sci. 13 (3) (2020) 1179–1210.
- [2] Riya, B. Gupta, S.S. Lamba, Structure-aware adaptive bilateral texture filtering, Digit. Signal Process. 123 (2022) 103386.
- [3] Y. Liu, F. Zhang, Y. Zhang, X. Li, C. Zhang, Image smoothing based on histogram equalized content-aware patches and direction-constrained sparse gradients, Signal Process. 183 (2021) 108037.
- [4] M.K. Ng, X. Yuan, W. Zhang, Coupled variational image decomposition and restoration model for blurred cartoon-plus-texture images with missing pixels, IEEE Trans. Image Process. 22 (6) (2013) 2233–2246.
- [5] M. Li, W. Zhang, M. Xiao, C. Xu, Image decomposition and completion using relative total variation and schatten quasi-norm regularization, Neurocomputing 458 (2021) 639–654.
- [6] J. Lim, M. Heo, C. Lee, C.-S. Kim, Contrast enhancement of noisy low-light images based on structure-texture-noise decomposition, J. Vis. Commun. Image Represent. 45 (2017) 107–121.
- [7] Y. Meyer, Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures, Vol. 22, American Mathematical Soc., 2001.
- [8] L.I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (1–4) (1992) 259–268.
- [9] L.A. Vese, S.J. Osher, Modeling textures with total variation minimization and oscillating patterns in image processing, J. Sci. Comput. 19 (1) (2003) 553–572.
- [10] S. Osher, A. sole, L. Vese, Image decomposition and restoration using total variation minimization and the H^{-1} norm, Multiscale Model. Simul. 1 (3) (2003) 349–370.
- [11] J.F. Aujol, G. Gilboa, T.F. Chan, S. Osher, Structure-texture decomposition by a TV-Gabor model, in: International Workshop on Variational, Geometric, and Level Set Methods in Computer Vision, Springer, 2005, pp. 85–96.
- [12] J.-F. Aujol, G. Aubert, L. Blanc-Feraud, A. Chambolle, Image decomposition application to SAR images, in: International Conference on Scale-Space Theories in Computer Vision, Springer, 2003, pp. 297–312.
- [13] W. Yin, D. Goldfarb, S. Osher, Image cartoon-texture decomposition and feature selection using the total variation regularized L_1 functional, in: International Workshop on Variational, Geometric, and Level Set Methods in Computer Vision, Springer, 2005, pp. 73–84.
- [14] T. Chan, A. Marquina, P. Mulet, High-order total variation-based image restoration, SIAM J. Sci. Comput. 22 (2) (2000) 503–516.
- [15] J. Duan, Z. Qiu, W. Lu, G. Wang, Z. Pan, L. Bai, An edge-weighted second order variational model for image decomposition, Digit. Signal Process. 49 (2016) 162–181.
- [16] X. Liu, A new TGV-Gabor model for cartoon-texture image decomposition, IEEE Signal Process. Lett. 25 (8) (2018) 1221–1225.
- [17] X. Liu, Y. Chen, NLTV-Gabor-based models for image decomposition and denoising, Signal Image Video Process. 14 (2) (2020) 305–313.
- [18] L. Xu, Q. Yan, Y. Xia, J. Jia, Structure extraction from texture via relative total variation, ACM Trans. Graph. 31 (6) (2012) 1–10.
- [19] J.-L. Starck, Y. Moudden, J. Bobin, M. Elad, D. Donoho, Morphological component analysis, in: Wavelets XI, Vol. 5914, International Society for Optics and Photonics, 2005, 59140Q.
- [20] J.-L. Starck, D. Donoho, M. Elad, Redundant multiscale transforms and their application for morphological component separation, Adv. Imaging Electron. Phys. 132 (DAPNIA-2004-88) (2004).
- [21] G. Peyre, J. Fadili, J.-L. Starck, Learning the morphological diversity, SIAM J. Imaging Sci. 3 (3) (2010) 646–669.
- [22] H. Zhang, V. Patel, Convolutional sparse coding-based image decomposition, in: Proceedings of the British Machine Vision Conference (BMVC), 2016, pp. 125.1–125.11.
- [23] R. Xu, Y. Xu, Y. Quan, Structure-texture image decomposition using discriminative patch recurrence, IEEE Trans. Image Process. 30 (2021) 1542–1555.
- [24] H. Schaeffer, S. Osher, A low patch-rank interpretation of texture, SIAM J. Imaging Sci. 6 (1) (2013) 226–262.
- [25] T.-H. Ma, T.-Z. Huang, X.-L. Zhao, Group-based image decomposition using 3-D cartoon and texture priors, Inform. Sci. 328 (2016) 510–527.
- [26] Z. Zhang, H. He, A customized low-rank prior model for structured cartoon-texture image decomposition, Signal Process., Image Commun. 96 (2021) 116308.
- [27] S. Ono, T. Miyata, I. Yamada, Cartoon-texture image decomposition using blockwise low-rank texture characterization, IEEE Trans. Image Process. 23 (3) (2014) 1128–1142.
- [28] Y. Kim, B. Ham, M.N. Do, K. Sohn, Structure-texture image decomposition using deep variational priors, IEEE Trans. Image Process. 28 (6) (2018) 2692–2704.
- [29] F. Zhou, Q. Chen, B. Liu, G. Qiu, Structure and texture-aware image decomposition via training a neural network, IEEE Trans. Image Process. 29 (2019) 3458–3473.
- [30] J.-L. Starck, M. Elad, D.L. Donoho, Image decomposition via the combination of sparse representations and a variational approach, IEEE Trans. Image Process. 14 (10) (2005) 1570–1582.
- [31] R. Shahidi, C. Moloney, Decorrelating the structure and texture components of a variational decomposition model, IEEE Trans. Image Process. 18 (2) (2009) 299–309.
- [32] Q. Liu, J. Liu, P. Dong, D. Liang, SGTD: Structure gradient and texture decorrelating regularization for image decomposition, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1081–1088.
- [33] J. Xu, Y. Hao, X. Zhang, J. Zhang, A cartoon+texture image decomposition variational model based on preserving the local geometric characteristics, IEEE Access 8 (2020) 46574–46584.

- [34] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.* 3 (1) (2011) 1–122.
- [35] P.-Y. Chen, I.W. Selesnick, Translation-invariant shrinkage/thresholding of group sparse signals, *Signal Process.* 94 (2014) 476–489.
- [36] F. Sur, A non-local dual-domain approach to cartoon and texture decomposition, *IEEE Trans. Image Process.* 28 (4) (2018) 1882–1894.
- [37] K. Valkealahti, E. Oja, Reduced multidimensional co-occurrence histograms in texture classification, *IEEE Trans. Pattern Anal. Mach. Intell.* 20 (1) (1998) 90–94.
- [38] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (4) (2004) 600–612.