# Supplementary Materials

CONTENTS

## I. MATRIX-FORM OF THE PROPOSED MODEL

The proposed model in the main paper is given in a vectorized form, which is concise but may be a little bit hard to understand. To help to understand the proposed cartoon and texture regularization terms, a matrix-form formulation of our model is given in the follow. Let $\{\boldsymbol{G}_m\}_{m=1}^{M}$ denote a set of 2D wavelet filters and $\{\boldsymbol{H}_q\}_{q=1}^{Q}$ denote a set of 2D DCT filters. Given a gray-scale image $\boldsymbol{F} \in \mathbb{R}^{H\times W}$, our goal is to extract its cartoon layer $\boldsymbol{U}^{H\times W}$ and texture layer $\boldsymbol{V}^{H\times W}$. Let $\mathcal{L}^{(d)}$ denote the directional nonlocal operator for the $d$-th band, which is defined as

$$\mathcal{L}^{(d)}(\boldsymbol{X}) = \mathcal{V}^{-1}(\tilde{\boldsymbol{L}}^{(d)}\mathcal{V}(\boldsymbol{X})),$$

where $\mathcal{V} : \mathbb{R}^{H\times W} \to \mathbb{R}^{HW}$ denotes the vectorization operator that sequentially concatenates the columns of an input image $\boldsymbol{X} \in \mathbb{R}^{HW}$, $\mathcal{V}^{-1} : \mathbb{R}^{HW} \to \mathbb{R}^{H\times W}$ denotes the inversion of $\mathcal{V}$ which reshapes an input vector to a matrix-form image, and $\tilde{\boldsymbol{L}}^{(d)}$ is the directional nonlocal transform defined by Equation (5) in the main paper. Then the proposed decomposition model defined by Equation (9) in the main paper can be rewritten in the matrix form as follows:

$$\begin{aligned}&\min_{\boldsymbol{U},\boldsymbol{V}} \sum_{m=1}^{M}\|\boldsymbol{\Lambda}_1 \odot (\boldsymbol{G}_m * \boldsymbol{U})\|_1 + \sum_{d=1}^{D}\sum_{q=1}^{Q}\|\boldsymbol{\Lambda}_2 \odot (\mathcal{L}^{(d)}(\boldsymbol{H}_q * \boldsymbol{V}))\|_1\\ &\text{s.t.}\ \ \boldsymbol{U} + \boldsymbol{V} = \boldsymbol{F},\end{aligned}$$

where $\|\cdot\|_1$ denotes the sum of absolute value of the elements in a matrix, $\odot$ denotes the element-wise product, $*$ denotes the 2D discrete convolution, and $\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2$ are the weighting matrices for the cartoon and texture layers respectively. We have $\boldsymbol{\Lambda_1} = \mathcal{V}^{-1}(\bar{\boldsymbol{\lambda}}_1)$ and $\boldsymbol{\Lambda_2} = \mathcal{V}^{-1}(\bar{\boldsymbol{\lambda}}_2)$ where $\bar{\boldsymbol{\lambda}}_1, \bar{\boldsymbol{\lambda}}_2$ are the weighting parameter vectors calculated by the scheme (Equation (12)) proposed in Section IV-B in the main paper.

## II. THE 2D WAVELET FILTER USED BY THE PROPOSED MODEL

$$\boldsymbol{G}_1 = \tfrac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix},\quad \boldsymbol{G}_2 = \tfrac{\sqrt{2}}{16}\begin{bmatrix}1&2&1\\0&0&0\\-1&-2&-1\end{bmatrix},\quad \boldsymbol{G}_3 = \tfrac{1}{16}\begin{bmatrix}-1&-2&-1\\2&4&2\\-1&-2&-1\end{bmatrix},$$

$$\boldsymbol{G}_4 = \tfrac{\sqrt{2}}{16}\begin{bmatrix}1&0&-1\\2&0&-2\\1&0&-1\end{bmatrix},\quad \boldsymbol{G}_5 = \tfrac{1}{8}\begin{bmatrix}1&0&-1\\0&0&0\\-1&0&1\end{bmatrix},\quad \boldsymbol{G}_6 = \tfrac{\sqrt{2}}{16}\begin{bmatrix}-1&0&1\\2&0&-2\\-1&0&1\end{bmatrix},$$

$$\boldsymbol{G}_7 = \tfrac{1}{16}\begin{bmatrix}-1&2&-1\\-2&4&-2\\-1&2&-1\end{bmatrix},\quad \boldsymbol{G}_8 = \tfrac{\sqrt{2}}{16}\begin{bmatrix}-1&2&-1\\0&0&0\\1&-2&1\end{bmatrix},\quad \boldsymbol{G}_9 = \tfrac{1}{16}\begin{bmatrix}1&-2&1\\-2&4&-2\\1&-2&1\end{bmatrix}.$$

## III. Pseudo code of the proposed method

**Algorithm 1:** Pseudo code of the proposed method

**INPUT:** Image $\boldsymbol{f}$

**OUTPUT:** Cartoon layer $\boldsymbol{u}$, Texture layer $\boldsymbol{v}$

**STAGE 1:** Directional Patch Matching Scheme

- **foreach** *patch* $\boldsymbol{p}_i$ *from image* $\boldsymbol{f}$ **do**
  - **for** $d = 1, .., D$ **do**
    - Find the index set of the top-$K$ similar patches in the $d$-th directional banded region $\mathbb{N}_{\boldsymbol{p}_i}^{(d)}$ and store it as $\mathbb{S}_i^{(d)}$.
- **for** $d = 1, ...D$ **do**
  - Calcalute the directional nonlocal transform for the $d$-th direction as

$$\tilde{\boldsymbol{L}}^{(d)}(i,j) = \begin{cases} 1, & i = j; \\ -\frac{\omega(\boldsymbol{p}_i, \boldsymbol{p}_j)}{\sum_{k \in \mathbb{S}_i^{(d)}} \omega(\boldsymbol{p}_i, \boldsymbol{p}_k)}, & j \in \mathbb{S}_i^{(d)}; \\ 0, & \text{otherwise}, \end{cases}$$

- Constructed the nonlocal sparsification transform as $\boldsymbol{L} = [\tilde{\boldsymbol{L}}^{(1)}; \cdots; \tilde{\boldsymbol{L}}^{(D)}]$.

**STAGE 2:** Calculation of regularization parameters

- **for** $k = 1, ..., N$ **do**
  - $\phi(k) = \frac{1}{DQ} \sum_{d=1}^{D} \sum_{i=1}^{Q} |(\tilde{\boldsymbol{L}}^{(d)} \mathcal{S}_{\boldsymbol{H}_i} \boldsymbol{f})(k)|^2$
- Calculate $\bar{\boldsymbol{\lambda}}_1$,$\bar{\boldsymbol{\lambda}}_2$ as $\bar{\boldsymbol{\lambda}}_1(k) = 1 + e^{-\frac{\phi(k)}{\eta}}, \bar{\boldsymbol{\lambda}}_2(k) = 1 - e^{-\frac{\phi(k)}{\eta}}$,
- Construct $\boldsymbol{\lambda}_1$,$\boldsymbol{\lambda}_2$ as $\boldsymbol{\lambda}_1 = [\underbrace{\beta_1 \bar{\boldsymbol{\lambda}}_1; \cdots; \beta_1 \bar{\boldsymbol{\lambda}}_1}_{M}], \quad \boldsymbol{\lambda}_2 = [\underbrace{\beta_2 \bar{\boldsymbol{\lambda}}_2; \cdots; \beta_2 \bar{\boldsymbol{\lambda}}_2}_{DQ}]$.

**STAGE 3:** Numerical algorithm

- Construct $\boldsymbol{W} = [\mathcal{S}_{\boldsymbol{G}_1}; \cdots; \mathcal{S}_{\boldsymbol{G}_M}]$ and $\boldsymbol{J} = [\boldsymbol{L}\mathcal{S}_{\mathcal{H}_1}; \cdots; \boldsymbol{L}\mathcal{S}_{\boldsymbol{H}_Q}]$,
- Construct the system $\boldsymbol{D} = [\boldsymbol{W}\ \boldsymbol{0};\ \boldsymbol{0}\ \boldsymbol{J}]$, $\boldsymbol{A} = [\boldsymbol{I}\ \boldsymbol{I}]$, $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1^\top, \boldsymbol{\lambda}_2^\top]^\top$ and solve

$$\boldsymbol{x}^* = \min_{\boldsymbol{x}} \|\text{diag}(\boldsymbol{\lambda})\boldsymbol{D}\boldsymbol{x}\|_1, \quad \text{s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{f}$$

$$\text{or} \quad \boldsymbol{x}^* = \min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{f}\|_2^2 + \|\text{diag}(\boldsymbol{\lambda})\boldsymbol{D}\boldsymbol{x}\|_1$$

  with the numerical scheme described in Section IV-C of the main paper.
- **Return** $\boldsymbol{u} = [\boldsymbol{I}, \boldsymbol{0}]\ \boldsymbol{x}^*$ and $\boldsymbol{v} = [\boldsymbol{0}, \boldsymbol{I}]\ \boldsymbol{x}^*$.

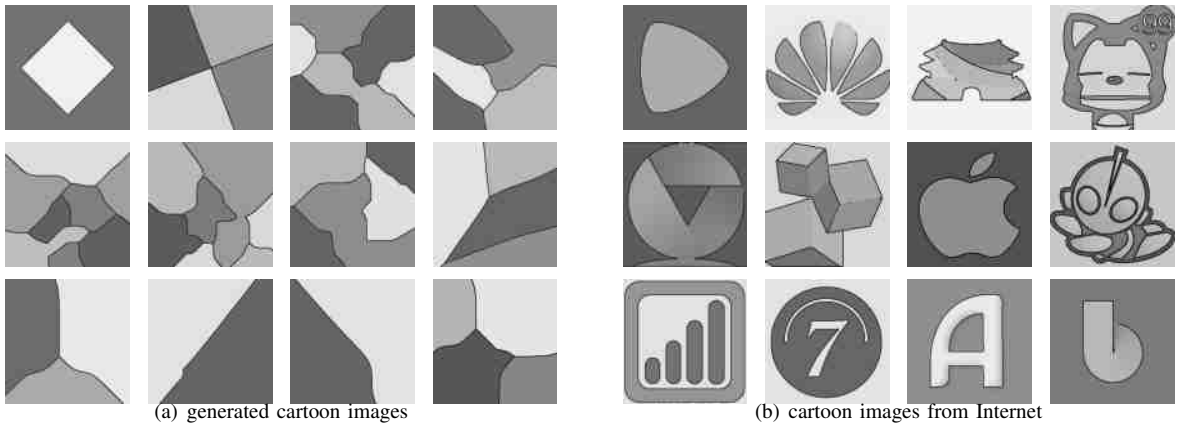
## IV. Synthesized images for evaluation

For quantitative evaluation, we synthesized 100 images as follows.

**Cartoon layers.** Regarding the truth cartoon layers, we generated some piece-wise constant images using the following scheme: (i) randomly/manually select several seed points; (ii) divide the image pixels into several non-overlapping groups, which is done via the spatially-nearest-neighbor clustering using the seed points as centers, with the $p$th-order Minkowski distance; (iii) assign a unique pixel value to each group of pixels and add edges at the boundaries. See Fig. 1(a) for some sample images of the generated cartoon layers. Note that the region boundaries are line segments when $p = 2$ and become curves when $p > 2$. Besides the generated ones, we also include some cartoon images from Internet, such as icons, logos and cartoon characters. See Fig. 1(b) for some examples.

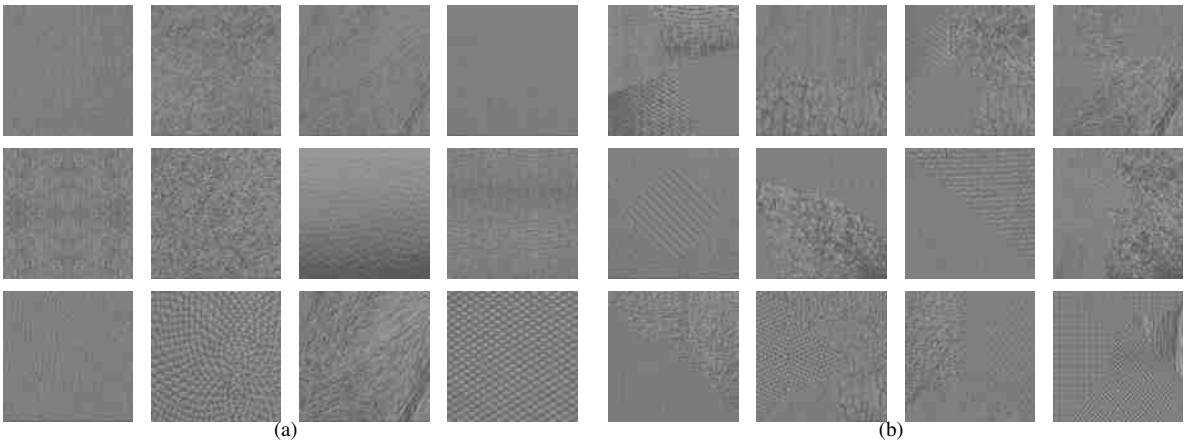
**Texture layers.** Regarding the truth texture layers, we picked up some pure texture images from several texture image datasets, including Brodatz [1], Kylberg [2], KTH-TIPS [3] and DTD [4]. See Fig. 2 for some examples.

**Synthesis schemes.** Two schemes were used to synthesize the test images from the truth cartoon layers and texture layers: (i) weighted average of two layers; (2) using a randomly-generated cartoon layer as the mask, assign different texture layers to different cartoon regions. See Fig. 3 for some examples of the synthesized images.



(a) generated cartoon images (b) cartoon images from Internet

Fig. 1. Examples of ground-truth cartoon layers.



(a) (b)

Fig. 2. Examples of ground-truth texture layers.

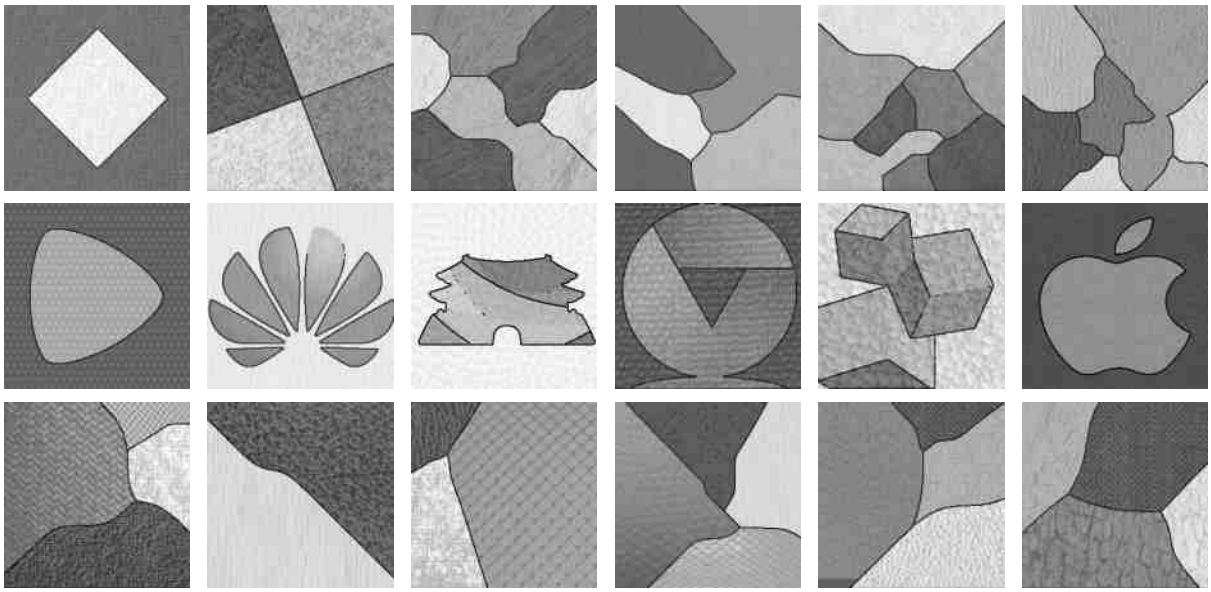


Fig. 3. Examples of synthesized images.

## V. More results on noiseless decomposition on synthesized images

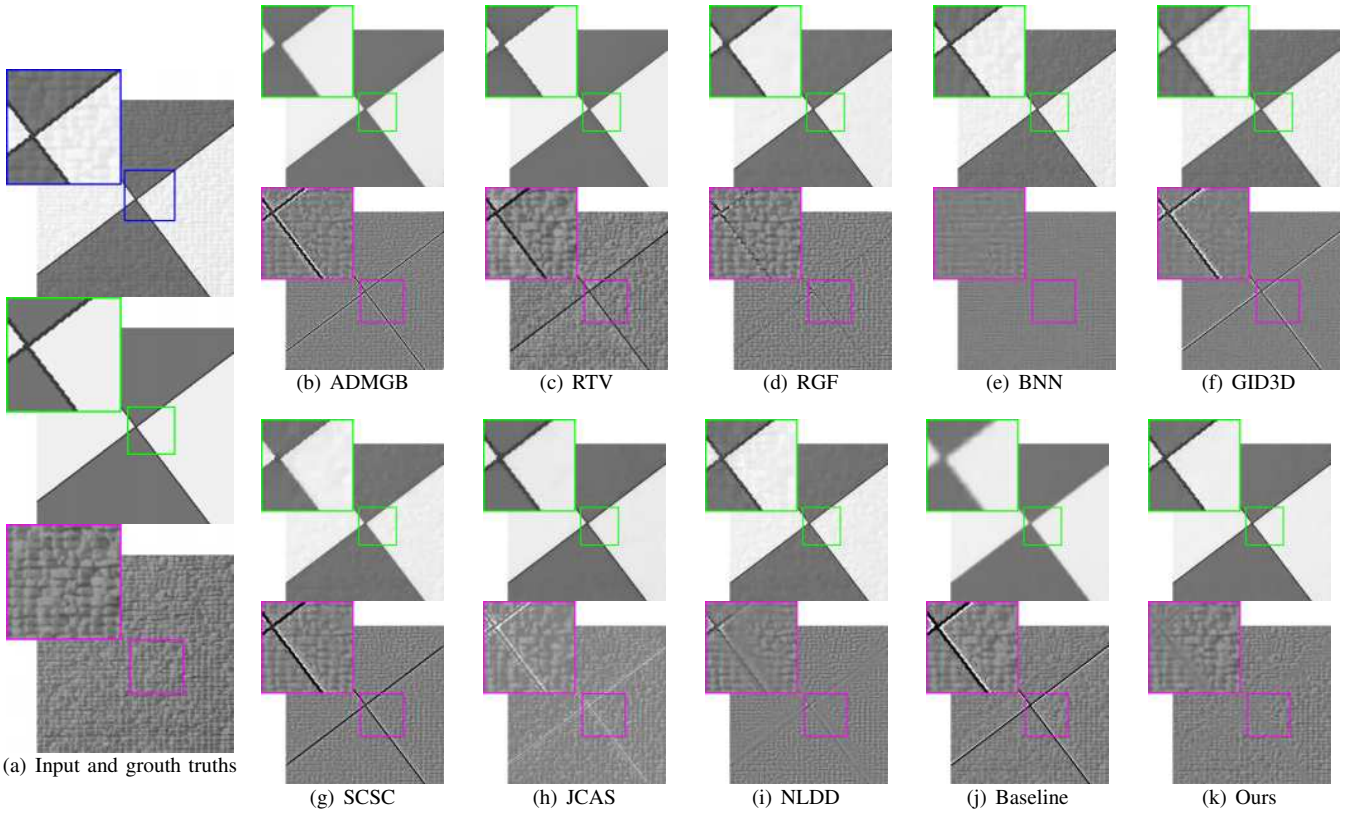



Fig. 4. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'test1 (Cross)'. (a) Input & truth. (b)-(k) Results of different methods.

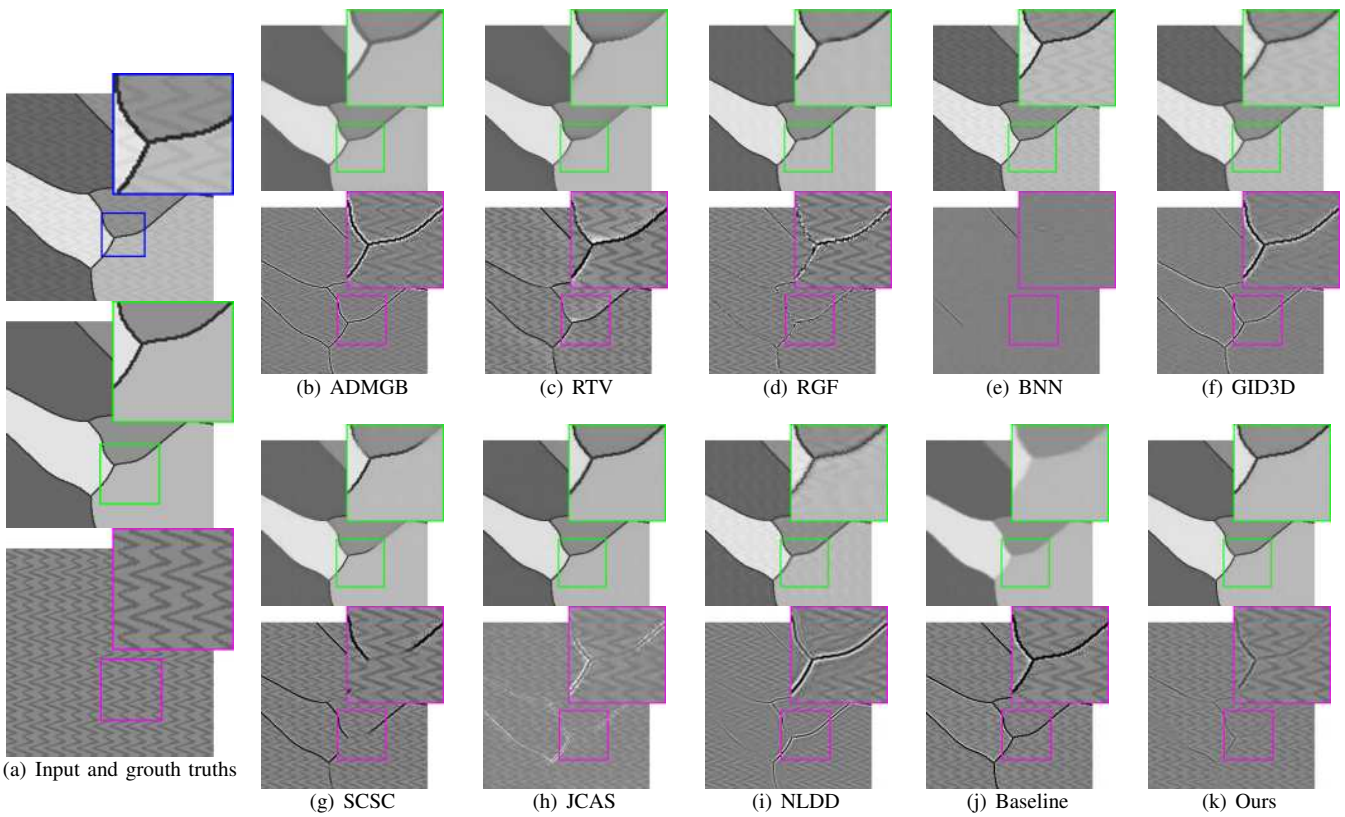



Fig. 5. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'test2'. (a) Input & truth. (b)-(k) Results of different methods.

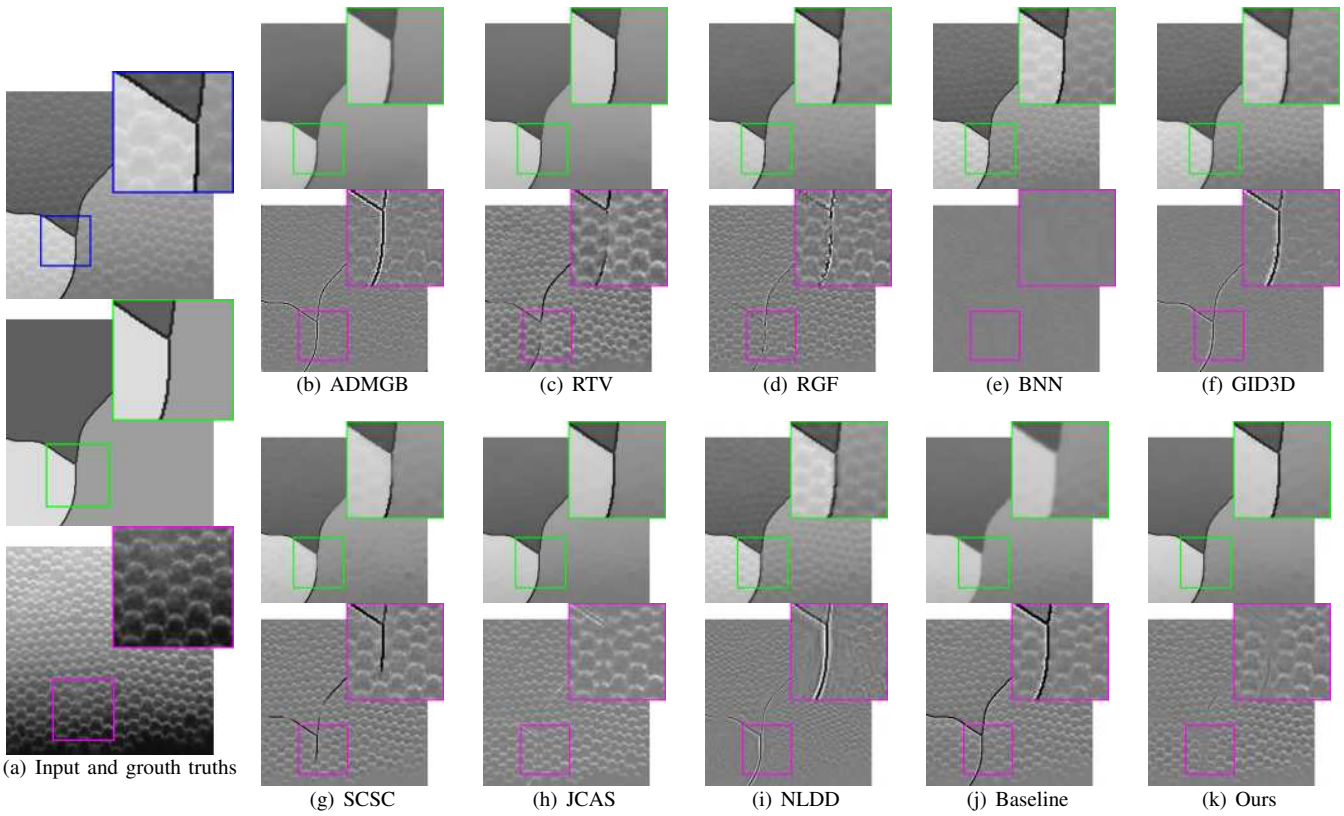



Fig. 6. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'test3'. (a) Input & truth. (b)-(k) Results of different methods.

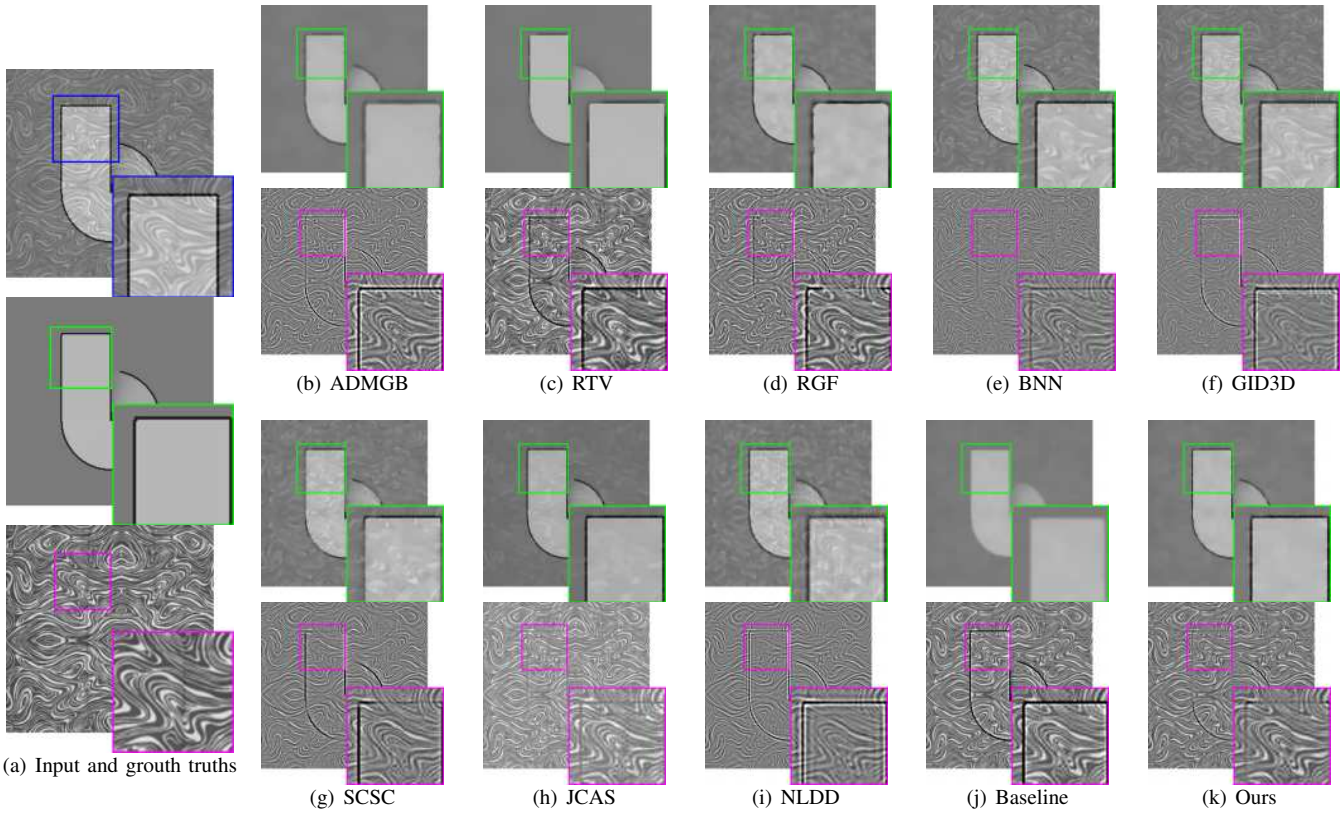



Fig. 7. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'test4'. (a) Input & truth. (b)-(k) Results of different methods.

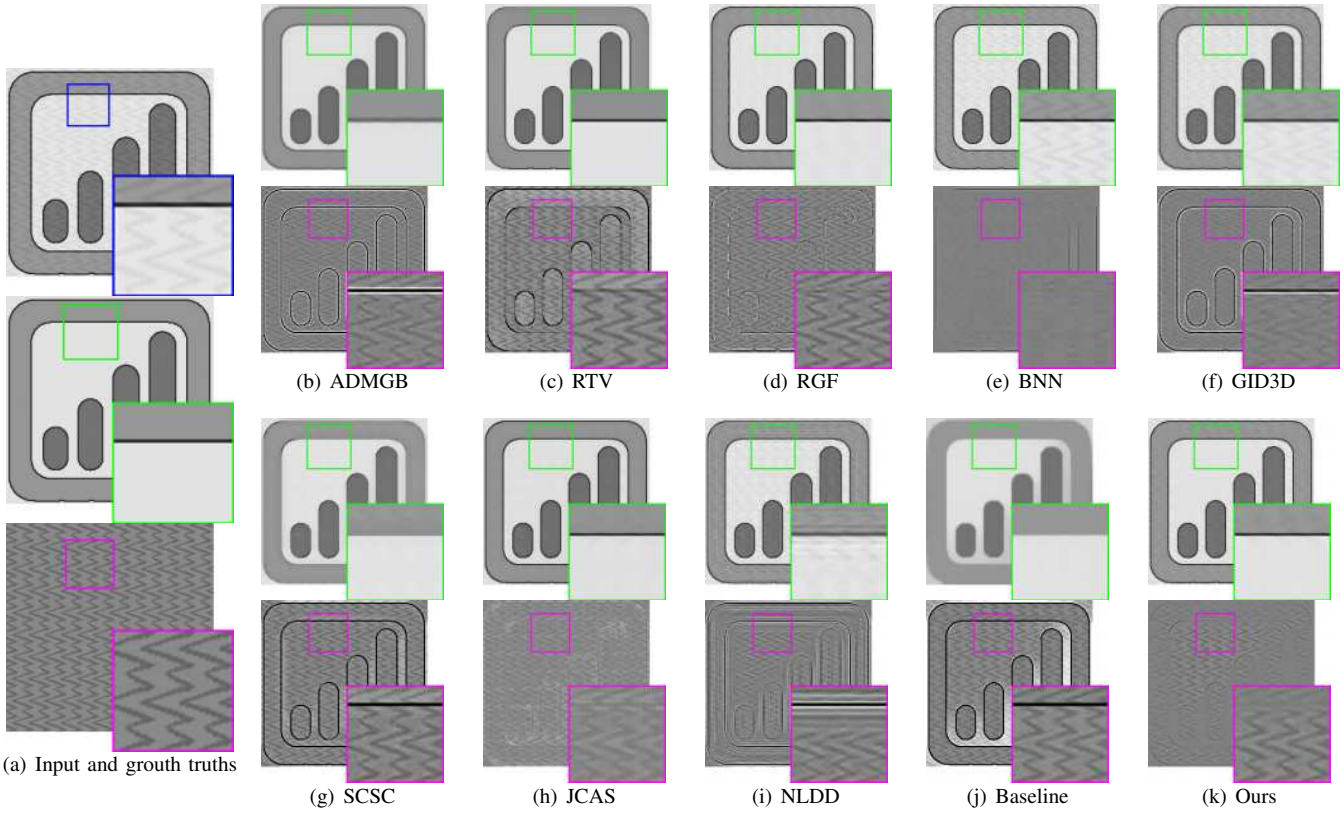



Fig. 8. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'test5'. (a) Input & truth. (b)-(k) Results of different methods.

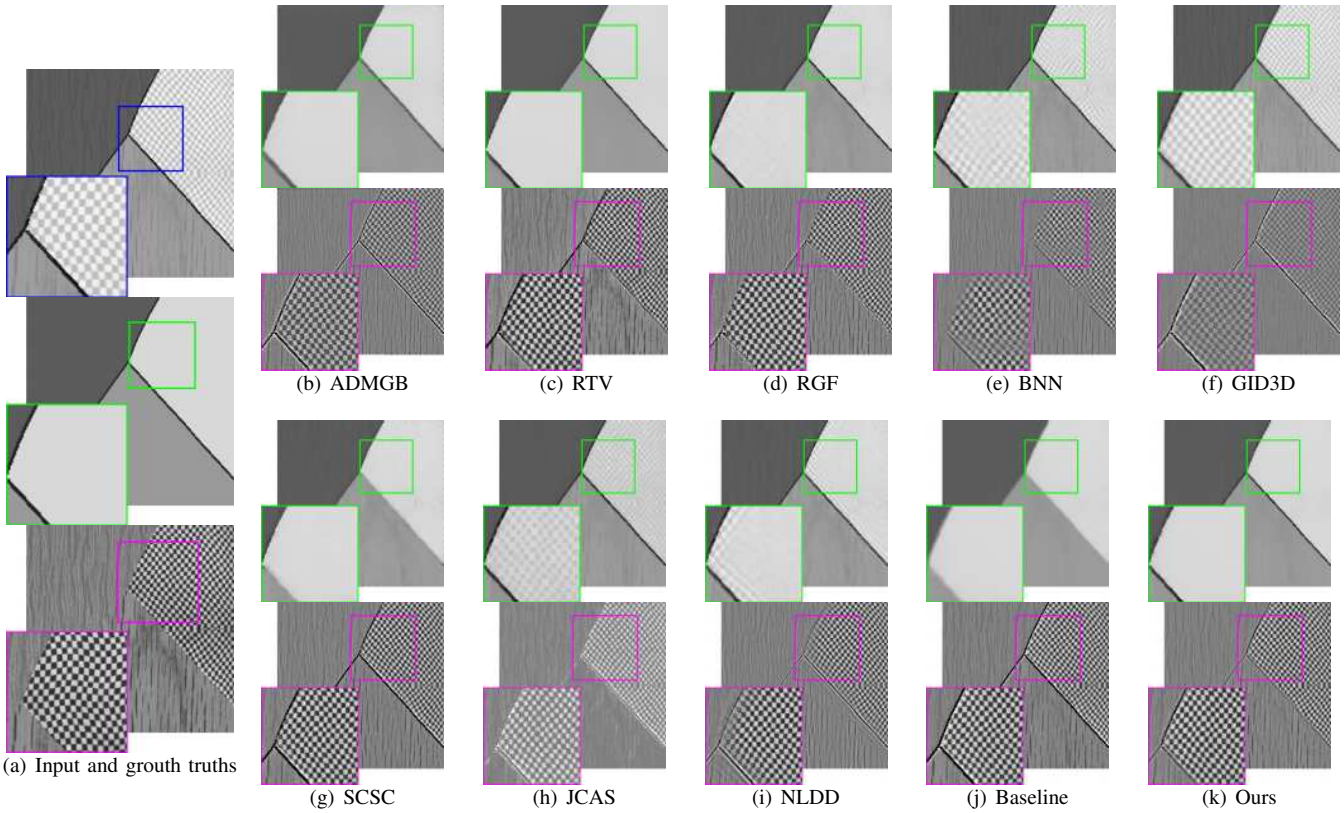
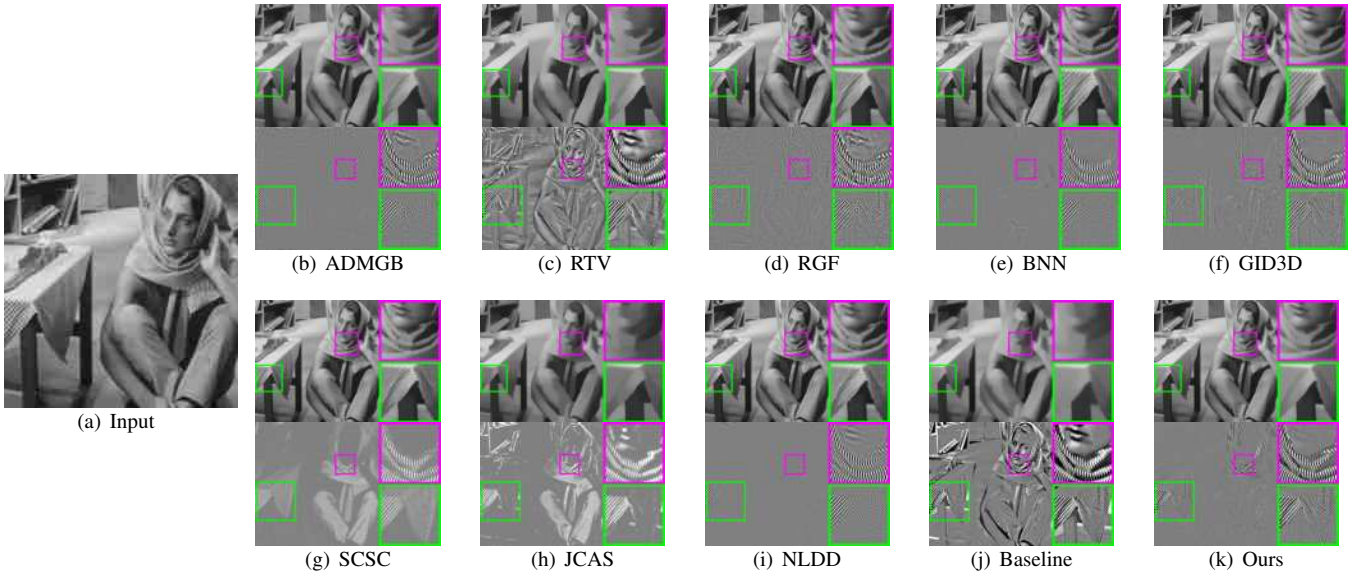


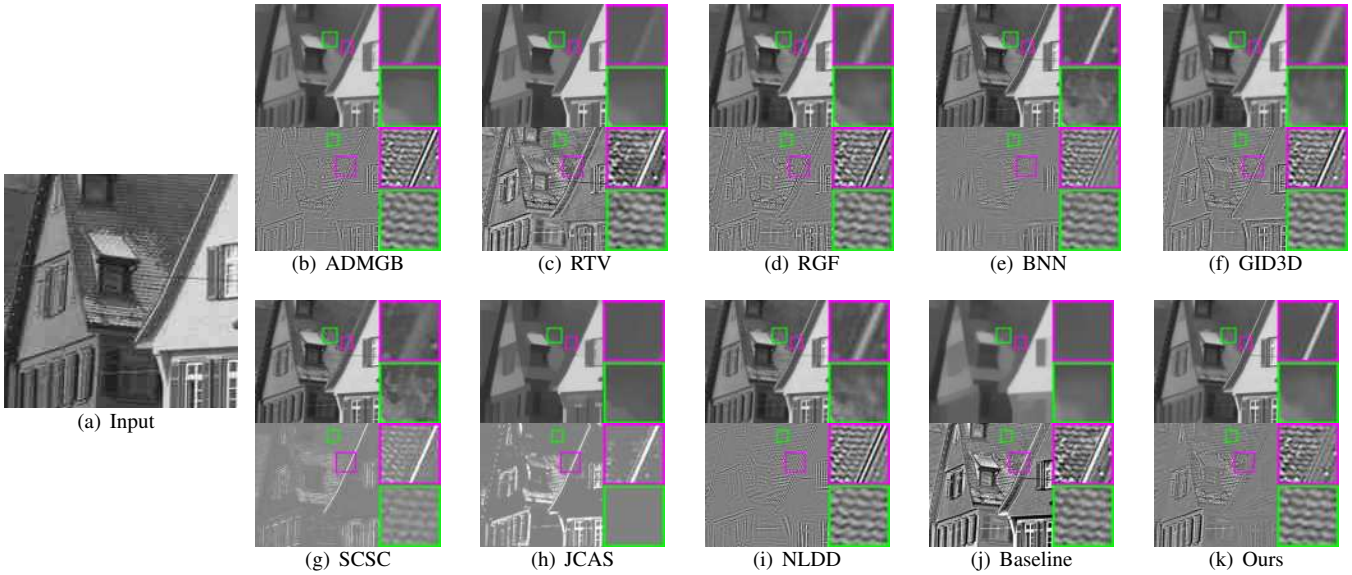

Fig. 9. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'test6'. (a) Input & truth. (b)-(k) Results of different methods.

## VI. More results on natural images



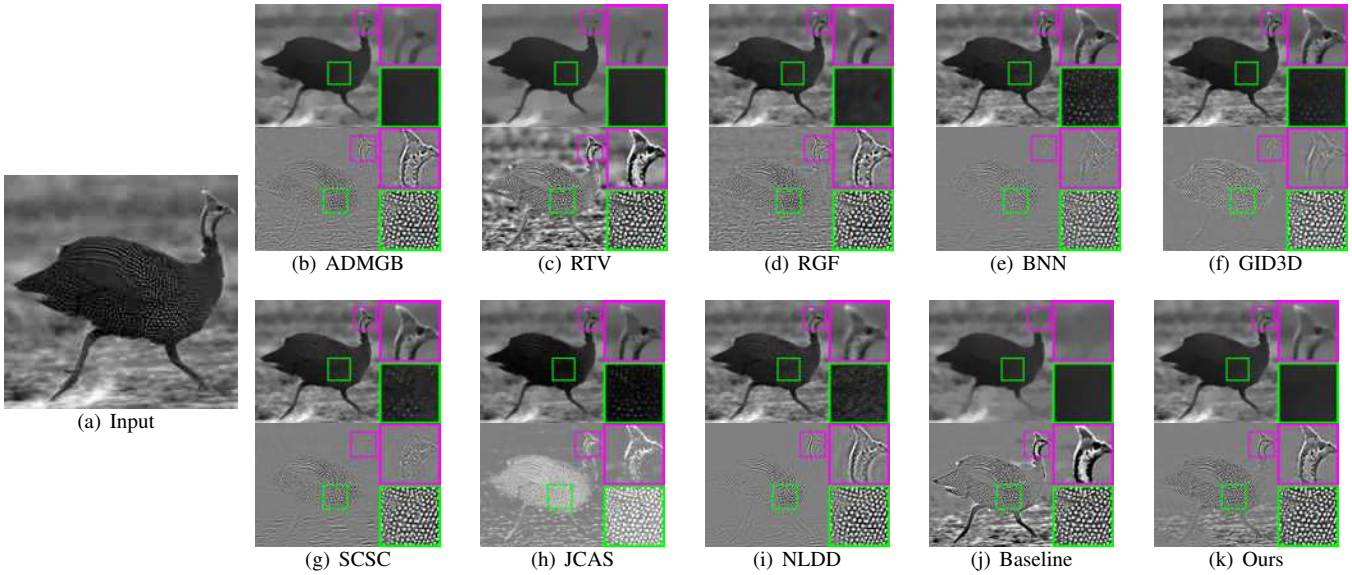
(a) Input (b) ADMGB (c) RTV (d) RGF (e) BNN (f) GID3D (g) SCSC (h) JCAS (i) NLDD (j) Baseline (k) Ours

Fig. 10. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'Barbara'. (a) Ground truth. (b)-(k) Results of different methods.



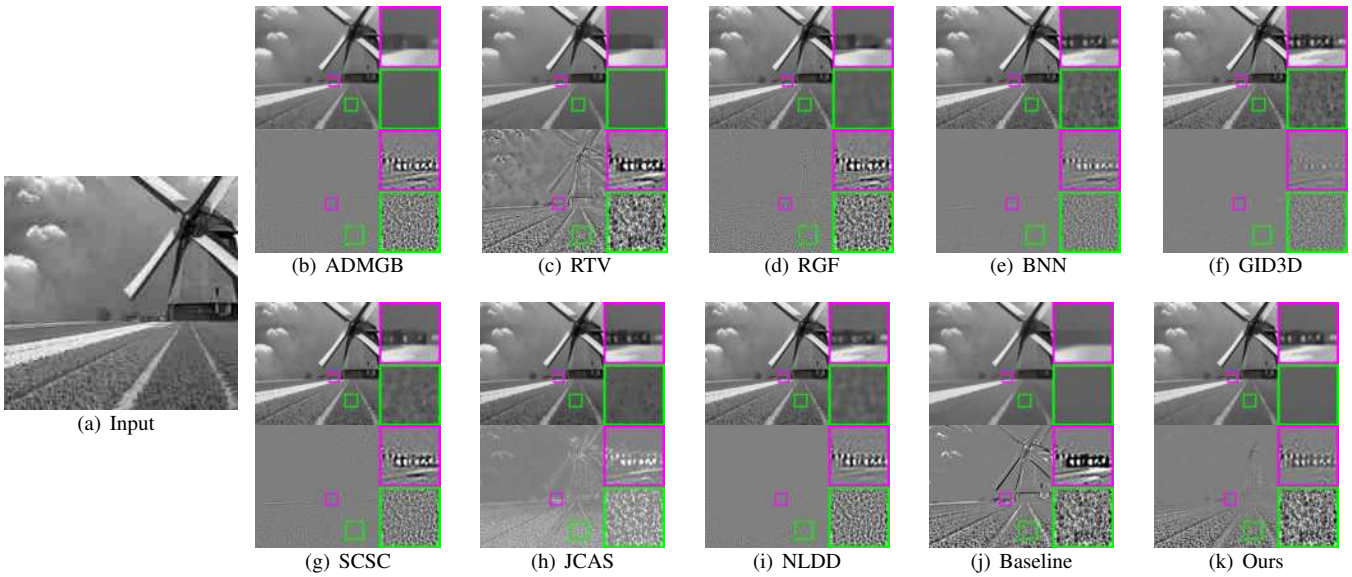
(a) Input (b) ADMGB (c) RTV (d) RGF (e) BNN (f) GID3D (g) SCSC (h) JCAS (i) NLDD (j) Baseline (k) Ours

Fig. 11. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'House'. (a) Ground truth. (b)-(k) Results of different methods.



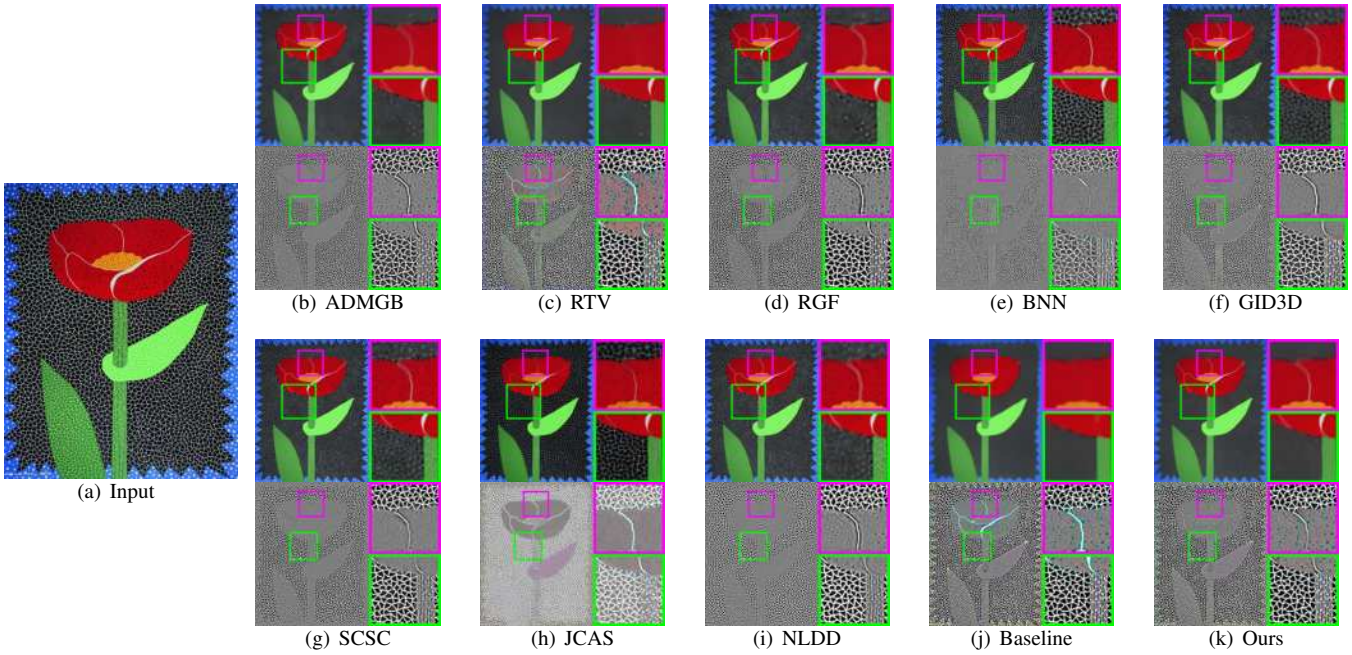
(a) Input (b) ADMGB (c) RTV (d) RGF (e) BNN (f) GID3D (g) SCSC (h) JCAS (i) NLDD (j) Baseline (k) Ours

Fig. 12. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'Guinea'. (a) Ground truth. (b)-(k) Results of different methods.



(a) Input (b) ADMGB (c) RTV (d) RGF (e) BNN (f) GID3D (g) SCSC (h) JCAS (i) NLDD (j) Baseline (k) Ours

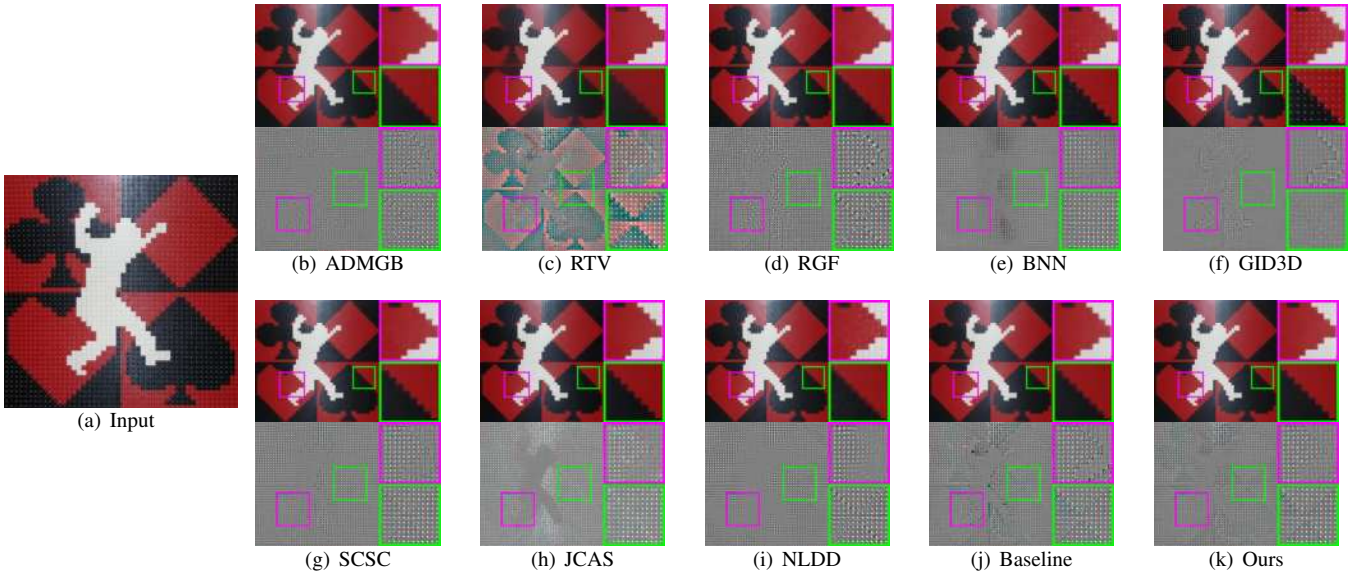
Fig. 13. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'Windmill'. (a) Ground truth. (b)-(k) Results of different methods.



(a) Input (b) ADMGB (c) RTV (d) RGF (e) BNN (f) GID3D

(g) SCSC (h) JCAS (i) NLDD (j) Baseline (k) Ours

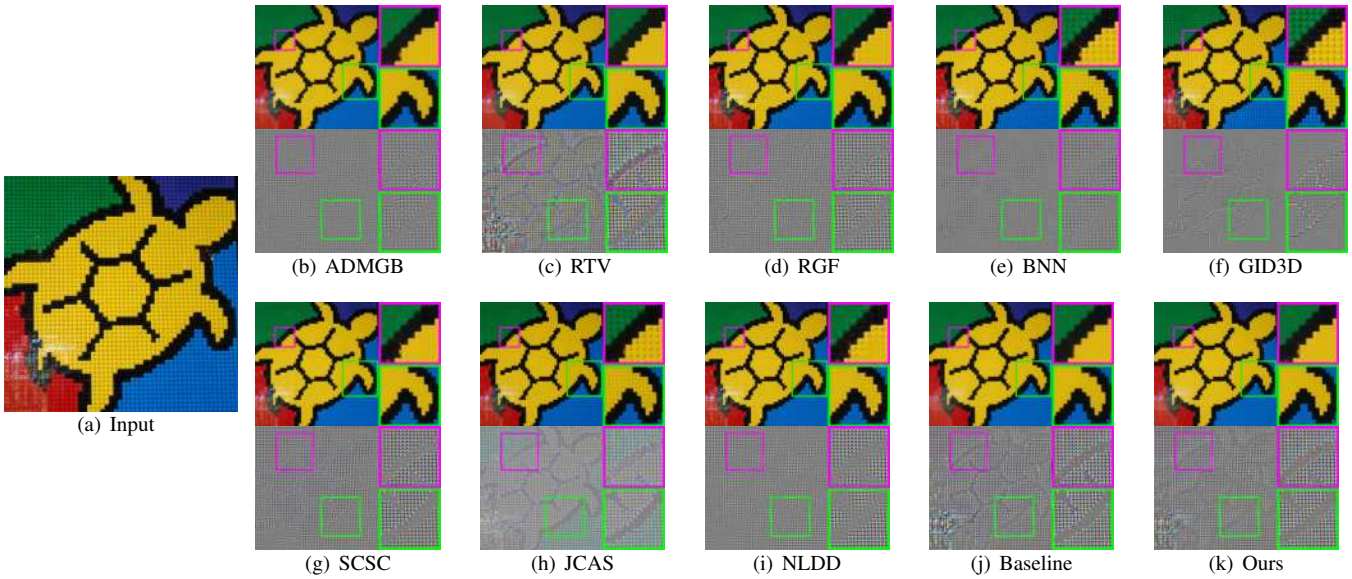
Fig. 14. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'Flower'. (a) Ground truth. (b)-(k) Results of different methods.



(a) Input (b) ADMGB (c) RTV (d) RGF (e) BNN (f) GID3D

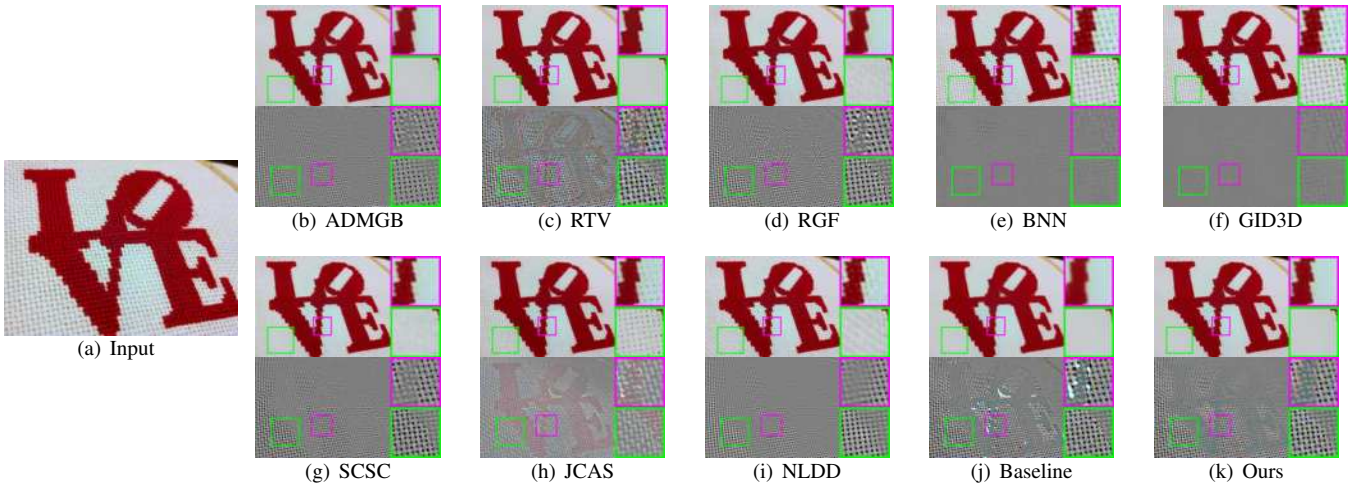
(g) SCSC (h) JCAS (i) NLDD (j) Baseline (k) Ours

Fig. 15. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'Poker'. (a) Ground truth. (b)-(k) Results of different methods.



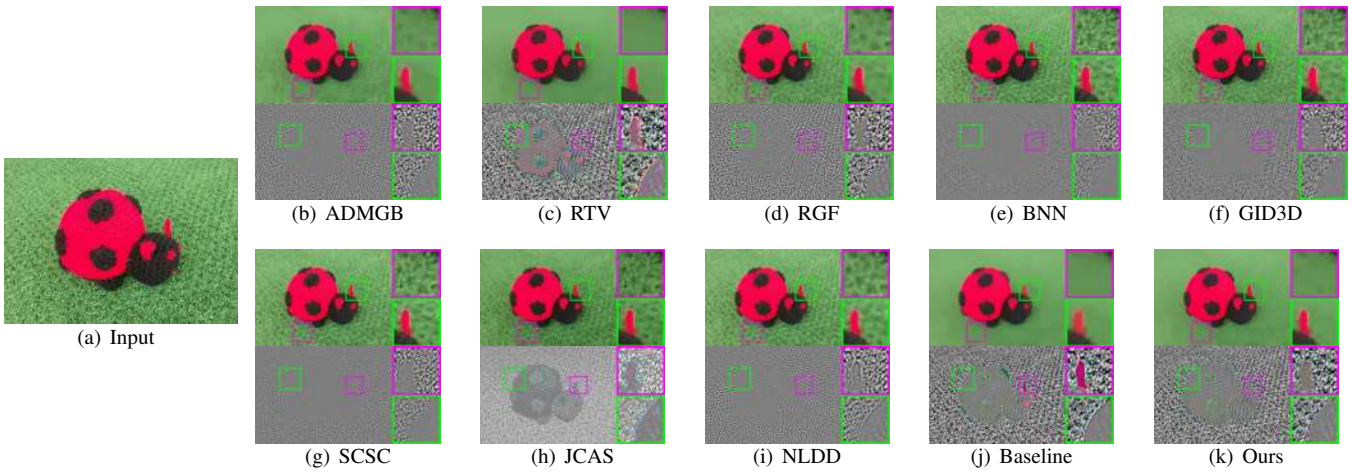
(a) Input (b) ADMGB (c) RTV (d) RGF (e) BNN (f) GID3D (g) SCSC (h) JCAS (i) NLDD (j) Baseline (k) Ours

Fig. 16. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'Tortoise'. (a) Ground truth. (b)-(k) Results of different methods.



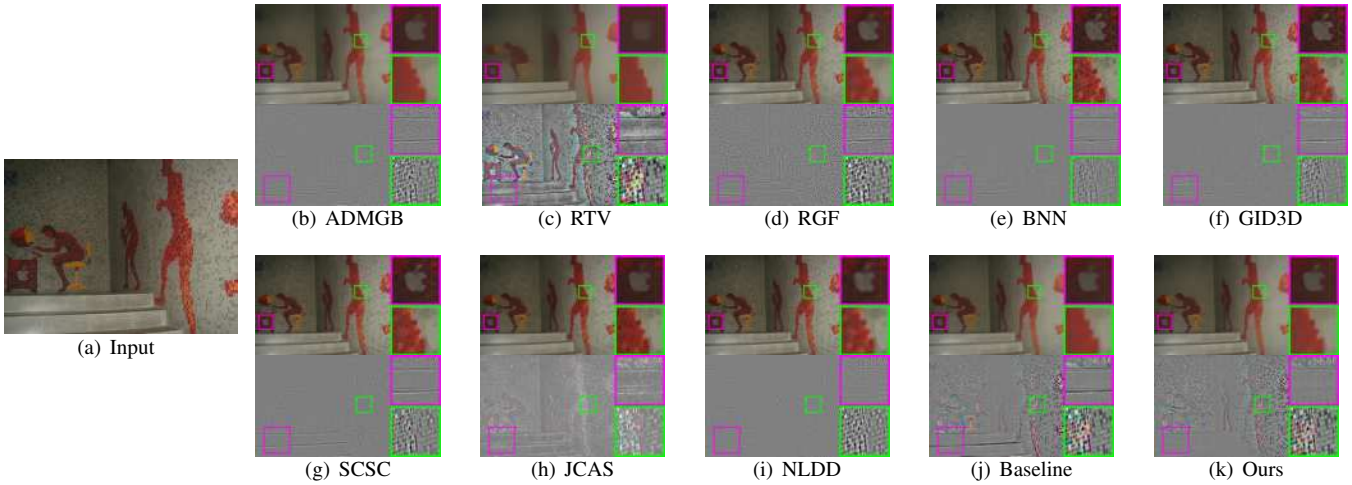
(a) Input (b) ADMGB (c) RTV (d) RGF (e) BNN (f) GID3D (g) SCSC (h) JCAS (i) NLDD (j) Baseline (k) Ours

Fig. 17. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'Love'. (a) Ground truth. (b)-(k) Results of different methods.



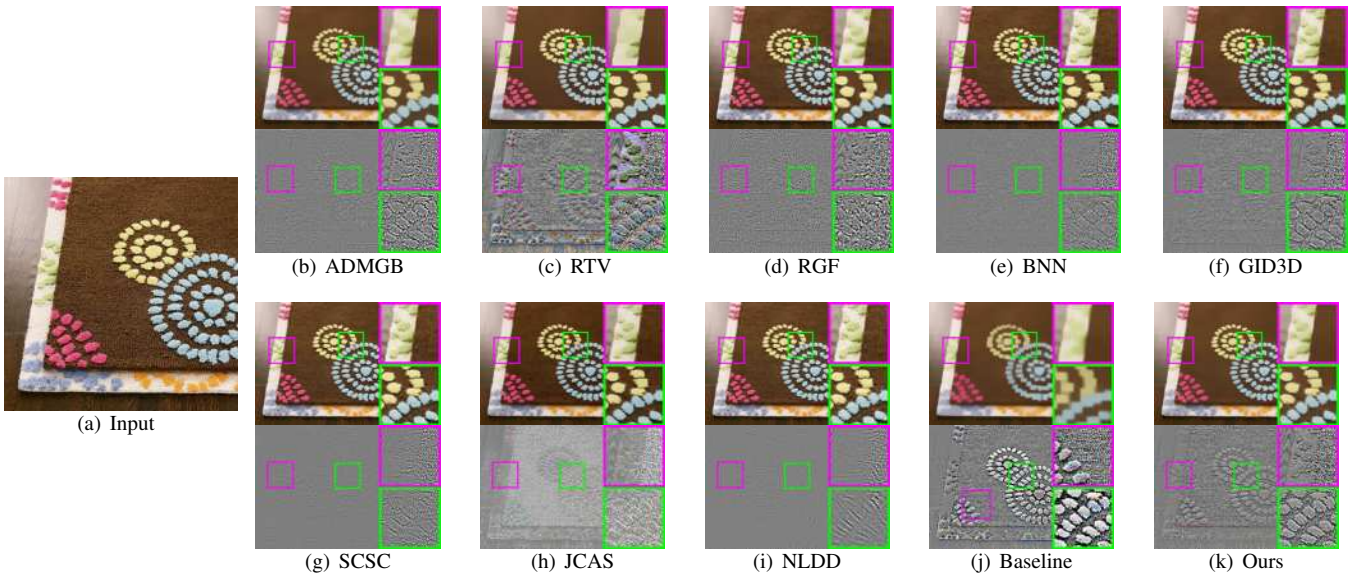
(a) Input (b) ADMGB (c) RTV (d) RGF (e) BNN (f) GID3D (g) SCSC (h) JCAS (i) NLDD (j) Baseline (k) Ours

Fig. 18. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'Ladybird'. (a) Ground truth. (b)-(k) Results of different methods.



(a) Input (b) ADMGB (c) RTV (d) RGF (e) BNN (f) GID3D (g) SCSC (h) JCAS (i) NLDD (j) Baseline (k) Ours

Fig. 19. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'Wall'. (a) Ground truth. (b)-(k) Results of different methods.

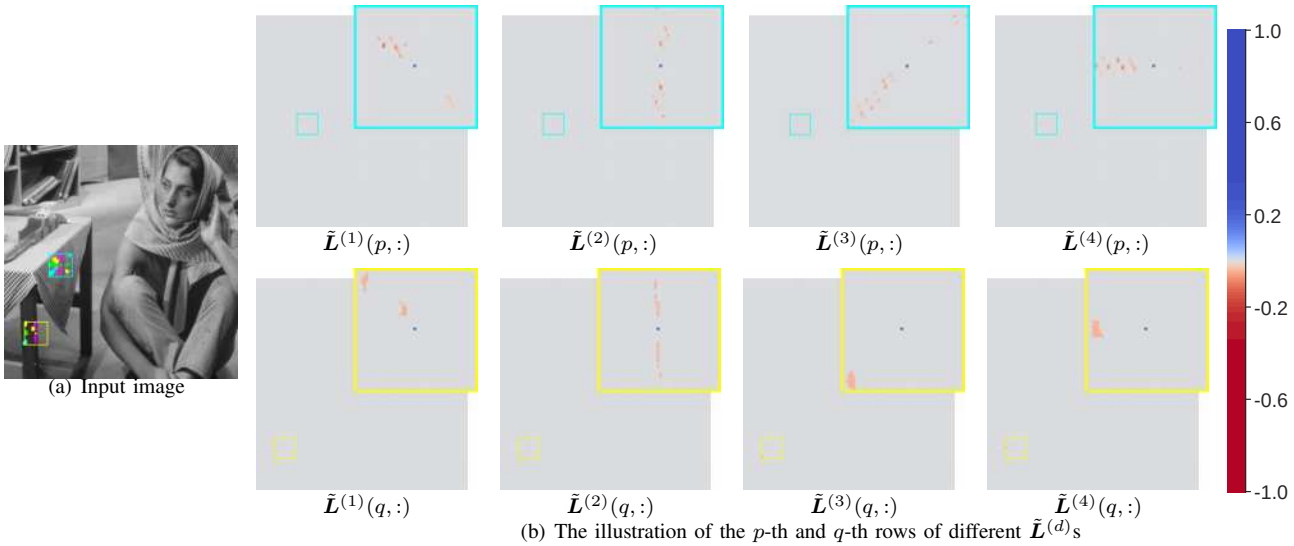


(a) Input (b) ADMGB (c) RTV (d) RGF (e) BNN (f) GID3D (g) SCSC (h) JCAS (i) NLDD (j) Baseline (k) Ours

Fig. 20. Cartoon layers (top) and texture layers (bottom) of the decomposition results on 'Carpet'. (a) Ground truth. (b)-(k) Results of different methods.

## VII. Visual illustration of matrix $\tilde{\boldsymbol{L}}^{(d)}$

This section is for better understanding the construction and structure of $\tilde{\boldsymbol{L}}^{(d)}$. In Figure 21(a), we show two example patches and their matching results in an image $\boldsymbol{f}$. One example patch locates in a textured region (*i.e.* the patch in cyan box), and its corresponding index in $\tilde{\boldsymbol{L}}^{(d)}$ is denoted by $p$. The other example patch locates on a contour edge (*i.e.* the patch in the yellow box), and its corresponding index in $\tilde{\boldsymbol{L}}^{(d)}$ is denoted by $q$. The corresponding rows of these two examples patches in $\tilde{\boldsymbol{L}}^{(d)}$, *i.e.* $\tilde{\boldsymbol{L}}^{(d)}(p,:)$ and $\tilde{\boldsymbol{L}}^{(d)}(q,:)$ for $d=1,2,3,4$, are reshaped to 2D images and illustrated in Figure 21(b). Regarding the example patch in the texture region, its similar patches can be found in all directions. Therefore, it can be seen that the weights with large magnitude in the corresponding rows of $\tilde{\boldsymbol{L}}^{(d)}$ all relate to the positions of similar patches, making $\tilde{\boldsymbol{L}}^{(d)}\boldsymbol{f}$ close to $\mathbf{0}$. In the contrary, regarding the example patch on the contour edge, similar patches can only be found in one direction. Therefore, the weights with large magnitude in the corresponding rows of $\tilde{\boldsymbol{L}}^{(d)}$s relate to the position of dissimilar patches when $d=2,3,4$, leading to large $\tilde{\boldsymbol{L}}^{(d)}\boldsymbol{f}$ for $d=2,3,4$.




(a) Input image

(b) The illustration of the $p$-th and $q$-th rows of different $\tilde{\boldsymbol{L}}^{(d)}$s

Fig. 21. Visual illustration of the structure of $\boldsymbol{L}^{(d)}$ on two example patches locating in texture region and contour edge respectively.

## VIII. Influence of replacing $\ell_1$ norm with $\ell_0$ norm

It is interesting to see the how our model performs when using the $\ell_0$ norm for measuring the sparsity. We replace the $\ell_1$ norm with the $\ell_0$ norm in our model, and their performance is compared on the decomposition of synthetic images. See Table I for the results. The results produced by the algorithm with $\ell_0$ norm are slightly worse than that produced by the one with $\ell_1$ norm. The inferior performance of $\ell_0$ norm may be due to the difficulty of optimizing the $\ell_0$-norm-related problems.

TABLE I
Average PSNR (dB) and SSIM values of the decomposition results on synthetic images with different sparsity measures

| Measure | Cartoon | | Texture | |
|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM |
| $\ell_0$-norm | 33.37 | 0.975 | 33.20 | 0.933 |
| $\ell_1$-norm | 33.48 | 0.977 | 33.32 | 0.938 |

## IX. Visual results from the influence test on patch size

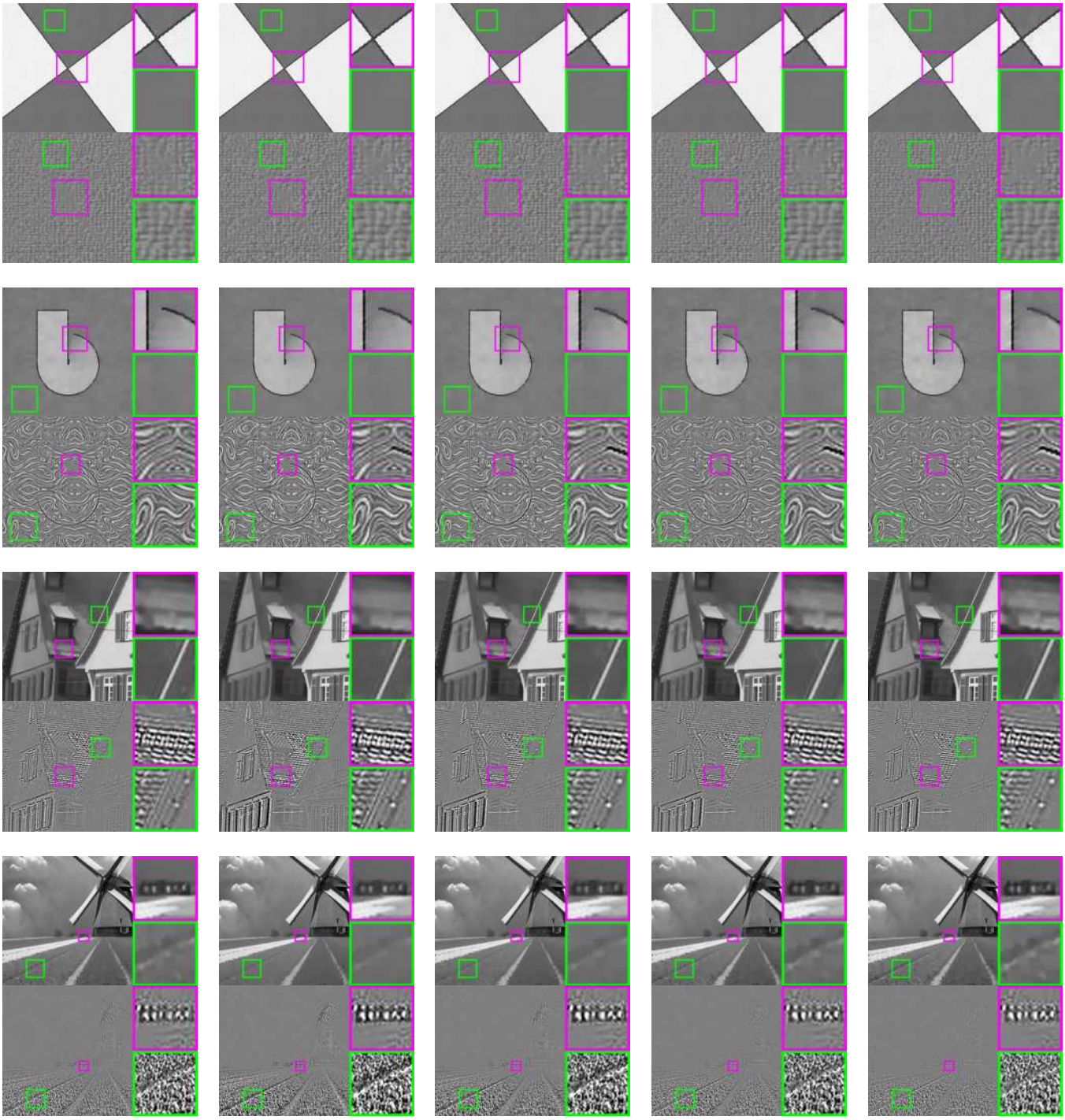



Fig. 22. Decomposition results of the proposed method on synthetic images and real images, with patch size varied from $3 \times 3$ to $11 \times 11$.

## X. Influence of parameters $\beta_1, \beta_2$

By varying $\beta_1, \beta_2$ within a moderate range, we tested the performance of the proposed method. It is observed in Fig. 23 that increasing $\beta_1$ will move the elements from cartoon layers to texture layers. For example, in the results of 'Snake', the eyes in the cartoon layer become more and more blurry as $\beta_1$ increases, and meanwhile they get more and more clear in the texture layer.

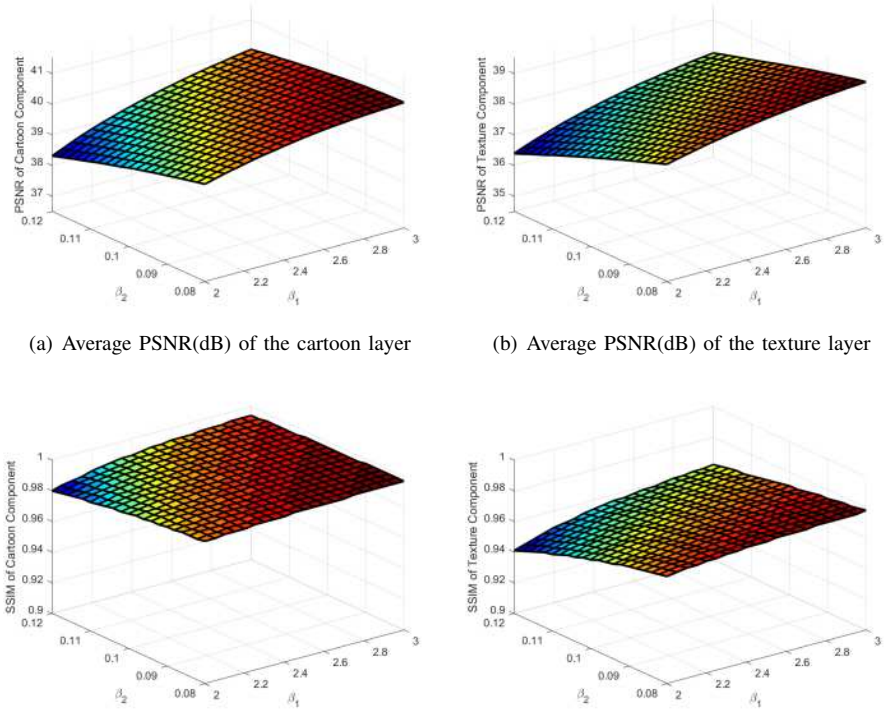

Fig. 23. Decomposition results of the proposed method on different images with $\beta_1$ varied from 1.25 to 3.75.

On the other hand, it can be seen from Fig. 24 that the elements are moved from the texture layers to the cartoon layers as $\beta_2$ is increased. For instance, in the cartoon layer of 'Barbara' the face has heavy blurring effect when $\beta_2$ is small, while the blur becomes less when $\beta_2$ becomes large. It is interesting to see in the results of 'House' that, the proposed method can always well preserve the strong edges in the cartoon layer when $\beta_1, \beta_2$ are varied within a moderate range.

$\beta_2$=0.05 $\beta_2$=0.07 $\beta_2$=0.10 $\beta_2$=0.13 $\beta_2$=0.15

Fig. 24. The decomposition results of the proposed method on different images with $\beta_2$ varied from 0.05 to 0.15.

To further quantitatively analyze the influences of these two parameters, we tested the decomposition on the synthetic images by setting $\beta_1, \beta_2$ to different values. The results are shown in Fig. 25, where both the PSNR and SSIM of the decomposition results are stable to the moderate changes of $\beta_1$ and $\beta_2$.



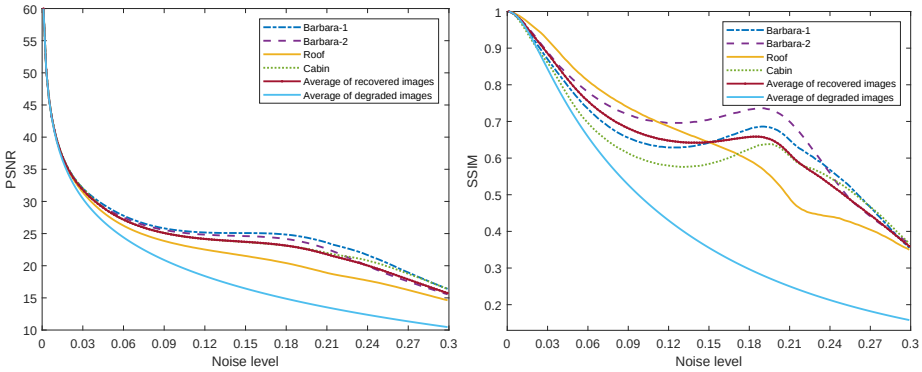

(a) Average PSNR(dB) of the cartoon layer (b) Average PSNR(dB) of the texture layer

(c) Average SSIM of the cartoon layer (d) Average SSIM of the texture layer

Fig. 25. PSNR and SSIM values of decomposition results of synthetic images by setting different values of $\beta_1, \beta_2$.

## XI. Decomposition on images with different noise levels

We further investigate the robustness of the proposed method to different levels of noise. In the experiment, the input images are degraded by zero-mean Gaussian noises, with standard deviation $\sigma$ varying from $0.001$ to $0.300$. To adapt the proposed method for different noise levels, the model parameters are set as $\beta_1 = 1200\sigma^2$ and $\beta_2 = 20\sigma^2$ empirically. The recovery results in terms of PSNR and SSIM are shown in Fig. 26, which shows that the proposed method is capable of recovering the images in varying noise levels. We also show the decomposition results in Fig. 27. It can be seen that the proposed method can produce reasonable decomposition results for noisy images with $\sigma$ in a moderate range, though obvious artifacts arise in the cartoon layers when $\sigma$ grows from $0.24$ to $0.28$.




(a) PSNR(db) of the recovered images (b) SSIM of the recovered images

Fig. 26. PSNR and SSIM values of recovery results for the noisy images with varying values of $\sigma$.

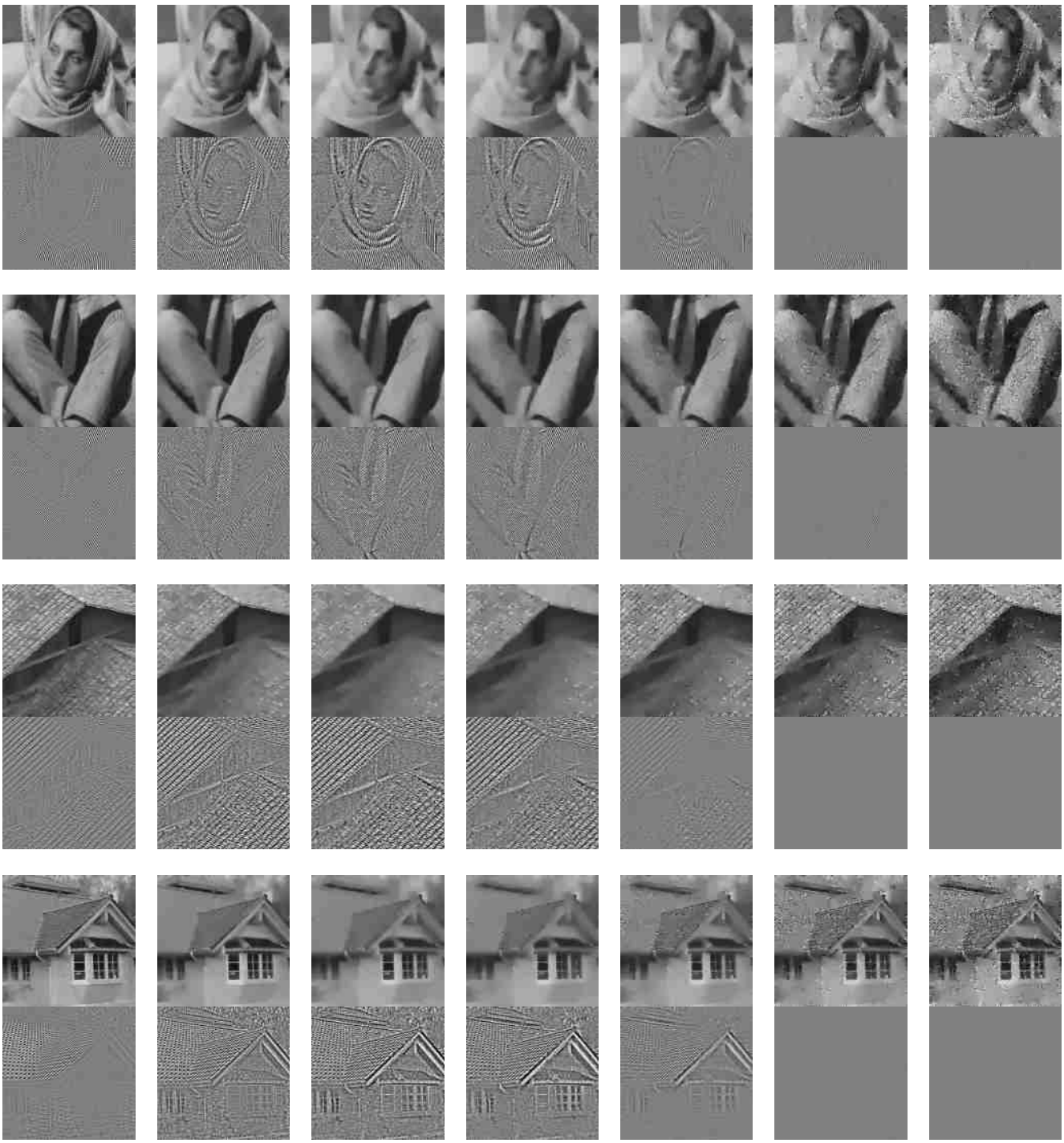



Fig. 27. The decomposition results of the proposed method on different images with $\sigma$ varying from $0.04$ to $0.28$. The odd rows show the cartoon layers and the even rows show the texture layer.

## XII. Decomposition on noisy image with different patch sizes in patch matching

We further analyze the influence of the patch size to the decomposition on noisy images. Fig.28 shows the decomposition results on noisy 'BarbaraHead' of the proposed method using different patch sizes in patch matching. It can be seen that as the patch size grows, more and more texture parts (*e.g.* the bars) are moved from the texture component to the cartoon component.


Fig. 28. The decomposition and denoised results of the proposed method on a noisy image with patch size varying from $5 \times 5$ to $23 \times 23$ in the patch matching stage. The rows from top to bottom corresponds to the cartoon layers, the texture layers, and the denoised images.

## References

[1] K. Valkealahti and E. Oja, "Reduced multidimensional co-occurrence histograms in texture classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 90–94, 1998.
[2] G. Kylberg, "The kylberg texture dataset v. 1.0," Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden, External report (Blue series) 35, September 2011. [Online]. Available: http://www.cb.uu.se/ gustaf/texture/
[3] M. Fritz, E. Hayman, B. Caputo, and J.-O. Eklundh, "The kth-tips database," 2004.
[4] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi, "Describing textures in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition*, 2014.