NTHU Rain Removal Project

Networked Video Lab, National Tsing Hua University, Hsinchu, Taiwan

People

Li-Wei Kang, Institute of Information Science, Academia Sinica, Taipei, Taiwan
Chia-Wen Lin*, Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan
Yu-Hsiang Fu, MStar Semiconductor Inc., Hsinchu, Taiwan

*Contact author: Prof. Chia-Wen Lin, email: cwlin@ee.nthu.edu.tw
This project was supported in part by the National Science Council, Taiwan, under Grants NSC98-2221-E-007-080-MY3, NSC100-2218-E-001-007-MY3, and NSC100-2811-E-001-005.

Documents

Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu, "Automatic single-frame-based rain streak removal via image decomposition," *IEEE Trans. Image Processing*. (submitted, March 2011, revised, August 2011)

Yu-Hsiang Fu, Li-Wei Kang, Chia-Wen Lin, and Chiou-Ting Hsu, "Single-frame-based rain removal via image decomposition," in *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing*, May 2011, Prague, Czech Republic.

Abstract

Rain removal from a video is a challenging problem and has been recently investigated extensively. Nevertheless, the problem of rain removal from a single image was rarely studied in the literature, where no temporal information among successive images can be exploited, making the problem very challenging. In this paper, we propose a single-image-based rain removal framework via properly formulating rain removal as an image decomposition problem based on morphological component analysis (MCA). Instead of directly applying conventional image decomposition technique, we first decompose an image into the low-frequency and high-frequency parts using a bilateral filter. The high-frequency part is then decomposed into "rain component" and "non-rain component" by performing dictionary learning and sparse coding. As a result, the rain component can be successfully removed from the image while preserving most original image details. Experimental results demonstrate the efficacy of the proposed algorithm.

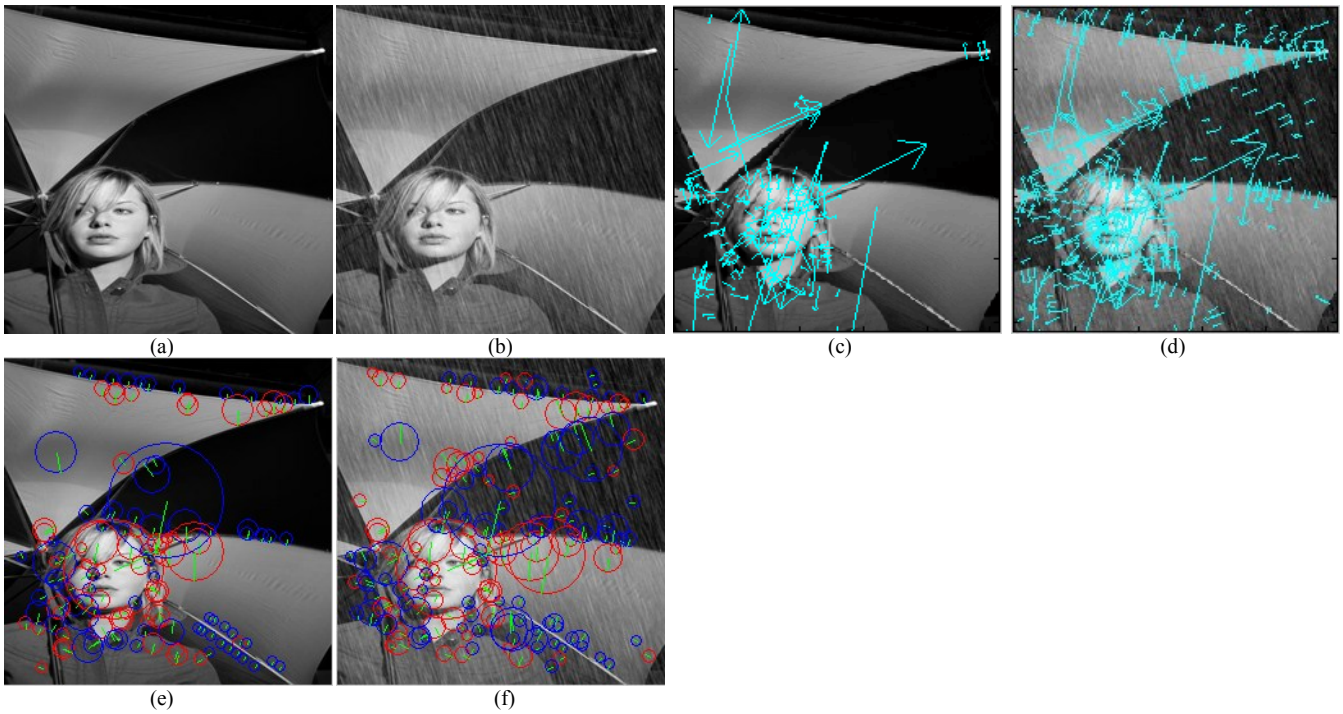
Introduction

In this project, we propose a single-image-based rain streak removal framework [1]-[2] by formulating rain streak removal as an image decomposition problem based on MCA [3]. In our method, an image is first decomposed into the low-frequency and high-frequency parts using a bilateral filter. The high-frequency part is then decomposed into "rain component" and "non-rain component" by performing dictionary learning and sparse coding based on MCA. The major contribution of this paper is three-fold: (i) to the best of our knowledge, our method is among the first to achieve rain streak removal while preserving geometrical details in a single frame, where no temporal or motion information among successive images is required; (ii) we propose the first automatic MCA-based image decomposition framework for rain steak removal; and (iii) the learning of the dictionary for decomposing rain steaks from an image is fully automatic and self-contained, where no extra training samples are required in the dictionary learning stage.

Motivation

So far, the research works on rain streak removal found in the literature have been mainly focused on video-based approaches that exploit temporal correlation in multiple successive frames [4]-[6]. Nevertheless, when only a single image is available, such as an image captured from a digital camera/camera-phone or downloaded from the Internet, a single-image based rain streak removal approach is required, which was rarely investigated before. Moreover, many image-based applications such as mobile visual search, object detection/recognition, image registration, image stitching, and salient region detection heavily rely on extraction of gradient-based features that are rotation- and scale-invariant. Some widely-used features (descriptors) such as SIFT (scale-invariant feature transform) [7], SURF (speeded up robust features) [8], and HOG (histogram of oriented gradients) [9] are mainly based on computation of image gradients. The

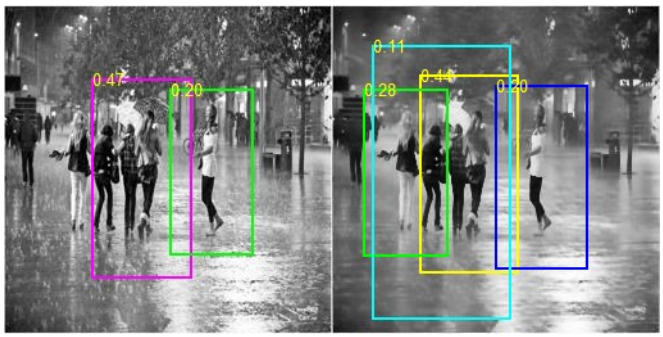
performances of these gradient-based feature extraction schemes, however, can be significantly degraded by rain streaks appearing in an image since the rain streaks introduce additional time-varying gradients in similar directions, as examples shown in Figs. 1-2. In addition, visual attention models [10] compute a saliency map topographically encoding for saliency at each location in the visual input that simulates which elements of a visual scene are likely to attract the attention of human observers. Nevertheless, the performances of the model for related applications may also be degraded if rain streaks directly interact with the interested target in an image, as an example shown in Fig. 3.



(a) (b) (c) (d)

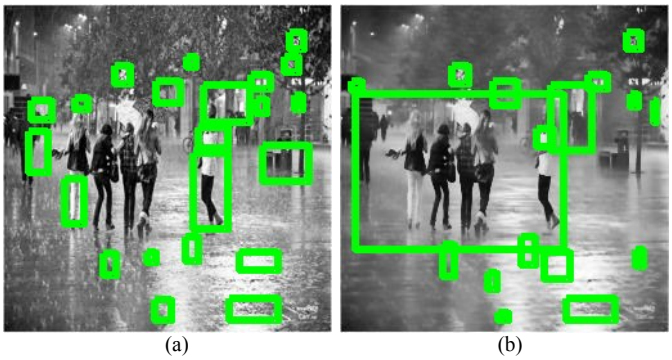
(e) (f)

Fig. 1. Examples of interesting point detection: **(a)** the original non-rain image; **(b)** the rain image of (a); **(c)** SIFT interesting point detection for (a) (169 points); **(d)** SIFT interesting point detection for (b) (421 points); **(e)** SURF interesting point detection for (a) (131 points); and **(f)** SURF interesting point detection for (b) (173 points).



(a) (b)

Fig. 2. Applying the HOG-based pedestrian detector released from [14] to: **(a)** the original rain image (4 pedestrians detected); and **(b)** the rain-removed version (obtained by the proposed method) of (a) (5 pedestrians detected).



(a) (b)

Fig. 3. Applying the visual attention model-based blob detector released from [15] to: **(a)** the original rain image (only 2 people partially marked in the 5 more remarkable people); and **(b)** the rain-removed version (obtained by the proposed method) of (a) (the 5 more remarkable people marked).

Proposed Automatic Single-Image-Based Rain Streaks Removal Framework

Fig. 4 shows the proposed single-image-based rain streak removal framework, in which rain streak removal is formulated as an image decomposition problem. In our method, the input rain image is first roughly decompose into the low-frequency (LF) part and the high-frequency (HF) part using the bilateral filter [11], where the most basic information will be retained in the LF part while the rain streaks and the other edge/texture information will be included in the HF part of the image as illustrated in Figs. 5(a) and 5(b). Then, we perform the proposed MCA-based image decomposition to the HF part that can be further decomposed into the rain component [see Fig. 5(c)] and the geometric (non-rain) component [see Fig. 5(d)]. In the image decomposition step, a dictionary learned from the training exemplars extracted from the HF part of the image itself can be divided into two sub-dictionaries by performing HOG [9] feature-based dictionary atom clustering. Then, we perform sparse coding [12] based on the two sub-dictionaries to achieve MCA-based image decomposition, where the geometric component in the HF part can be obtained, followed by integrating with the LF part of the image to obtain the rain-removed version of this image as illustrated in Figs. 5(e) and 5(f).

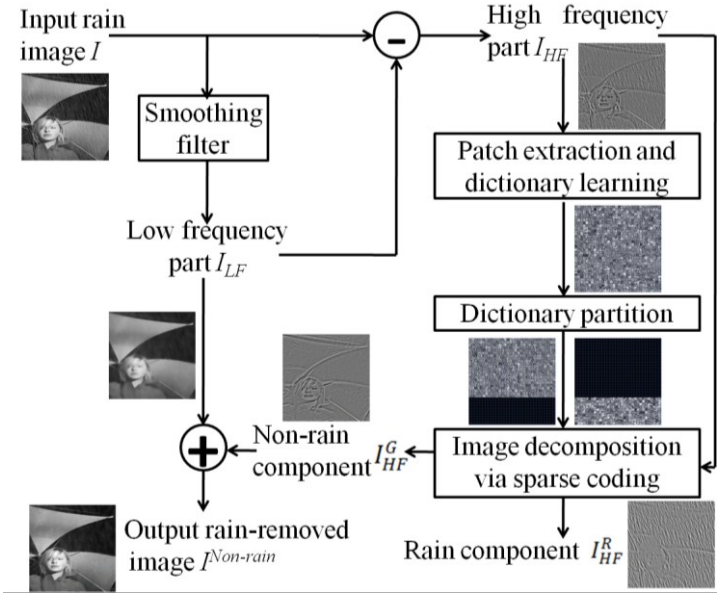


Fig. 4. Block diagram of the proposed rain streak removal method.

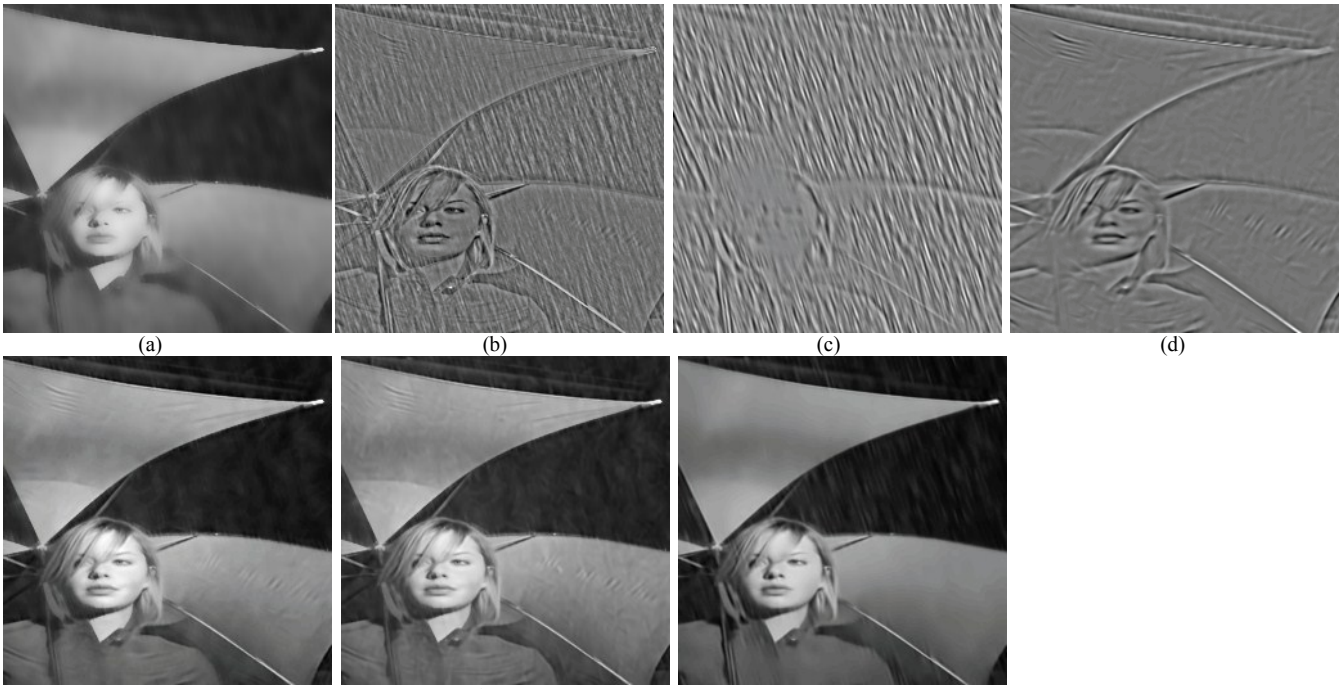


Fig. 5. Step-by-step results of the proposed rain streak removal process: **(a)** the low-frequency (LF) part of the rain image in Fig. 1(b) decomposed using the bilateral filter [11] (VIF = 0.33); **(b)** the high-frequency (HF) part; **(c)** the rain component; and **(d)** geometric component. Combining (d) and the LF part shown in (a) to obtain: **(e)** the rain-removed version for the rain image shown in Fig. 1(b) (VIF = 0.50); **(f)** the rain-removed version for the rain image shown in Fig. 1(b) with extended dictionary (VIF = 0.52); and **(g)** the rain-removed version for the rain image shown in Fig. 1(b) via the K-SVD-based denoising [16] (VIF = 0.46).

Algorithm: **Single-Image-Based Rain Streaks Removal**

Input: a rain image *I*.

Output: the rain-removed version I^{Non_Rain} of I.

1. Apply the bilateral filter to obtain the LF part I_{LF} and HF part I_{HF}, such that $I = I_{LF} + I_{HF}$.
2. Extract a set of image patches $y^k \in R^n, k = 1,2, \dots, P$, from I_{HF}. Apply the online dictionary learning for sparse coding algorithm to solve

$$\min_{D_{HF} \in R^{n \times m}, \theta^k \in R^m} \frac{1}{P} \sum_{k=1}^{P} \left(\frac{1}{2} \left\| y^k - D_{HF} \theta^k \right\|_2^2 + \lambda \left\| \theta^k \right\|_1 \right)$$

   to obtain the dictionary D_{HF} consisting of the atoms that can sparsely represent $y^k, k = 1,2, \dots, P$.
3. Extract HOG feature descriptor for each atom in D_{HF}. Apply *K*-means algorithm to classify all of the atoms into two clusters based on their HOG feature descriptors.
4. Identify one of the two clusters as "rain sub-dictionary," D_{HF_R} and the other one as "geometric sub-dictionary," D_{HF_G}.
5. Apply MCA by performing OMP to solve

$$\min_{\theta_{HF}^k \in R^m} \left\| b_{HF}^k - D_{HF} \theta_{HF}^k \right\|_2^2 \; s.t. \left\| \theta_{HF}^k \right\|_0 \leq L$$

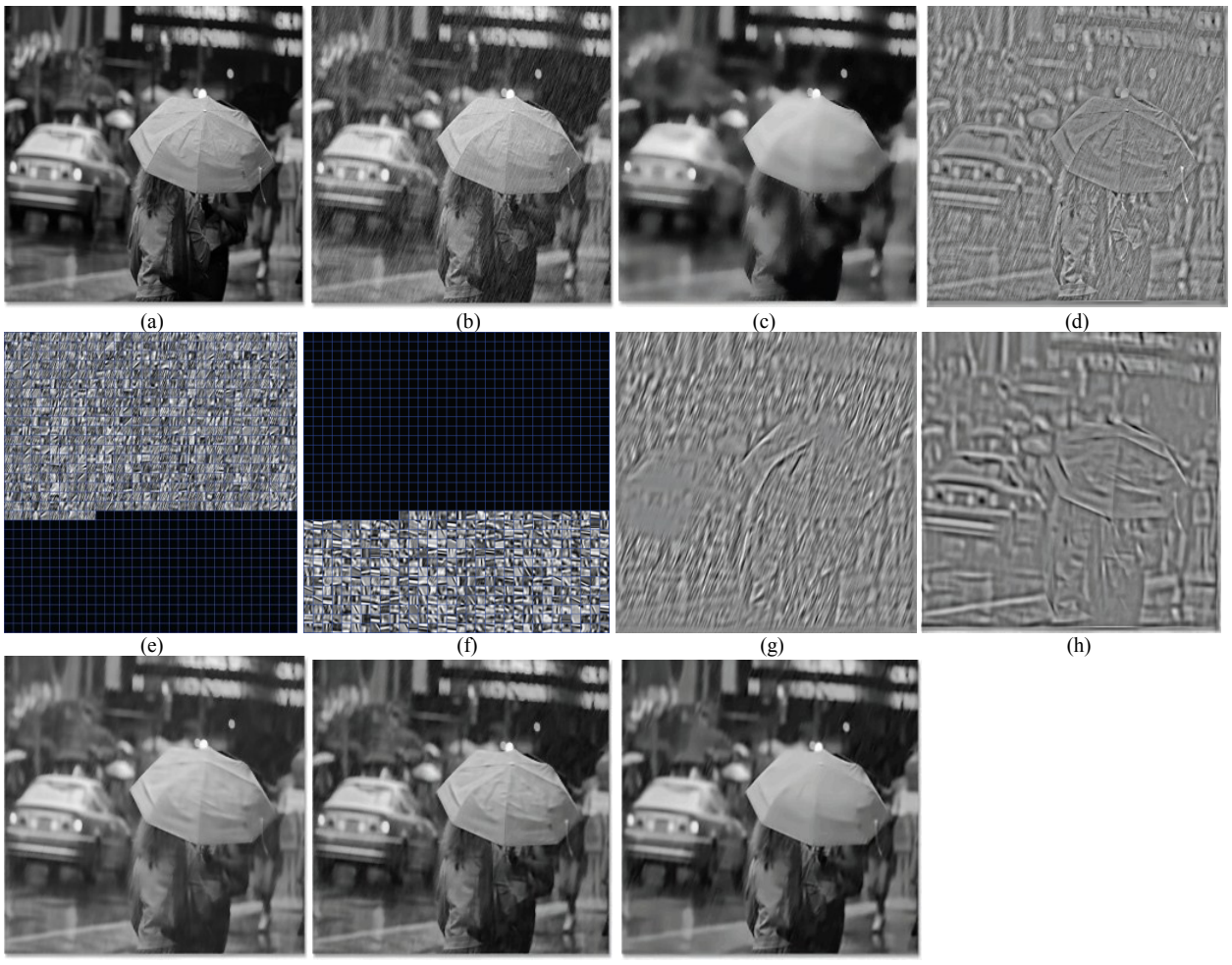
   for each patch $b_{HF}^k \in \mathrm{R}^n$, k= 1, 2, …, *P*, in I_{HF} with respect to $D_{HF} = \left[D_{HF\_G} | D_{HF\_R} \right]$.
6. Reconstruct each patch b_{HF}^k to recover either geometric component I_{HF}^G or rain component I_{HF}^R of I_{HF} based on the corresponding sparse coefficients obtained from Step 5.
7. Return the rain-removed version of I via $I^{Non\_Rain} = I_{LF} + I_{HF}^G$.

Experiments

Because we cannot find any other single-image-based approach, to evaluate the performance of the proposed algorithm, we first compare the proposed method with a low-pass filtering method called the bilateral filter proposed in [11], which has been extensively applied and investigated recently for image processing, such as image denoising. Besides, to demonstrate that existing image denoising methods cannot well address the problem of single-image-based rain

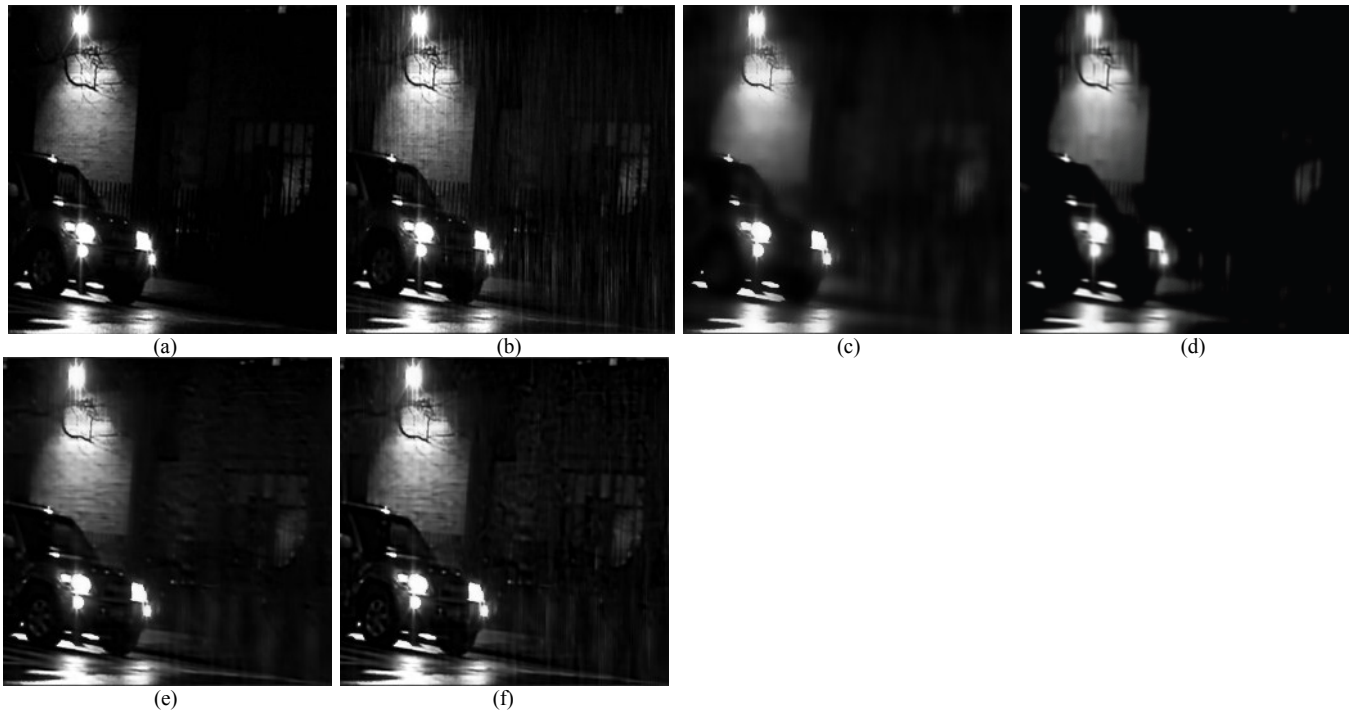
removal, we also compare the proposed method with the state-of-the-art image denoising method based on K-SVD dictionary learning and sparse representation proposed in [16] based on the assumption that the standard deviation of noise, which is assumed to be Gaussian distributed, can be known in advance, with a released source code available from [17] (denoted by "K-SVD-based denoising"). In rain removal applications, the standard deviation value of the rain noise is usually unknown. To estimate the standard deviation of the rain noise, for a rendered rain image with a ground-truth, we direct calculate the deviation of each rain component patch as an initial value, whereas for a natural rain image, we use the rain component obtained from the proposed method to estimate the initial value. We then manually tune the value around the initial value to ensure that most of the rain streaks in the rain image can be removed.

We collected several natural or synthesized rain images from the Internet with ground-truth images (non-rain versions) for a few of them. To evaluate the quality of a rain-removed image with a ground-truth, we used the visual information fidelity (VIF) metric [13] which has been shown to outperform PSNR (peak signal-to-noise ratio) metric. Besides, we also compare our method with a video-based rain removal method based on adjusting camera parameters proposed in [6] (denoted by "video-based camera see"), which should outperform most of other video-based techniques without adjusting cameras. We captured some single frames from the videos released from [6] and compared our results with the ones of [6] from the same videos. For each video released from [6], the preceding frames are rain frames, followed by succeeding rain-removed frames in the same scene. We pick a single rain frame from the preceding frames for rain removal and compared our results with the rain-removed one [6] of a similar frame from the succeeding frames in the same video (no exactly the same frame is available for comparison). The rain removal results are shown in Figs. 5-25. The simulation results demonstrate that although the bilateral filter can remove most rain streaks, it simultaneously removes much image detail as well. The proposed method successfully removes most rain streaks while preserving most non-rain image details in most cases. Moreover, the simulation results also demonstrate that the performance of the proposed methods is comparable with the "video-based camera see" method when rain streaks are obviously visible in a single frame. Moreover, we also demonstrate the significance of the proposed method with extended dictionary by calculating the VIF values for selected image regions, as shown in Fig. 26.



(i) (j) (k)

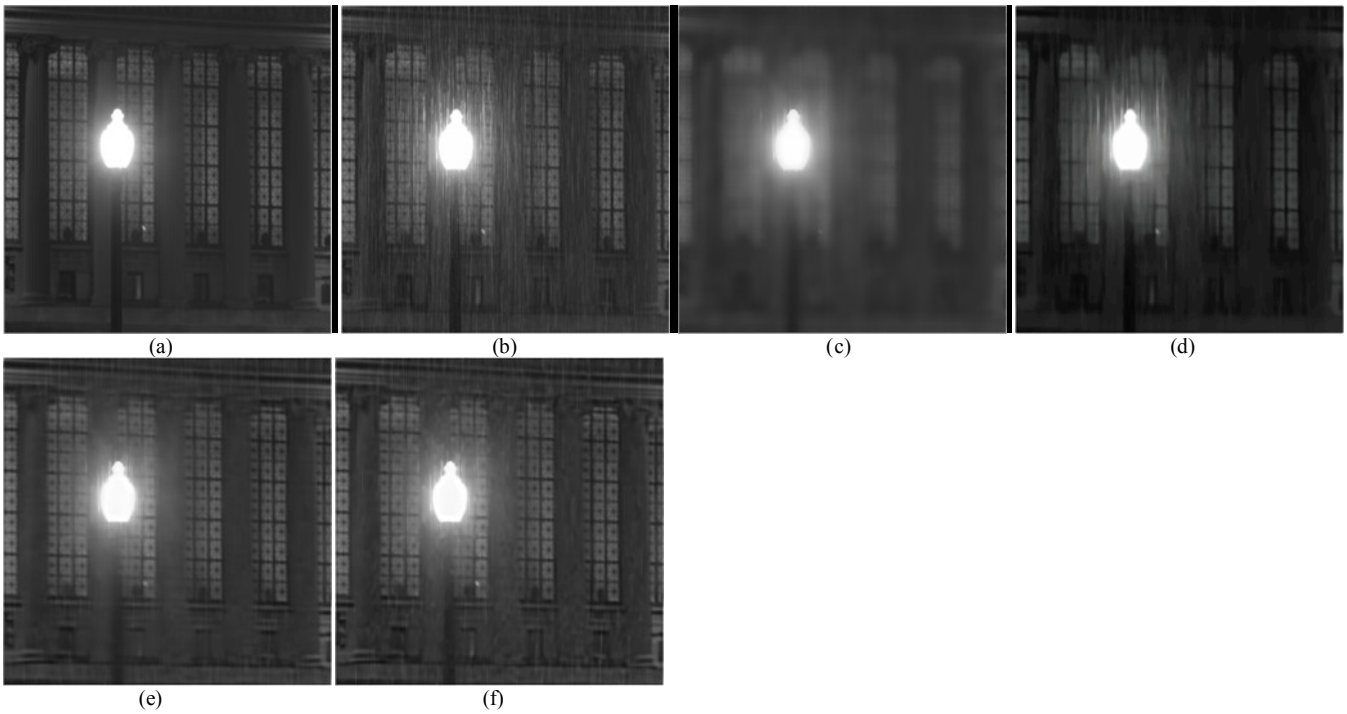
Fig. 6. Rain removal results: **(a)** the original non-rain image (ground-truth); **(b)** the rain image of (a); **(c)** the rain-removed version of (b) via the bilateral filter [11] (VIF = 0.31); **(d)** the HF part of (b); **(e)** the rain sub-dictionary for (d); **(f)** the geometric sub-dictionary for (d); **(g)** the rain component of (d); **(h)** the geometric component of (d); **(i)** the rain-removed version of (b) via the proposed method (VIF = 0.53); **(j)** the rain-removed version of (b) via the proposed method with extended dictionary (VIF = 0.57); and **(k)** the rain-removed version of (b) via the K-SVD-based denoising [16] (VIF = 0.51).



(a) (b) (c) (d)

(e) (f)

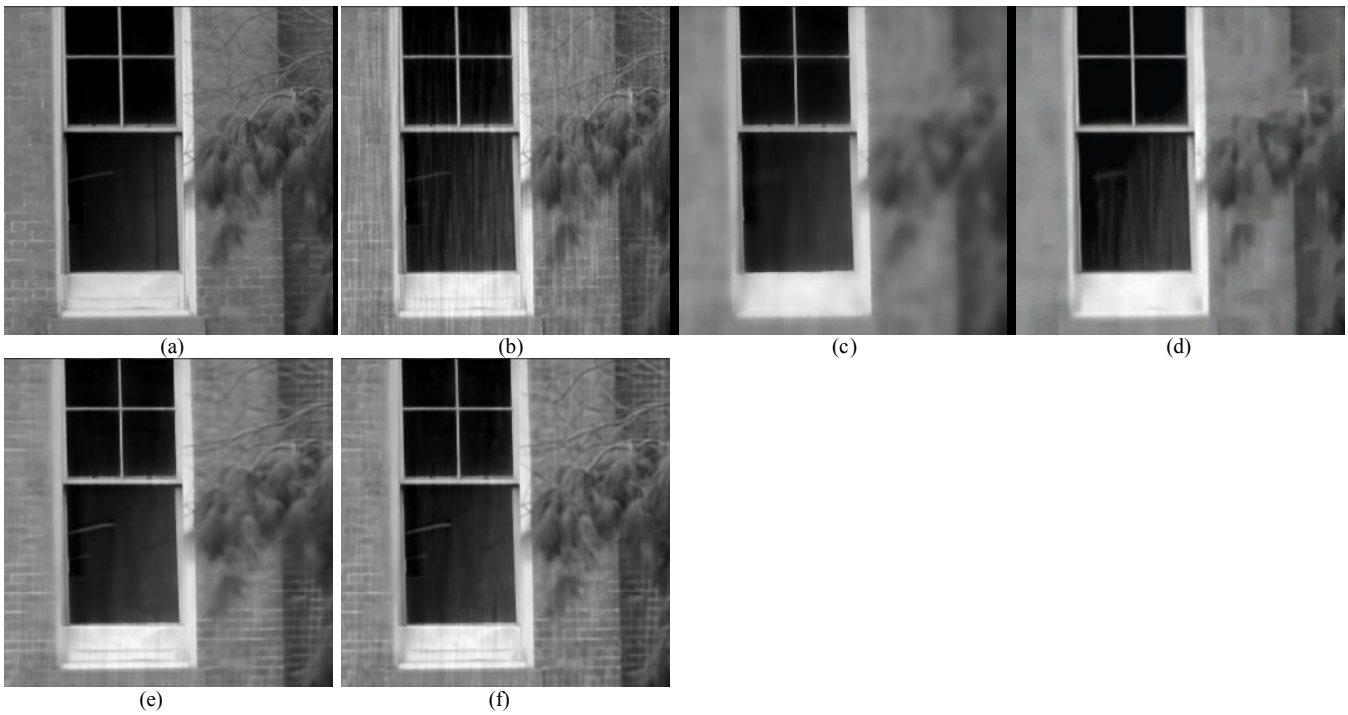
Fig. 7. Rain removal results: **(a)** the original non-rain image; **(b)** the rain image of (a); the rain-removed versions of (b) via the: **(c)** bilateral filter [11] (VIF = 0.21); **(d)** K-SVD-based denoising [16] (VIF = 0.21); **(e)** proposed method (VIF = 0.36); and **(f)** proposed method with extended dictionary (VIF = 0.38).



(a) (b) (c) (d)

(e) (f)

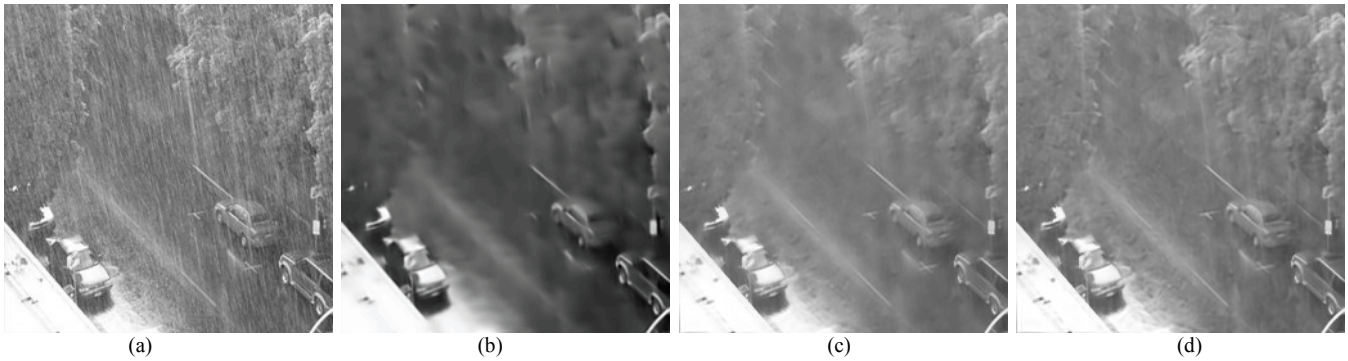
Fig. 8. Rain removal results: **(a)** the original non-rain image; **(b)** the rain image of (a); the rain-removed versions of (b) via the: **(c)** bilateral filter [11] (VIF = 0.09); **(d)** K-SVD-based denoising [16] (VIF = 0.16); **(e)** proposed method (VIF = 0.20); and **(f)** proposed method with extended dictionary (VIF = 0.20).



(a) (b) (c) (d)

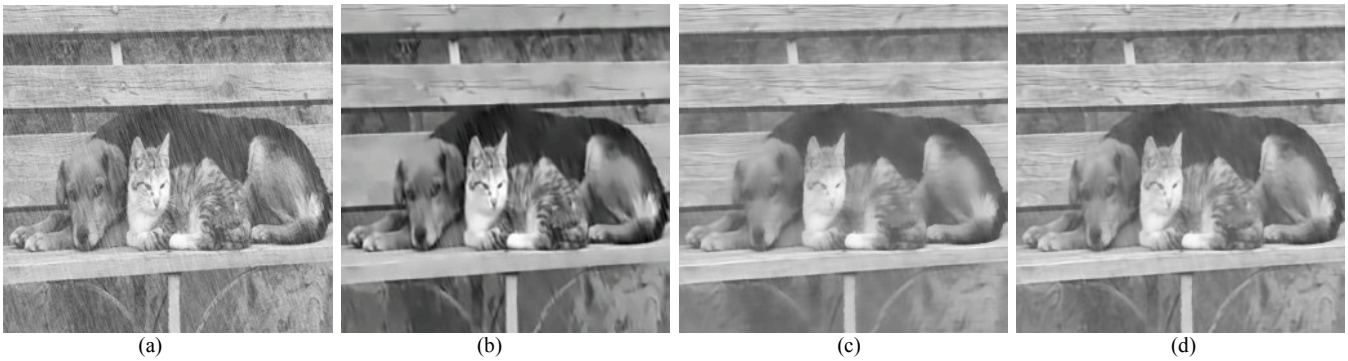
(e) (f)

Fig. 9. Rain removal results: **(a)** the original non-rain image; **(b)** the rain image of (a); the rain-removed versions of (b) via the: **(c)** bilateral filter [11] (VIF = 0.29); **(d)** K-SVD-based denoising [16] (VIF = 0.38); **(e)** proposed method (VIF = 0.56); and **(f)** proposed method with extended dictionary (VIF = 0.60).



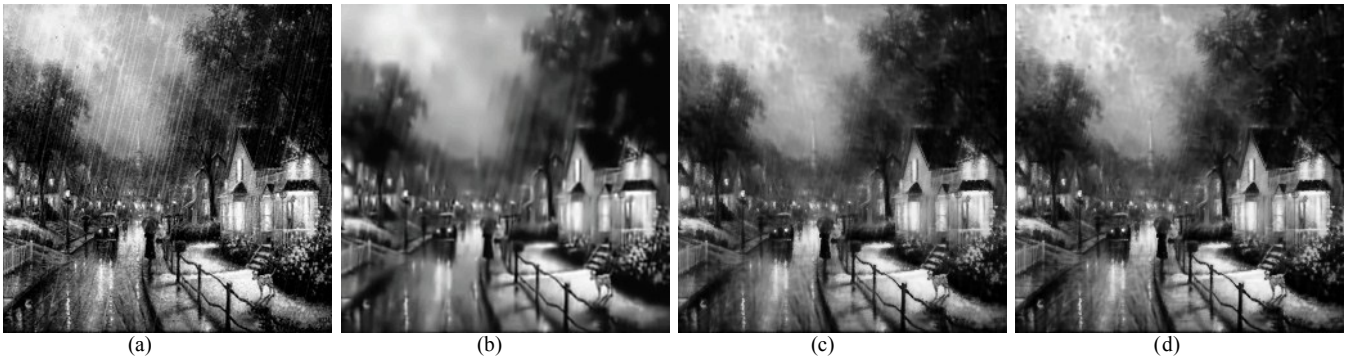
(a) (b) (c) (d)

Fig. 10. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.



(a) (b) (c) (d)

Fig. 11. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.

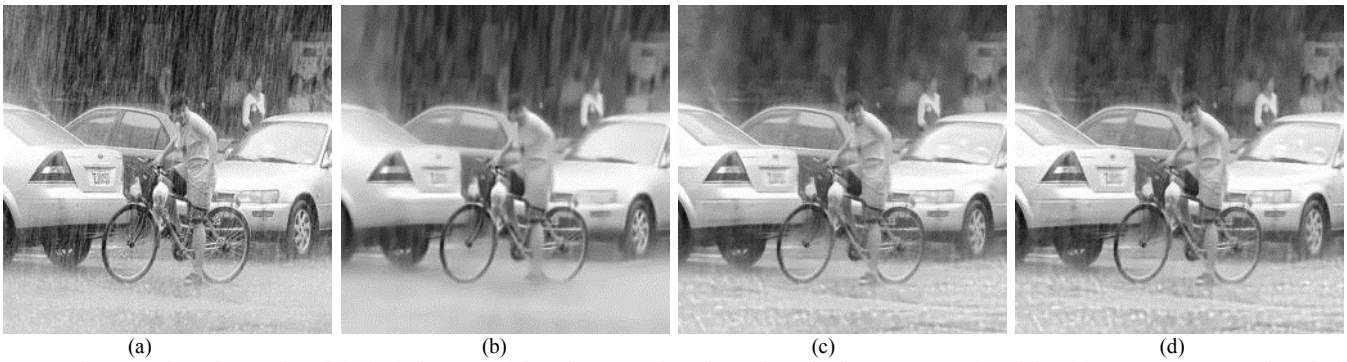


(a) (b) (c) (d)

Fig. 12. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.

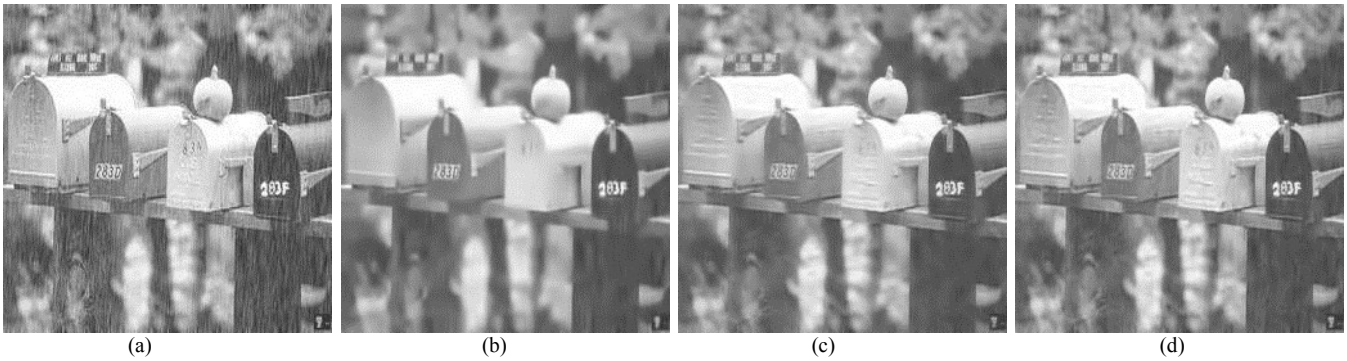
(a) (b) (c) (d)

Fig. 13. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.



(a) (b) (c) (d)

Fig. 14. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.



(a) (b) (c) (d)

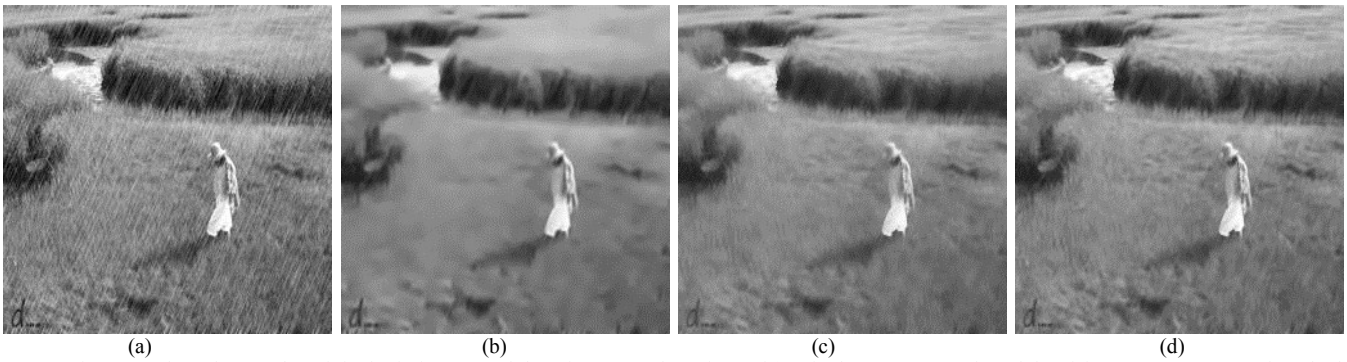
Fig. 15. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.

(a) (b) (c) (d)

Fig. 16. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.

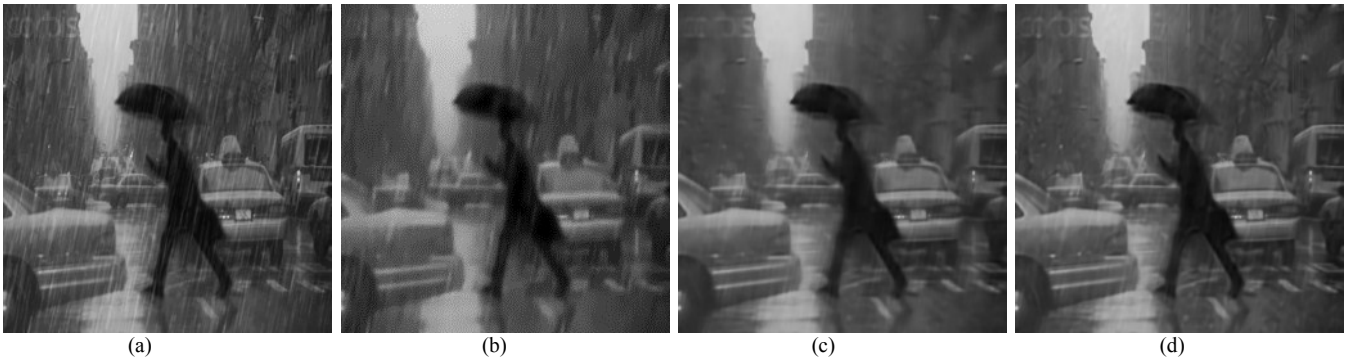
(a) (b) (c) (d)

Fig. 17. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.



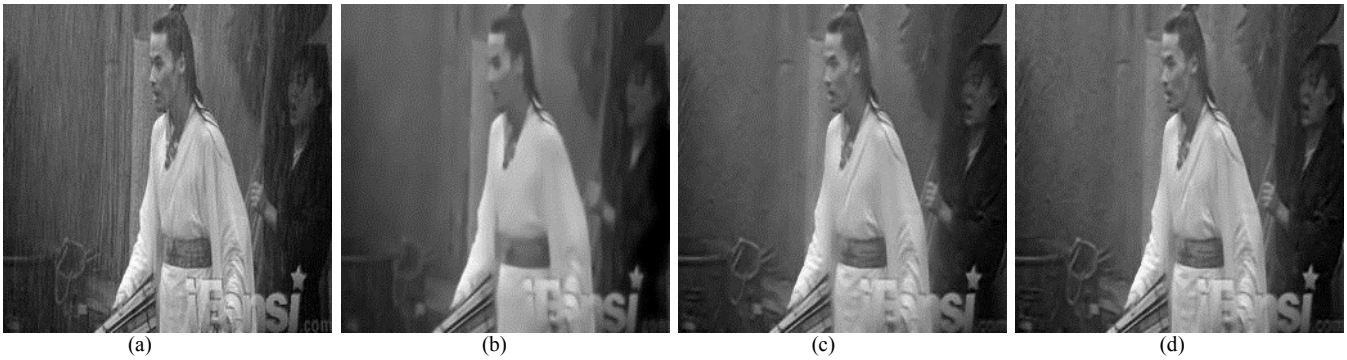
(a) (b) (c) (d)

Fig. 18. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.



(a) (b) (c) (d)

Fig. 19. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.

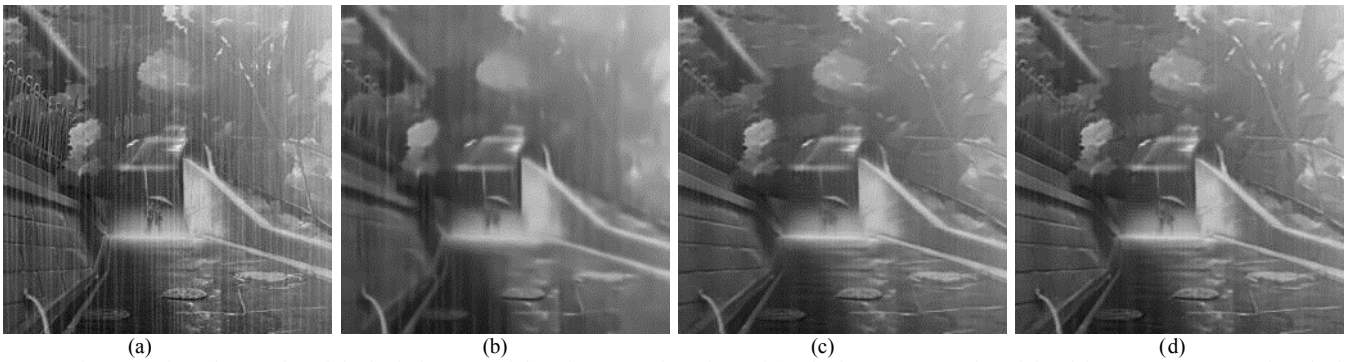


(a) (b) (c) (d)

Fig. 20. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.

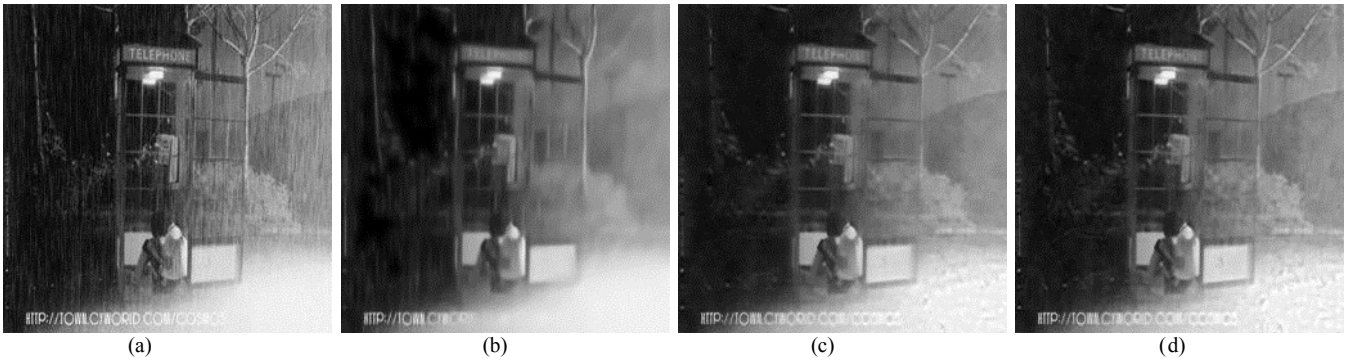
(a) (b) (c) (d)

Fig. 21. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.



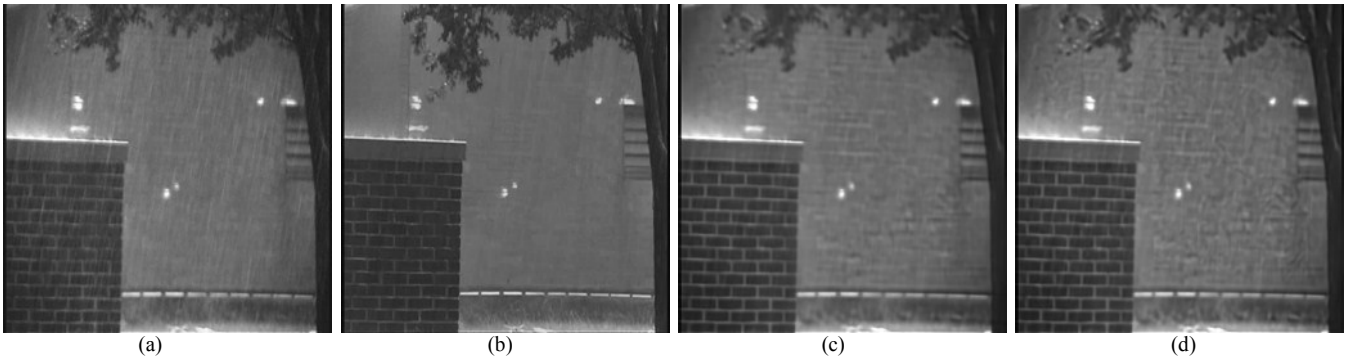
(a) (b) (c) (d)

Fig. 22. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.



(a) (b) (c) (d)

Fig. 23. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** K-SVD-based denoising [16]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.

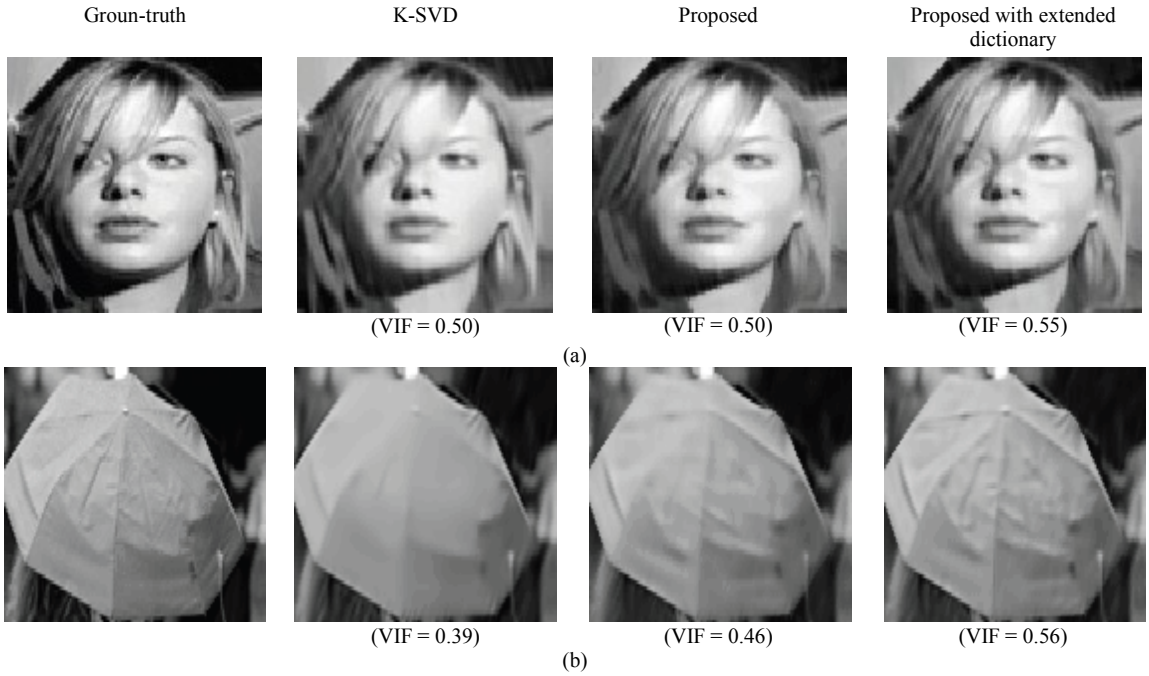


(a) (b) (c) (d)

Fig. 24. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** video-based camera see method [6]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.

(a) (b) (c) (d)

Fig. 25. Rain removal results: **(a)** the original rain image; and the rain-removed versions of (a) via the: **(b)** video-based camera see method [6]; **(c)** proposed method; and **(d)** proposed method with extended dictionary.



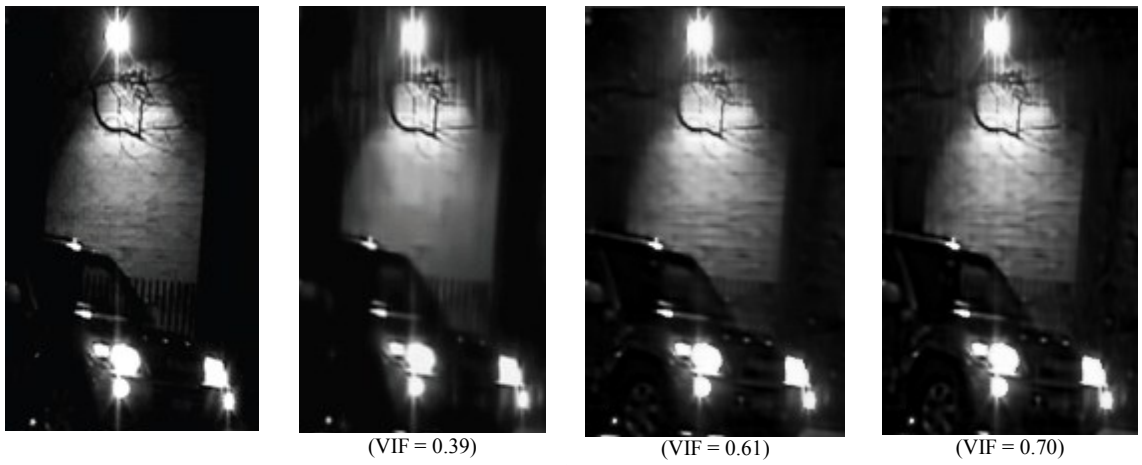


Fig. 26. Rain removal results of selected image regions. From left to right are the original non-rain image; and the rain-removed versions via the: K-SVD-based denoising method [16]; proposed method; and proposed method with extended dictionary.

Conclusions

In this paper, we have proposed a single-image-based rain streak removal framework by formulating rain removal as an MCA-based image decomposition problem solved by performing sparse coding and dictionary learning algorithms. Our experimental results show that the proposed method can effectively remove rain steaks without significantly blurring the original image details. For future work, the performance may be further improved by enhancing the sparse coding, dictionary learning, and partition of dictionary steps. For example, when performing sparse coding, some locality constraint may be imposed to guarantee that similar patches should have similar sparse codes/coefficients [18]. Moreover, the proposed method may be extended to remove rain streaks from videos or other kinds of repeated textures.

References

[1] Y. H. Fu, L. W. Kang, C. W. Lin, and C. T. Hsu, "Single-frame-based rain removal via image decomposition," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, May 2011, Prague, Czech Republic.

[2] L. W. Kang, C. W. Lin, and Y. H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Trans. Image Process.* (revised, Aug. 2011).

[3] J. M. Fadili, J. L. Starck, J. Bobin, and Y. Moudden, "Image decomposition and separation using sparse representations: an overview," *Proc. IEEE*, vol. 98, no. 6, pp. 983–994, June 2010.

[4] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *Int. J. Comput. Vis.*, vol. 86, no. 2–3, pp. 256–274, 2010.

[5] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2004, vol. 1, pp. 528–535.

[6] K. Garg and S. K. Nayar, "When does a camera see rain?" in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2005, vol. 2, pp. 1067-1074.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[8] H. Baya, A. Essa, T. Tuytelaarsb, and L. V. Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understanding*, vol. 110, no. 3, pp. 346–359, June 2008.

[9] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, June 2005, vol. 1, pp. 886–893.
[10] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov 1998.
[11] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Bombay, India, Jan. 1998, pp. 839–846.
[12] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online learning for matrix factorization and sparse coding,” *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.
[13] H. R. Sheikh and A. C. Bovik, “Image information and visual quality,” *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.
[14] S. Maji, A. C. Berg, and J. Malik, “Classification using intersection kernel support vector machines is efficient,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, Alaska, USA, June 2008, pp. 1-8.
[15] M. Jahangiri and M. Petrou, “An attention model for extracting components that merit identification,” in *Proc. IEEE Int. Conf. Image Processing*, Cairo, Egypt, Nov. 2009, pp. 965-968.
[16] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.
[17] M. Elad, *Sparse and Redundant Representations: from Theory to Applications in Signal and Image Pprocessing*, Springer, 2010.
[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained linear coding for image classification,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, June 2010, pp. 3360–3367.