Rain Streaks Detection and Removal from Color-Image Video Using Sparse Representation

P. Vengatachalam and Vijayakumar

Abstract--- Rain streaks detection and removal from color image -video is a challenging problem. The rain streak removal is considered as image denoising task. In color image –video based rain streaks removal, where the dictionary learning process can be only applied once for the first frame in a video clip of the same scene. The dictionary learning can be also used for removal of rain streaks for the succeeding frames in the clip, which is useful to both reduce the computational complexity and maintain the temporal consistency of the video. The proposed Color image - video based rain streaks removal framework based on the sparse representation. Then the high frequency part was decomposed into rain and non-rain component by using the learning sparse representation based dictionaries. To separate a rain streaks from high frequency part using the muti set feature. The multi set feature, including HOG (Histogram Oriented Gradients), DOF (Depth of Field), and eigen color. The high frequency part and muti set features is applied to remove the most rain streaks. The DOF feature is used to help to identify the main subjects to preserve in a rain image. The rain streaks are usually neutral color, where the eigen color feature is used to analyse the key features of the rain streaks. The both DOF and Eigen color features are used to identify and separate the non rain component from the misidentified rain component of an image. Our proposed framework may be also integrated with any sparse representation–based super-resolution framework to achieve super-resolution of a low quality video and noisy image video. Rain removal is a very useful and important technique in applications such as security surveillance, audio/video editing and investigations.

P. Vengatachalam, Asst Prof, School of Computing Sciences, Vels University, Chennai, India
Vijayakumar, MCA(M.phil), Research Scholar, School of Computing Sciences, Vels University, Chennai, India

I. INTRODUCTION

ASSUMING an image is a linear mixture of multiple source components, image decomposition aims at determining such components and the associated weights [1], [2].For example, how to properly divide an image into texture and Manuscript received December 13, 2012; revised April 18, 2013 and June 17, 2013; accepted July 08, 2013. Date of publication October 07, 2013; date of current version December 12, 2013. This paper is an extended version of the original paper which appeared in the Proceedings of IEEE ICME 2012 Conference and was among the top-rated 4% of ICME'12 submissions. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sen-ChingCheung.D.-A. Huang and Y.-C. F. Wang are with the Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan (email:andrew800619@gmail.com;ycwang@citi.sinica.edu.tw).L.-W. Kang is with the Graduate School of Engineering Science and Technology-Doctoral Program, and the Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin, Taiwan (e-mail: lwkang@yuntech.edu.tw). C.-W. Lin is with the Department of Electrical Engineering and the Institute of Communications Engineering, National Tsing Hua University, Hsinchu, Taiwan (e-mail: cwlin@ee.nthu.edu.tw). Color versions of one or more of the figures in this paper are available online at Digital Object Identifier 10.1109/TMM.2013.2284759non-texture parts has been investigated in applications of image compression [3], image in painting [4], [5], or related image

analysis and synthesis tasks. Consider a fundamental problem of decomposing an image of *N* pixels into *C* different *N*-dimensional components, one needs to solve a linear regression problem with unknown variables. While this problem is ill-posed, image sparsity prior has been exploited to address this task [1]. As a result, an input image can be morphologically decomposed into different patches based on such priors for a variety of image processing applications. Before providing the overview and highlighting the contributions of our proposed method, we will first briefly review morphological component analysis (MCA), which is a sparsere presentation based image decomposition algorithm, and has been successfully applied and extended to solve the problems of image denoising [6]–[8], image in painting [5], [8], and image deraining (i.e., rain removal) [9], [10].

A. MCA for Image Decomposition

MCA utilizes the morphological diversity of different features contained in the data to be decomposed and to associate each morphological component to a dictionary of atoms [1],[5], [11]. Suppose an image *I* of *N* pixels is a superposition of *K* components (called morphological components), denoted by where denotes the *k*-th component, such asthe geometric or textural component of the image *I*. To decompose*I* into ,MCA iteratively minimizes the following energy function: where denotes the sparse coefficients corresponding to with respect to the dictionary , is a regularization parameter, and is the energy function defined according to the type of (global or local dictionary).The MCA algorithms solve (1) by iteratively performing for each component , the following two steps: (i) update of the sparse coefficients: this step performs sparse coding to solve or , where represents the sparse coefficients of the *p*-th patch extracted from , and *P* is the total number ofextracted patches, to minimize while fixing and.

(ii) update of the components: this step updates or while fixing or . More details about MCA can be found in [1], [5], [11].

Symbols	Meanings
I, I_L, I_H	Input image to be decomposed, low-frequency part of I, and high-frequency part of I
M, K	Number of atoms, number of image components/clusters
τ, λ	Regularization parameters
$\boldsymbol{y}_H^p$	The p-th training exemplar patch of size $n \times n$ extracted from I_H
$\boldsymbol{D}_H$	Dictionary of size $n^2 \times M$ for sparsely representing I_H
$\boldsymbol{d}_i$	The i-th atom in $\boldsymbol{D}_H$
$\boldsymbol{d}_i^k$	The i-th atom of the k-th group dictionary.
$\boldsymbol{\theta}_H^p$	Sparse coefficient vector of $\boldsymbol{y}_H^p$ with respect to $\boldsymbol{D}_H$
$\boldsymbol{D}_H^k$	Dictionary for sparsely representing I_H^k
I_H^k	The k-th component of I_H

II. EXPERIMENTS

To evaluate the performance of our proposed method, we conduct experiments for addressing two single-image denoising tasks: rain removal and denoising (with Gaussian noise). We consider the patch size of each image as 16 16 pixels, and the number of dictionary atoms . As suggested in [17],the regularization parameter and the maximum sparsity value for the OMP algorithm are set as 0.15 and 10, respectively. For LPF preprocessing techniques, we have the spatial and intensity- domain standard deviations for bilateral filtering as 6 and0.2, respectively. All images are of size 256 256 pixels in our experiments.

A. Performance Evaluation On Single Image Rain Removal

We collect several synthetic rain images from the Internet or the photo-realistically rendered rain video frames provided in[21], and thus we have ground-truth images without rain streaks presented for PSNR calculation. To evaluate the performance of our proposed method for rain removal, we compare our method with bilateral filtering (denoted by “Bilateral”) [12],K-SVD [7],and BM3D [13] denoising algorithms. We set large standard deviation values and 35 for K-SVD and BM3D algorithms, respectively. We note that, during the preprocessing stage of our framework, larger values allow us to remove high spatial frequency patterns including possible rain streaks from the low spatial frequency parts of the input image. We do not (and it is not possible) fine tune such parameters for removing the rain streaks only. In addition to the above methods, we consider two of our prior rain removal works: MCA-based rain removal (denoted

Self-Learning based Image Decomposition with Applications to Single Image Denoising

Fig. 7.Example rain removal results. Note that the input image is the noisy version of ground truth image with rain streaks presented. Rain removal outputs are

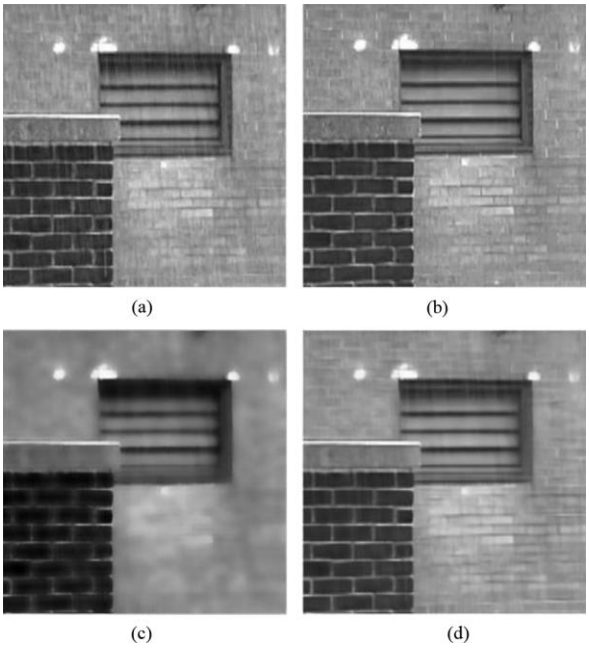
produced by the methods of (a) Context-based [10], (b) MCA-based [9], (c) Bilateral [12], (d) ours with Bilateral, (e) K-SVD [7], (f) ours with K-SVD, (g) BM3D[13], and (h) ours with BM3D.

Fig. 8.Example rain removal results. Note that the input image is the noisy version of ground truth image with rain streaks presented. Rain removal outputs are produced by the methods of (a) Context-based [10], (b) MCA-based [9], (c) Bilateral [12], (d) ours with Bilateral, (e) K-SVD [7], (f) ours with K-SVD, (g) BM3D[13], and (h) ours with BM3D.by "MCA-based") [9] and rain removal via common context pattern discovery (denoted by "Context-based") [10]. These two methods can be considered as bilateral-filtering based methods, since they require a LPF stage with a bilateral filter. Table II lists the PSNR values of different bilateral-filtering based methods over three different rain images. From this table, we see that our proposed method achieved the highest or comparable PSNR values among different approaches. To show that we do not limit the use of bilateral filtering as the LPF algorithm, we further apply K-SVM andBM3D in our preprocessing stage, and compare the rain removal results with using these two denoising algorithms directly. As listed in Table III, it can be seen that our proposed method clearly improved the PSNR values than these two state-of-the-art denoising algorithms. To qualitatively evaluate the performance, we show an example rain removal result in Fig. 6, in which an input color image and its rain removed version are presented. We note that when removing rain streaks from color images, we represent such images in the YUV space and perform denoising in the Y domain. To better visualize and to compare the results, Figs. 7and 8 show example rain removal images in grayscale. From these figures, it can be observe that although Bilateral, K-SVD, and BM3D methods were able to remove most rain streaks, these denoising techniques inevitably disregarded image details(e.g., high spatial frequency parts). While applying these techniques in our LPF preprocessing stage, we were able to successfully identify/recover most non-rain image details and thus achieved improved visual quality.

Transactions on Multimedia

Fig. 9.Example rain removal results. (a) Original image with rain streaks presented,(b) the ground truth version of (a) (i.e., rain removed), (c) denoising output using bilateral filtering, (d) our denoising result. It is worth noting that, although our prior MCA-based approach successfully discarded most rain streaks without significantly degrading image quality, parts of non-rain components were also removed due to the heuristic dictionary partition by *K*-means clustering algorithm. While our prior context-based method produced comparable rain removal results, it requires one to perform context-constrained image segmentation [10] on input images, and thus significantly increases the computationalcosts.In addition, we perform single-image denoising experiments on real-world rainy images. In particular, we consider theimage frames of the video data which were utilized in [25].The videos in [25] were captured in real rainy scenes withstatic backgrounds, and the authors proposed to adjust cameraparameters for removing or enhancing the presence of rainstreaks. Thus, using their video data, we are able to collect real-world rainy images and the corresponding ground truth versions. We show example denoising results in Figs. 9 and10. From these two figures, it can be seen that our approach produced satisfactory rain removal results on real-world images with rainy scenes. We note that, although bilateral filtering was able to remove high spatial frequency patterns such as rain streaks while preserving edges in Figs. 9 and 10, a large portion of image details were also removed. As a result, an automatic and self-learning based approach likes ours is preferable in removing particular noise patterns from the input image.

B. Performance Evaluation on Image Denoising



To evaluate the performance of our approach for image denoising(with Gaussian noise), we collect and conduct experiments on several images considered in [13]. We manually add Gaussian noise with to the input noise-free images Fig. 10. Example rain removal results. (a) Original image with rain streaks presented, (b) the ground truth version of (a) (i.e., rain removed), (c) denoising output using bilateral filtering, (d) our denoising result. for addressing this task. Note that if the for the Gaussian function is known in advance, both K-SVD and BM3D algorithms will be expected to achieve excellent denoising results. However, we assume this exact parameter choice is not known(which is practical), and we simply set large standard deviation values for both algorithms. Similar to the scenarios for rain removal, this would allow us to remove high spatial frequency patterns including possible Gaussian noise from the low spatial frequency parts of the input image without finetuning the parameter . We also compare our algorithm with denoising methods not requiring the prior knowledge on for the Gaussian noise. We consider the SURE-LET algorithm [26], which relies on a purely data-adaptive unbiased estimate of the mean-squared error, so that the Gaussian noise can be removed without knowing the Gaussian parameter in advance. Table IV lists the PSNR of different denoising approaches, including ours with three different LPF/denoisng techniques applied. From this table, it can be seen that our approach produced improved denoising results than the standard LPF/denoising approaches did (i.e., Bilateral filtering, K-SVD, SURE-LET, andBM3D). For qualitative comparisons, Figs. 11 and 12 show example denoising results produced by different methods. From these figures, we see that standard LPF/denoising methods were not able to achieve satisfactory results if parameters like are not given in advance. Furthermore, although the SURE-LET based approach was able to outperform approaches using K-SVD for Gaussian noise removal, BM3D-based approaches still achieved the best denoising performance (i.e., BM3D with ours).It is worth noting that, while our method quantitatively and qualitatively outperformed others, we do not need to fine-tuneour approach with or assume such parameters are known in

PERFORMANCE COMPARISONS (IN TERMS OF PSNR) OF DIFFERENT IMAGE DENOISING APPROACHES.

NOTE THAT WE PRESENT OUR RESULTS USING THREE DIFFERENT LPF OR DENOISING TECHNIQUES

Fig. 11.Example image denoising results. Note that the input image is the noisy version of ground truth image (with Gaussian noise). Denoising outputs are produced by the methods of (a) Bilateral [12], (b) ours with Bilateral, (c) K-SVD [7], (d) ours with K-SVD, (e) SURE-LET [26], (f) ours with SURE-LET, (g) BM3D [13], and (h) ours with BM3D.Fig. 12.Example image denoising results. Note that the input image is the noisy version of ground truth image (with Gaussian noise). Denoising outputs are produced by the methods of (a) Bilateral [12], (b) ours with Bilateral, (c) K-SVD [7], (d) ours with K-SVD, (e) SURE-LET [26], (f) ours with SURE-LET, (g)BM3D [13], and (h) ours with BM3D.advance (which might not be practical). From the

above experiments, we again confirm the effectiveness and robustness of our approach for image denoising, which can be integrated with existing LPF/denoising techniques in the LPF preprocessing stage. In other words, we do not limit the use of our proposed framework to any particular LPF or denoising algorithm. Although real-time processing is not of concern of this paper, we provide the remarks on computation time for different

TRANSACTIONS ON MULTIMEDIA, learning stages of our proposed framework as follows. Inour proposed method, it takes about 100 seconds to perform denoising for an input image of 256 x 256 pixels. In particular, it takes about 3 seconds to perform bilateral filtering (i.e., identifying potential high-frequency noise patterns), 1 minute for learning the sparse-representation based dictionary, 30seconds for performing affinity propagation to identify image components of interest, and 5 seconds for reconstructing the image output. We note that, the above runtimes were obtained on an Intel Quad Core 2 PC with 2.66 GHz processors and 4GRAM.

III. CONCLUSION

In this paper, we presented a learning-based image decomposition framework for single image denoising. The proposed framework first observes the dictionary atoms from the input image for image representation. Image components associated with different context information will be automatically learned from the grouping of the derived dictionary atoms, which does not need the prior knowledge on the type of images nor the collection of training image data. To address the task of image denoising, our proposed method is able to identify image components which correspond to undesired noise patterns. Experiments on two types of single image denoising tasks (with structured and unstructured noise) confirmed the use of our proposed method, which was shown to quantitatively and qualitatively outperform existing denoising approaches.

REFERENCE

[1] 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, Jun. 2010.

[2] J. Bobin, J. L. Starck, J. Fadili, and Y. Moudden, "Sparsity and morphological diversity in blind source separation," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2662–2674, Nov. 2007.

[3] F. G.Meyer, A. Z. Averbuch, and R. R. Coifman, "Multilayered image representation: application to image compression," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 1072–1080, Sep. 2002.

[4] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 882–889, Aug. 2003.

[5] J. M. Fadili, J. L. Starck, M. Elad, and D. L. Donoho, "Mcalab: reproducible research in signal and image decomposition and inpainting," *IEEE Comput. Sci. Eng.*, vol. 12, no. 1, pp. 44–63, Jan.-Feb. 2010.

[6] M. Aharon, M. Elad, and A. M. Bruckstein, "The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322,Nov. 2006.

[7] 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.

[8] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[9] L.-W. Kang, C.-W.Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1742–1755, Apr. 2012.

[10] D.-A. Huang, L.-W.Kang, M.-C.Yang, C.-W.Lin, and Y.-C. F.Wang, "Context-aware single image rain removal," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Melbourne, Australia, Jul. 2012, pp. 164–169.

[11] J. Bobin, J. L. Starck, J. M. Fadili, Y. Moudden, and D. L. Donoho, "Morphological component analysis: an adaptive thresholding strategy," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2675–2681, Nov. 2007.

[12] 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.

[13] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[14] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 13, pp. 607–609, Jun. 1996.

[15] S.Mallat and Z. Zhang, "Matching pursuitswith time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[16] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, Feb. 2009.

[17] J.Mairal, F. Bach, J. Ponce, andG. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.

[18] B. J. Frey and D.Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

[19] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, vol. 1, pp. 886–893.

[20] J. Bossu, N. Hautière, and J. P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *Int. J. Comput. Vis.*, vol. 93, no. 3, pp. 348–367, Jul. 2011.

[21] K. Garg and S. K. Nayar, "Vision and rain," *Int. J. Comput.Vis.*, vol. 75, no. 1, pp. 3–27, 2007.