Eradication of Mist and Dew from Ongoing Video and Images Using Shallow Rank Pattern and Carnal Interchange

A.Santhoshkumar[#1], K.Vigneshkumar[*2], K.Santhakumar[#3]
[1,2]*Electronics And Communication Engineering, Nandha Engineering College(Autonomous)*
Tamilnadu, India
[1]*santhoshsandy.arumugam@gmail.com*
[2]*vigneshkumar087@gmail.com*
[3]*Assistant professor/ECE*
Tamilnadu, India
[3]*ecebank@gmail.com*

Abstract **- The Video Deraining Algorithm (VDA) is used to eliminate drizzle, blizzard, hail and mist from a video sequence using temporal interaction and low-rank matrix fulfilment. If the drizzles are too small and move at the high rate it will affect the optical flow estimation between consecutive frames. To avoid these drizzles i.e rain and snow, an initial rain map is subtracting temporally deform frames from a current frame. After that, they decompose the initial rain map into basis vectors based on the sparse representation and classify those basis vectors into drizzle ones and outliers with a Support Vector Machine (SVM). Finally by excluding the outliers, the detected results will remove the drizzles by employing a low-rank matrix fulfillment technique. Thus the overall efficiency of VDA is recommend to high-fidelity video dedrizzling and improves the performance by removing the hail and dew drops in a video.**

Keywords **– Video Deraining Algorithm, Support Vector Machine, Deraining, Stereo Video deraining.**

I. INTRODUCTION

The video capturing devices, such as smart phones and digital cameras, are relatively cheap and popular nowadays, which facilitate the widespread production and consumption of high quality video sequences. Outdoor video sequences, however, are often degraded due to various weather conditions such as haze, fog, rain, and snow. Image processing and computer vision systems, including tracking and surveillance, may not work properly on these degraded videos. Therefore, attempts have been made to compensate for the weather- dependent degradation and enhance outdoor video sequences. Early attention has been drawn to haze removal. Haze exhibits a static distribution of tiny particles in the air, which degrades image contrast [1]. Thus, most dehazing algorithms attempt to enhance the contrast.

On the contrary, as illustrated in Fig. 1, rain or snow causes elongated streaks, the locations of which are randomly distributed within an image also based on time varying which

Fig 1. Desnowing of a frame (a) Input frame (b) Desnowing result (c) Snow map

dynamically in the case of a video sequence. Moreover, a rain or snow streak is larger than a haze element, hiding the colors of objects behind. Therefore, video deraining or desnowing, which recovers rain-free or snow-free video sequences, is more challenging than dehazing. Several deraining algorithms have been proposed [2]–[10]. However, most deraining algorithms do not consider global motions, object motions or various sizes of rain streaks, and thus they may fail to remove rain streaks clearly.

Thus, the video deraining algorithm using temporal correlation and low-rank matrix completion generate an initial rain map from the differences between the current frame and the warped adjacent frames using a Support Vector Machine (SVM). By removing the outliers, we refine the rain map and detect rain streaks. Finally, the result detected rainy pixels using a matrix completion algorithm, which performs the Expectation Maximization (EM) iterations for the low-rank approximation. Moreover, the proposed video deraining algorithm is used to stereo video deraining. Experimental results demonstrate that the proposed algorithm outperforms conventional algorithms, by removing rain streaks efficiently and reconstructing scene contents faithfully.

II. RELATED WORKS

Kang et al. [3] proposed a single image deraining algorithm, assuming that rain streaks can be represented by a selected set of basis vectors. They decomposed high frequency components in a rainy image into basis vectors via sparse representation. They then clustered the basis vectors into rainy components and structural components. They recovered a rain-free image by keeping the structural components only.

Hase et al. [4] proposed an early video deraining algorithm, which used a temporal median filter to restore pixel values. The median filtering is effective for deraining static scenes, but it may cause blurring artifacts around dynamic objects. Garg and Nayar [5] detected rainy pixels in a frame, which exhibited brighter intensities than the corresponding pixels in adjacent frames. They reduced outliers, falsely estimated as rain streaks, based on the constraints that rain streaks should have similar directions and their colors should be related to the background colors. This approach is physically sound [2], but yields good results only when rain streaks are distinguishable from moving objects.

Garg and Nayar [6] also proposed a hardware-based scheme, which suppresses rain streaks by increasing the exposure time of a camera or focusing on objects behind rain streaks. It implicitly performs low-pass filtering; resulting in spatio-temporal blurring and it may not handle rainy sequences with dynamic objects or diverse scene depths effectively. Zhang et al. [7] removed rain streaks by enforcing the photometric constraints: a pixel color should be dominated by the background color, and the brightness changes of rainy pixels should be similar through an entire sequence. However, they replaced a rainy pixel with a temporal average of background pixels, yielding blurring artifacts.

Barnum et al. [8] proposed a deraining algorithm based on frequency analysis. Their algorithm transforms rainy frames into the Fourier domain to detect rain streaks, which tend to have elliptical shapes and appear at similar locations in the frequency domain. It removes rain well even in the case of dynamic scenes, but it cannot remove thick rain streaks properly. Another approach to deraining is based on optical flow estimation [9], [10]. A current frame is first reconstructed by warping an adjacent frame according to the optical flow. In general, an overall scene structure is identical between the current frame and the warped frame, except for rain streak regions. Therefore, a rain-free image can be obtained by replacing pixel values, which are bigger in the current frame than in the warped frame, with those of the warped frame. However, [9], [10] do not consider outliers, such as occluded regions, where the optical flow estimation fails.

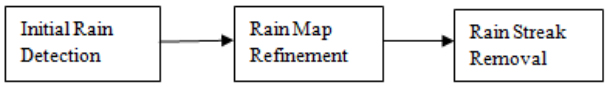

Fig 2. Overview of proposed algorithm

In Fig 2. an initial rain map from an image frame, which is then refined based on sparse representation and classification. Finally, the result is reconstructing a rain-free frame by exploiting the information in adjacent frames.

III. PROPOSED ALGORITHM

First step is to obtain an initial rain map by computing the difference between a current frame and an optimally warped frame. Second step is to decompose the initial rain map using sparse basis vectors, and employ an SVM classifier to dichotomize those vectors into valid ones and outliers. Then reconstruct a refined rain map by employing the valid vectors only. Finally, replace rainy pixels with rain-free values, by formulating the rain streak removal as a matrix completion.

A. Initial Rain Detection

Fig. 3 show a current frame I_k and its previous frame I_{k-1} in the video sequence "Mailbox." The rainstreaks appear randomly and each streak occupies a relatively small area in a frame and moves fast between consecutive frames. Also, when a rain streak passes across a pixel, the pixel value becomes brighter than its original color [5]. Hence detect a rainy pixel, which has a larger value in a current frame than in adjacent frames. This approach, however, is prone to false detection, since the same pixel coordinates in different frames may not represent the same scene point. A video sequence may contain dynamic objects or be captured with a moving camera. This approach, however, is prone to false detection, since the same pixel coordinates in different frames may not represent the same scene point. A video sequence may contain dynamic objects or be captured with a moving camera.

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

Fig 3. Initial Rain Detection (a) Current frame (b) Previous frame (c) Warped previous frame (d) difference between (a) and (c)

To compensate for these mismatches between consecutive frames, we warp the previous frame into the current frame by estimating the optical flow field. An optical flow estimation algorithm finds a dense motion field between two consecutive image frames [13]–[15]. For each pixel in a reference frame, a motion vector is determined to find the most similar pixel in a target frame, while maintaining the similarity of the motion vectors among neighboring pixels. The optical flow estimation is often formulated as a minimization problem, in which an energy function E is given by (1)

$$E(U) = E_d(U) + \lambda E_s(U) \qquad (1)$$

where **U** is a flow field and λ is a regularization parameter. The data term E_d measures the similarity between corresponding pixels in the reference frame $I1$ and the target frame $I2$ in (2),

$$E_d(U) = \int \Psi\left(I_2\big(x + u(x)\big) - I_1(x)\right)^2 dx \qquad (2)$$

where $\mathbf{u}(\mathbf{x})$ is the optical flow vector of pixel **x**, and ψ is a penalty function. The smoothness term E_S constrains neighboring vectors to be similar (3).

$$E_s(U) = \int \Psi\big(u(x)\big)dx \qquad (3)$$

Finally, we obtain an initial rain map R as the difference image between the current frame I_k and the hybrid warped frame $\tilde{I}_k$,

$$R(x) = \max(I_k(x) - \tilde{I}(x)), 0 \qquad (4)$$

where negative differences are truncated to 0, since rainy pixels are assumed to be brighter than their original colors (4). Note that we use only the luminance components of the initial rain map in the following steps of the rain map refinement and the rain mask generation.

B. Rain Map Refinement

To obtain a binary rain mask by thresholding an initial rain map. Fig. 4 shows an initial rain map and the resultant binary rain masks using different thresholds. Note that the binary rain masks contain falsely detected outliers, caused by warping errors or brightness changes between frames, or fail to detect some valid rain streaks. In general, as we decrease the threshold, many outliers are falsely detected as rain streaks. On the contrary, as we increase the threshold, we can reduce such outliers but also miss valid rain streaks. Moreover, outliers often overlap with valid rain streaks. Therefore, to detect valid rain streaks reliably while suppressing outliers, refine an initial rain map before the thresholding.

To refine an initial rain map, we exploit the directional property of rain streaks: rain streaks tend to have elliptical shapes, whose major axes deviate little from the vertical direction. In contrast, falsely detected outliers have arbitrary shapes or yield random directions of major axes. Therefore, the matrix R by multiplying the new dictionary D with the coefficient matrix A in (5).

$$R = D.A \qquad (5)$$

The Morphological Component Analysis (MCA) decomposes a given signal into basis vectors based on sparse representation, and then reconstructs the signal by employing selected basis vectors only. From the binary rain mask M, we further remove outliers by erasing small connected components of size 5 or less. Then, the result performs the dilation operation on the rain mask, since the refinement procedure may distort the boundaries of rain streaks.

Finally, we generate a binary rain mask M from the refined rain map is given by (6)

$$M(x) = \begin{cases} 1 \; if \; R(x) > \epsilon \\ 0 \; Otherwise \end{cases} \qquad (6)$$

From the binary rain mask M, we further remove outliers by erasing small connected components of size 5 or less. Then, we perform the dilation operation on the rain mask, since the refinement procedure may distort the boundaries of rain streaks.

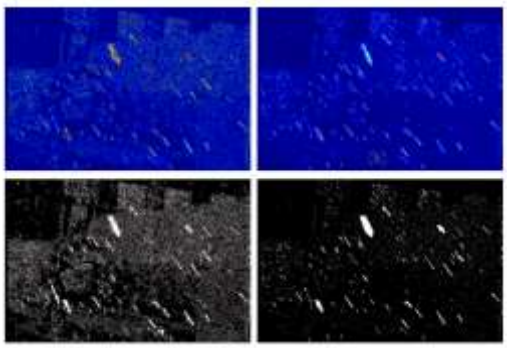

Fig 4. Rain Map Refinement
(Binary mask of initial & refined rain map)

C. Rain Streak Removal

The restore pixel values, detected as rainy in Fig 5, by exploiting temporal redundancies in adjacent frames. Specifically, we formulate the rain removal as a low-rank matrix completion problem. We first partition the current frame I_k into disjoint blocks. For each block **b**, we search for the l most similar blocks from each of the four adjacent frames: I_k–2 , I_k–1, I_k+1, I_k+2. Notice that we do not find similar blocks from the current frame. This is because similar blocks in the current frame tend to be selected near the given block **b** and affected by the same rain streak, degrading the deraining performance. To measure the similarity between two blocks, we compute the sum of the squared differences between rain-free pixels only. Then, we define a matrix **B**, by concatenating the given block **b** in the current frame and its $4l$ most similar blocks $\mathbf{b}_i$'s in the adjacent frames (7),

$$B = [b, b_1, b_2, \ldots, b_{4l}] \qquad (7)$$

where each block is represented by a column vector. We also define the binary rain mask matrix **M** for **B**, given by (8)

$$M = [m, m_1, m_2, \ldots, m_{4l}] \quad (8)$$

1) EM-Based Rain Streak Removal Algorithm

Input : pixel matrix B and mask matrix M
Output : $X=X^{(t)}$
Steps

1. Initialize t=0 $Y^{(1)}=B$
2. Repeat t=t+1
3. $X^{(t)}=1$
4. $Y^{(t+1)}= U^* V^t$
5. Until $t=t_{max}$

End.

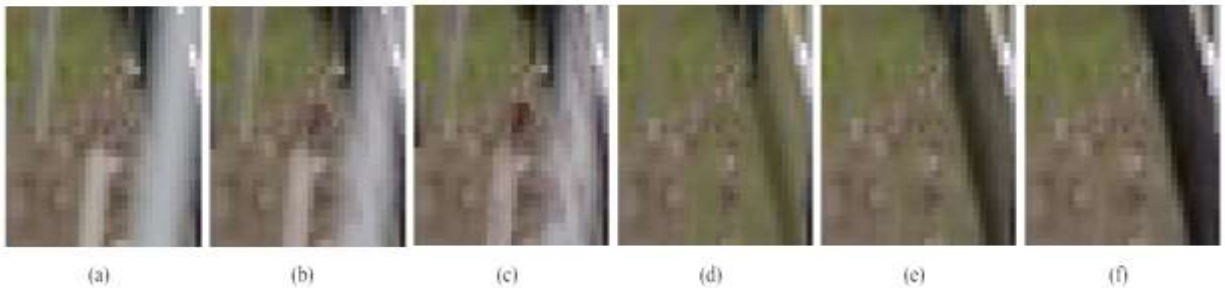

Fig 5. Iterative Removal of Rain streaks (a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 10 (e) Iteration 20 (f) Iteration 50

IV. EXPERIMENTAL RESULTS

Fig 6. Evaluates the performance of the proposed EM-based matrix completion algorithm on various video sequences. More results with those of different matrix completion algorithms achieve better performance than the conventional algorithms. In addition, the algorithm compare the rain removal performance with image inpainting techniques as toatl variation inpainting and ex-emplar based inpainting methods. And also the algorithm reconstructs the original scene contentsfaithfully.

Fig 6. Desnowing Experimental Results

A. *Execution Time*

The proposed algorithm is implemented in Matlab without code optimization. Note that the sparse representation for the rain map refinement and the block matching for the rain streak removal spend most of the execution time. Such high complexity could be reduced by approximating the sparse representation and employing a more efficient block matching algorithm.

V.CONCLUSION

A video deraining algorithm, which exploits temporal correlation in a video sequence, based the low-rank matrix completion. The proposed algorithm obtains an initial rain map, by warping previous and next frames and comparing those warped frame with a current frame. Then it refines the initial rain map by removing outliers based on the sparse representation and the classification. Finally, the proposed algorithm fills in rainy pixels using the EM-based low-rank matrix completion. They also extended the proposed algorithm to stereo video deraining. Extensive experimental results demonstrated that the proposed algorithm removes rain and snow streaks more efficiently, while preserving scene structures more faithfully, than the conventional algorithms.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments and insightful suggestions that improved the quality of this manuscript.

REFERENCES

[1] S. G. Narasimhan and S. K. Nayar, "Vision and theatmosphere," *Int.J. Comput. Vis.*, vol. 48, no. 3, Jul. 2002.
[2] K. Garg and S. K. Nayar, "Vision and rain," *Int. J. Comput. Vis.*, vol. 75, no. 1, pp. 3–27, Oct. 2007.
[3] 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.
[4] H. Hase, K. Miyake, and M. Yoneda, "Real-time snowfall noise elimi- nation," in *Proc. IEEE ICIP*, Oct. 1999, pp. 406–409.
[5] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun./Jul. 2004, pp. I-528–I-535.
[6] K. Garg and S. K. Nayar, "When does a camera see rain?" in *Proc. 10th IEEE ICCV*, Oct. 2005, pp. 1067–1074.
[7] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng, "Rain removal in video by combining temporal and chromatic properties," in *Proc. IEEE ICME*, Jul. 2006, pp. 461–464.
[8] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *Int. J. Comput. Vis.*, vol. 86, nos. 2–3, pp. 256–274, Jan. 2010.
[9] M. Shen and P. Xue, "A fast algorithm for rain detection and removal from videos," in *Proc. IEEE ICME*, Jul. 2011, pp. 1–6.
[10] V. Sathaseelan and V. K. Asari, "A phase space approach for detection and removal of rain in video," *Proc. SPIE*, vol. 8301, p. 830114, Jan. 2012.

