# A Novel Approach for Video De-Noising using Convex Optimization

Alaa E. Abdel-Hakim

*Electrical Engineering Department*
Assiut University, Assiut, Egypt, 71516
alaa.aly@eng.au.edu.eg
On a Sabbatical leave at *Computer Science Department,* University College
Umm-Alqura University, Makkah Al-Mukarramah, Saudi Arabia

***Abstract* - We model the problem of video denoising as an optimization problem. Inter-frame information is exploited to provide low-rank characteristics of an input video stream. The noise-free video frames are extracted as the low rank components of a convex optimization model. We use robust principle component analysis (RPCA) to extract away the noise from the input video streams. The fast version of RPCA, fast RPCA (FRPCA), is used for realtime denoising. A noisy video stream is considered as a 3D noisy signal which is a summation of a 3D noise-free signal and sparse additive noise. Exact augmented Lagrangian multipliers (EALM) method is used to solve the model for the low-rank terms, which represent the noise-free frames. The proposed approach has several advantages over existing video denoising approaches. It is model-independent, i.e. it does not require shape, appearance, or speed models. Also, it does not need prior information about the acquisition environment. Three different sets of data were used for evaluation: synthetic data for simulation experiments to provide quantitative results, real static videos, and real dynamic videos. The evaluation results proved the effectiveness of the proposed approach when compared to existing approaches.**

***Keywords - Video denoising, Rain removal, Low-rank recovery, RPCA, FRPCA***

## I. INTRODUCTION

Video denosing is a critical bottleneck for several applications. Video noise can be a result of acquisition defects, signal transmission, or improper imaging conditions. Existence of undesirable noise can ruin the performance of many vision-based systems. Bad weather conditions like rain, fog, snow, or haze are examples of video noise which create big challenges for many applications, such as object detection [1]–[3], image registration [4], event detection [5], attention modeling, and tracking [6]. Therefore, the removal of distortions that are caused by weather, specially rain, is growing rapidly and receiving a lot of interest.

Rain/snow removal problem is one of the main applications of video denoising. Several researchers have approached this problem in different ways. Many of them considered model-based methodologies. For instance, in [7], Garg and Nayar developed an approach that uses two models: one for capturing the dynamics of rain through exploiting correlation information between consequent frames. The second model is a motion-blurred-based one for modeling the rain photometry. Other approaches employ shape and/or appearance models to characterize the rain streaks for removal [4], [8]–[11].

Other research studies adopt early hardware adjustment strategies to reduce the effect of the rain streaks during the acquisition process itself. For example, Garg and Nayar [12] have proposed an approach to select camera parameters, e.g. the exposure time and field depth, to reduce the rain effect during at the acquisition time. However, this approach cannot deal with alreadycaptured videos. Some recent studies, e.g. [13], attempt to solve the problem for a single image by decomposing the image using morphological component analysis. Such attempts are quite useful when missing valuable temporal information that video data provides.

In this paper, we propose a general solution to the video denoising that is independent on the nature of the added noise. In other words, the proposed approach does not make prior assumptions about the noise source, i.e. it does not matter if it is caused by acquisition, transmission, or environmental conditions. The proposed approach is based on low-rank recovery. The additive video noise is usually sparse in its nature. Thus, the noise-free scene is a low-rank when considering adjacent frames. Therefore, the proposed approach extracts the low-rank components between consecutive frames away from the sparse components.

Figures 1(a,d) show the rain and snow effects examples of additive video noise and the denoising results using the proposed approach. The proposed approach has several advantages over the state of the art:

- It is model-independent.
- It has no constraints on the acquisition hardware.
- All the data used to denoise the input signals is already included in the input video sequence and there is no need for prior information.

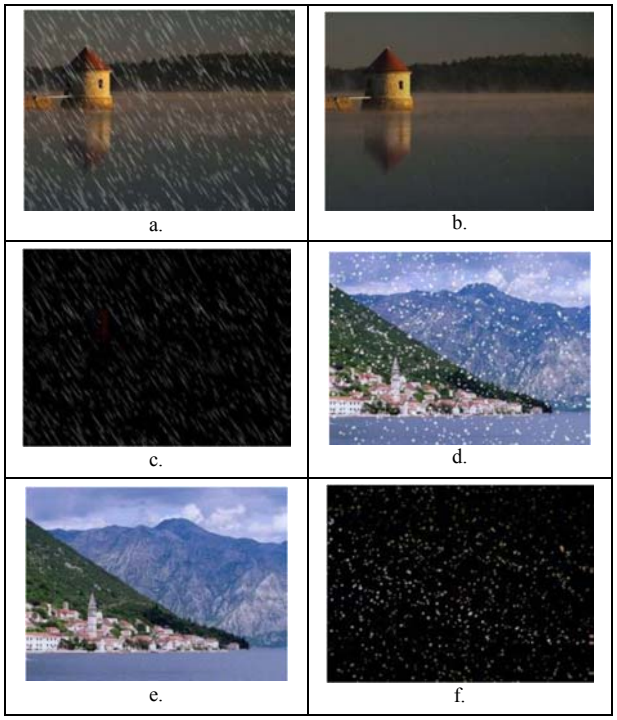



Figure 1. Exemplar samples of video denoising for removing rain and snow effects using the proposed approach. [a and d] The rain and snow distorted images, respectively [b and e] The removed rain and snow signals, respectively. [c and f] The cleaned images. Note that the water splashes above the water-trees septation line of the top image are preserved and were not confused with the rain drops.

## II. THE PROPOSED APPROACH

In this section, we explain the proposed approach. Firstly, we give a formulation of the denosing problem from the perspective of low-rank recovery. Then, we explain the solution of the proposed model.

### *A. Problem Statement*

We assume that the original video signal is a noisefree 3D signal, i.e. spatial 2D signal plus the time dimension. This noise-free signal is corrupted by some noise, which can be acquisition noise, additive noise caused by transmission medium, snow flakes, or rain drops. The main goal is to separate the original signal from the noise components. So, the problem can be formulated as follows:

The input is a video sequence that consists of N frames. The kth frame is Ik 2 Rh_w, where k = 1 : N, h and w are the height and the width of Ik, respectively. For the purpose of complying with the optimization model, let Ik be rearranged in a columnvector form Dk 2 Rm, m = h:w. Assume that a given frame Ik is composed of two components: a low rank noise-free component Ak, and an additive sparse noise component, Ek. In other words, each frame of the input video sequence contains some sparse additive noise signal added to the original noise-free signal. So, $D^k$ can be modelled as:

$$D = A + E \in \mathrm{R}^{m \times n} \quad (1)$$

where n is the number of frames which are to be used for low-rank signal recovery. Therefore, the problems turns out to be as follows:

*Given* $D = A + E \in R^{m \times n}$, *derive* $I^{*k} \in R^{h \times w}$, *where* $I^{*k}$ *is the noise-free* $k^{th}$ frame, $k = 1 : N$.

.

### *B. De-noising Model*

Computing the Singular Value Decomposition (SVD) of D optimally estimates A. The solution is estimated by projecting the columns of D onto the subspace spanned by the principal left singular vectors of D [14], [15]. Estimation of A using this procedures suffers from gross errors and distortions, even if they are sparse [16]. So, Wright et.al. [17] showed that low rank matrix A can be recovered from D = A + E by solving the optimization problem shown in (2).

$$(\mathrm{A;E}) = \arg_{A,E} \min \mathit{rank}\,(\mathrm{A}) + \lambda \| \mathrm{E} \|_0 \quad (2)$$

$$s.t. D = A + E$$

In other words, the solution of (1) consists of the components A;E which minimize both of the rank of A and the sparsity of E.

The optimization model shown in (2) is not easy to be solved from an algorithmic perspective as it is highly nonconvex problem and contains two NP-hard terms [17], [18]. It has been shown in [17] that this problem can be solved by acceptably modifying the optimization model of (2) into a convex optimization problem as shown in (3).

$$(A, E) = \arg_{A,E} \min \| \mathrm{A} \|_* + \lambda \| \mathrm{E} \|_1 \quad (3)$$

$$s.t. D = A + E$$

where $\| . \|_*$ and $\| . \|_1$ are the nuclear and L1-norms of a matrix, respectively, and $\lambda$ is an arbitrary constant.

Lin et.al. [14] have proposed an algorithm for solving (3). Their algorithm depends on using a convex optimization model to solve the low-rank recovery problem. Lin et.al [19] have presented a faster Augmented LaGargian Multipliers method, (ALM), for solving the RPCA problem. Two versions of ALM were presented:

Exact Augmented LaGargian Multipliers, (EALM) and Inexact Augmented LaGargian Multipliers, (IALM). Both

algorithms were proven to run about five times faster than the state-of-the-art algorithms, which were presented in [14]. In this work, we adopt the EALM algorithm to recover the low-rank and the sparse components of (3).

For real-time operation, Fast Robust Principal Component Analysis (FRPCA) [20] is used. In FRPCA, the low rank matrices of the incrementally-observed frames are estimated using a convex optimization model that exploits information obtained from the pre-estimated low-rank components of earlier observations.

## III. EXPERIMENTAL SETUP

We design our experiments to evaluate the proposed approach in terms of its end goals. In other words, measures are needed to indicate how accurate the proposed approach is in separating the original frames from the noisy ones. We consider distortions caused by rain drops and snow flakes as examples of additive noise. For this purpose, we conduct two main kinds of experiments. The first one is in the direction of developing a quantitative simulation-based evaluation procedure using synthetic video data. The second one is to apply the proposed approach to real video sequences, which contain different kinds of videos, e.g. static and dynamic scenes. The used real sequences allow visual comparison of the performance of the proposed approach with the performance of other approaches in the literature.

### *A. Simulation setup*

For simulation-based experiments, we synthetically generate rain and snow-distorted video sequences. The synthetic sequences were generated by adding rain and snow distortions with various densities. Specifically, ten rain densities ranging from 10% to 30% were created. Similar ten snow densities were created totalling twenty different synthetic video sequences. The performance of the proposed approach is evaluated by calculating the error in the recovered frames when compared to the ground truth frames. The error is calculated on the pixel-level. Then, it is averaged over all the frames of the entire sequence.

Five sets of results for rain and other five for snow are obtained by changing the number of the used frames in the estimation of the low-rank and the sparse terms. In other words, n of (1) takes five different values: 2, 5, 10, 15, and 20. It is expected that as n increases, the accuracy of the recovered frames will be improved. However, increasing n has a negative side effect in the case of dynamic videos, specially in existence of fast moving objects in the scene. The next subsection discusses this limitation in more details. Plotting the average error values for these different settings gives a clear insight about the effectiveness of the proposed approach in removing the rain and snow effects.

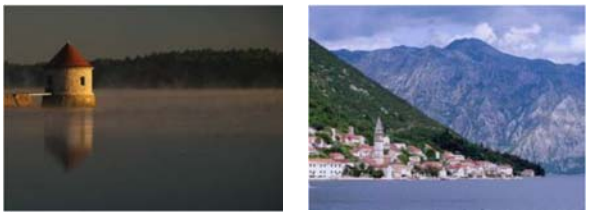


Figure 2. The original images that were used to synthesize the rain and snow-distorted video in the simulation experiments for the purposes of quantitative evaluation purposes. Images are taken from the ImageCLEF dataset, which are collected from Flickr [21].

### *B. Evaluation using real data*

In this section, we explain how we use real rainy video sequences to evaluate the performance of the proposed approach. We use two kinds of videos: static and dynamic videos. Static videos do not contain moving objects. Such kind of scenes gives relatively better rain/snow removal results than the dynamic ones. This is due to the nature of low-rank operation that tends to fetch the common components between frames in low-rank terms more than differences. So, dealing with dynamic videos is a bit more challenging. As a solution to this problem, we use fewer number of adjacent input frames for low-rank calculation, i.e. smaller n. Using smaller n guarantees maintaining the image of the moving objects. This is resulted form the fact that the motion of most moving objects in videos takes several frames to get reflected in a form of visual displacement, especially with high-frame-rate cameras. This means that the differences, with respect to the moving objects, between the adjacent frames are small. On the other side, this is not the case for the rain drops, which are usually faster than the normal moving objects in the video sequence.

Increasing the value of n causes some blurring in the areas of the moving objects. Nonetheless, too small n does not remove the rain drops or snow flakes completely. Therefore, a trade-off is needed depending on the nature of the input videos and the speed of their moving objects. These effects will be illustrated in the next section.

## IV. EVALUATION RESULTS

As illustrated in the previous section, we present two kind of results: quantitative, which depend on simulation experiments, and qualitative for the purposes of visual comparison with the state of the art. The results of real data are presented for both static and dynamic kinds of videos.

Figure 2 shows the natural images that were distorted by synthetic rain/snow signals to construct the used synthetic videos for the simulation experiments. As mentioned earlier, the value of n in (1) affects the denoising accuracy. Figure 3 illustrates the feeling of this effect. It shows visual illustration of the impact of changing the

value of n in (1) on the overall performance of the removal algorithm.

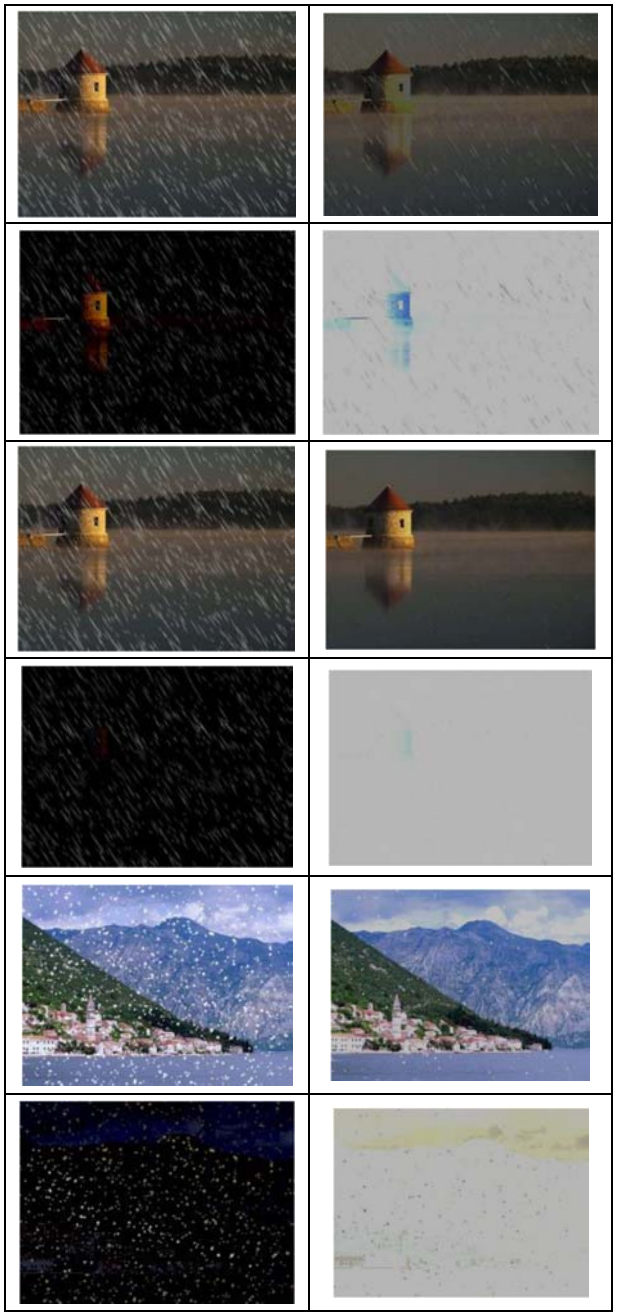


Figure 3. A visual illustration of the impact of changing the value of n in equation (1) on the overall performance of the removal algorithm. n = 2 for the first and third cluster of 4 images and n = 10 for the second cluster. For each cluster, the top left shows distorted frames. The second image shows the recovered frames. The third image shows the removed rain/snow. The forth shows the difference between the original and the recovered frames. The 4th images are inverted for better visibility.

Viewed in clusters of 4 images, the top left image in each cluster shows a rain distorted frame. The lower rows show similar frames for snow distortions. By comparing between the first and third clusters versus the second, this effect can be clearly appreciated. The bottom right image highlights the error by showing the difference between the recovered frames and the ground truth.

The algorithm was tested using the simulation data by running it on various videos with different rain and snow densities and different numbers of frames that are used for recovery, i.e. different values of n. The average error is measured on the pixel level and is plotted in Fig. 4. As discussed earlier, as n increases the error gets smaller. The worst case occurs with the smallest n, which equals two. As n exceeds three or four frames, the average pixel error falls below 1%.

For comparison purposes, we use the same video sequences that Garg and Nayar have used with their method [7]. Figure 5 shows the results of an exemplar frame of a static video. It is clear that as n increases, the performance gets better. However, beyond some value of n the performance is fixed since the error of the recovered frames become minimal. When comparing between the obtained results obtained using the proposed approach and those of [7], we can see that our results for $n > 10$ is performing good and at the same time it preserves the original frame color without big distortion. For example, the plant color in the lower right area of the frame was not distorted like the other case. Figure 6 shows similar results for a dynamic video.

In this case, the increase of n is restricted. Depending on both of the frame rate and the speed of the moving object, a blurring effect starts to appear near the edges of the moving object as n increases. So, a tradeoff here is a must. In this work, we empirically select the best tradeoff values of n. For most of videos whose moving objects have moderate speed, setting n between 5 and 10 is usually a good choice. When comparing our results for $n = 5$ and $n = 10$ with those of [7], we can see that our approach has less artifacts. This is illustrated by a closer look at the exemplar frame of Fig. 6. We make a closeup to an area at the top of the umbrella to the left of the image, which is shown in Fig. 7. It is clear that Fig. 7-a has less artifacts than Fig. 7-b. These artifacts usually resulted from continuous rain streaks that are falling from the umbrella's edge. The blurring of the umbrella's edge does not affect the quality of the recovered frame, because the movement rate of the pixels in the person region is still lower than five times the frame rate.

## V. CONCLUSION

In this paper, we presented a novel method for video denoising. We odeled the problem of video denoising as a convex optimization problem. The noisefree component is assumed to be low-rank, while the noise component is assumed to be sparse. The solution under this assumption is found by separating the noisefree signal from the added

noise using RPCA or its fast version FRPCA for realtime applications. The proposed approach does not require special knowledge about the source, shape, or appearance of the added noise. The results showed that the proposed approach outperforms existing methods.

REFERENCES


[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005., vol. 1. IEEE, 2005, pp. 886–893.

[2] O. Ludwig, D. Delgado, V. Gonc¸alves, and U. Nunes, "Trainable classifier-fusion schemes: an application to pedestrian detection," in 12th International IEEE Conference on Intelligent Transportation Systems, 2009. ITSC'09. IEEE, 2009, pp. 1–6.

[3] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE, 2008, pp. 1–8.

[4] M. Roser and A. Geiger, "Video-based raindrop detection for improved image registration," in IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops). IEEE, 2009, pp. 570–577.

[5] M. S. Shehata, J. Cai, W. M. Badawy, T. W. Burr, M. S. Pervez, R. J. Johannesson, and A. Radmanesh, "Video-based automatic incident detection for smart roads: The outdoor environmental challenges regarding false alarms," IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 2, pp. 349–360, 2008.

[6] L. Itti, C. Koch, and E. Niebur, "A model of saliencybased visual attention for rapid scene analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 11, pp. 1254–1259, 1998.

[7] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004, vol. 1. IEEE, 2004, pp. 528–535.

[8] J. C. Halimeh and M. Roser, "Raindrop detection on car windshields using geometric-photometric environment construction and intensity-based correlation," in IEEE Intelligent Vehicles Symposium. IEEE, 2009, pp. 610–615.

[9] J. Bossu, N. Hauti`ere, and J.-P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," International journal of computer vision, vol. 93, no. 3, pp. 348–367, 2011.

[10] N. Brewer and N. Liu, "Using the shape characteristics of rain to identify and remove rain from video," in Structural, Syntactic, and Statistical Pattern Recognition. Springer, 2008, pp. 451–458.

[11] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," International journal of computer vision, vol. 86, no. 2-3, pp. 256–274, 2010.

[12] K. Garg and S. K. Nayar, "When does a camera see rain?" in Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005., vol. 2. IEEE, 2005, pp. 1067–1074.

[13] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition, IEEE Transactions on Image Processing, vol. 21, no. 4, pp. 1742–1755, 2012.

[14] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–18, 2009. [Online]. Available: http: //decision.csl.illinois.edu/_abalasu2/Files/Lin09-pp.pdf

[15] I. Jolliffe, Principal Component Analysis. New York, New York: Springer Verlag, 1986.

[16] A. E. Abdel-Hakim and M. El-Saban, "Distortion impact on low-dimensional manifold recovery of highdimensional data," in Taibah University International Conference on Computing and Information Technology, ICCIT'12, Al-Munawwara, Saudi Arabia, March 2012, pp. 204–209.

[17] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization," in proceedings of Neural Information Processing Systems (NIPS), December 2009.

[18] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," Theoretical Computer Science, vol. 209, no. 1, pp. 237–260, 1998.

[19] Z. Lin, M. Chen, and L. Wu, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," Analysis, vol. math.OC, pp. 2209–2215, 2009. [Online]. Available: http: //arxiv.org/abs/1009.5055

[20] A. E. Abdel-Hakim and M. El-Saban, "Frpca: Fast robust principal component analysis for online observations," in 2012 21st IEEE International Conference on Pattern Recognition (ICPR). IEEE, 2012, pp. 413–416.

[21] http://www.imageclef.org.

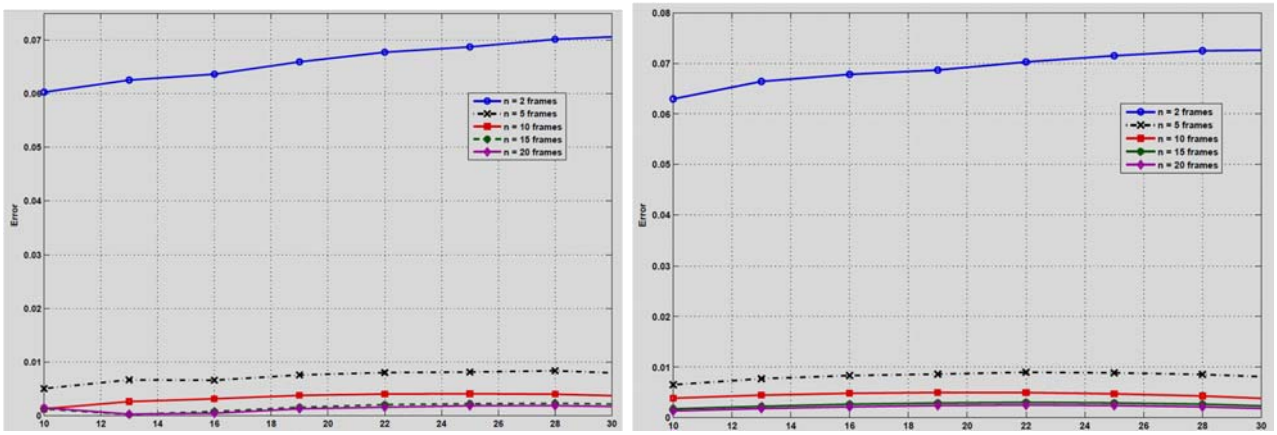




Figure 4. Quantitative results using the simulation setup. The average error values are calculated on the pixel-level for different values of n and different distortion levels.

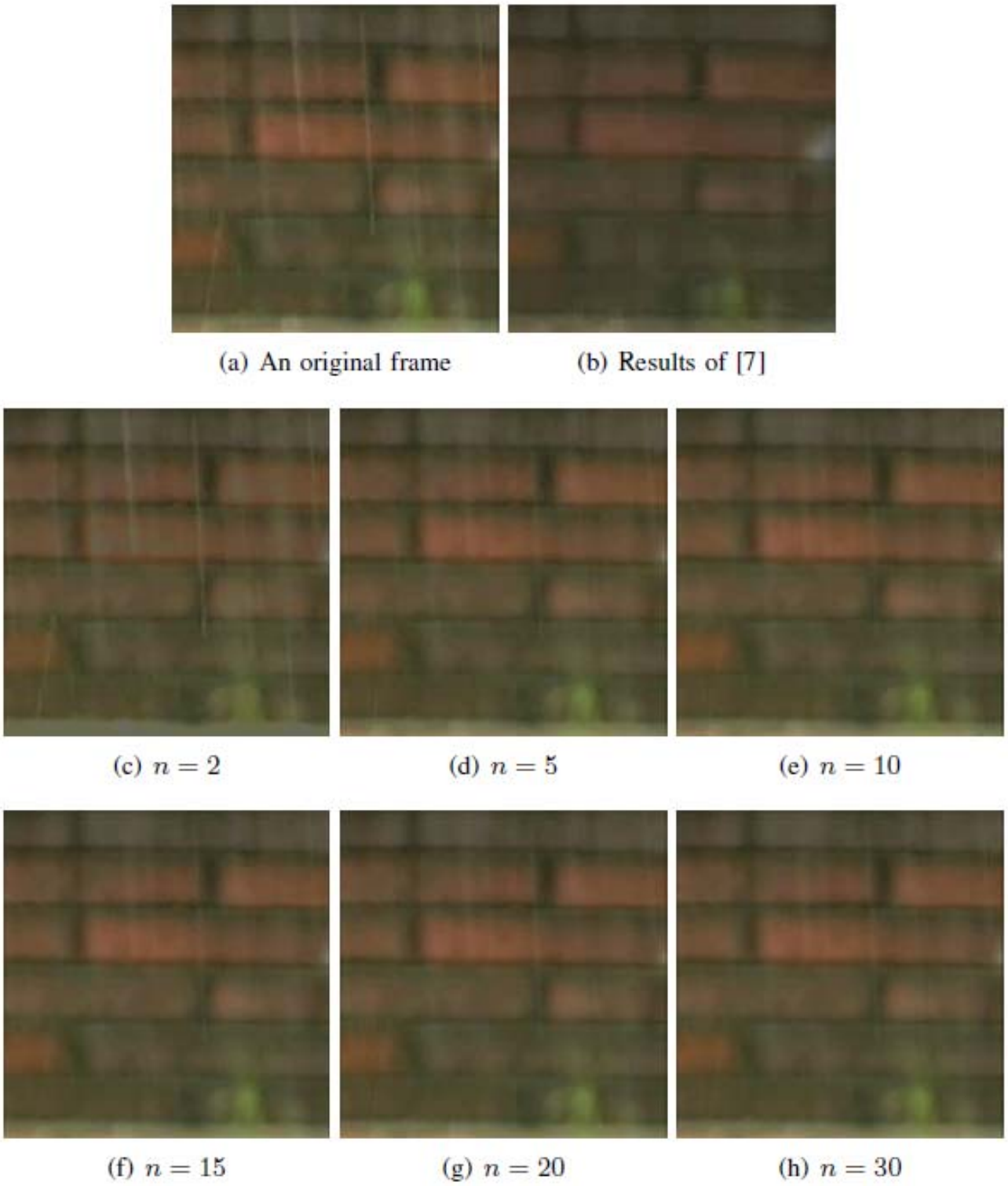
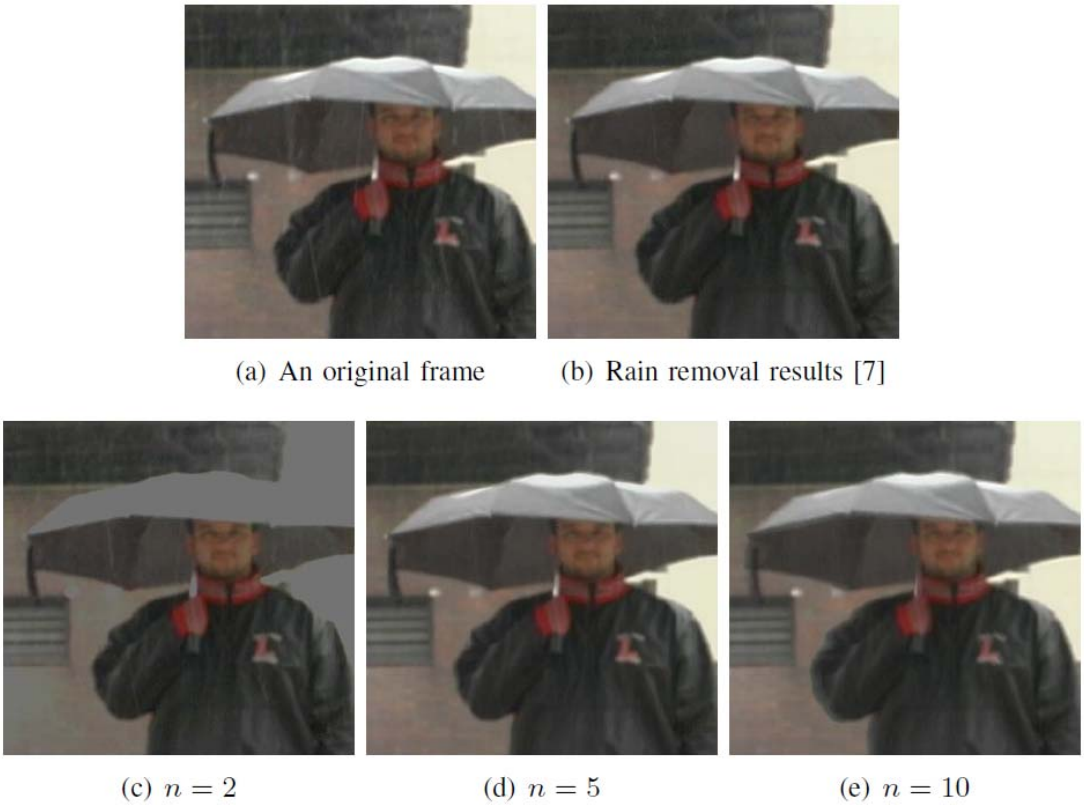


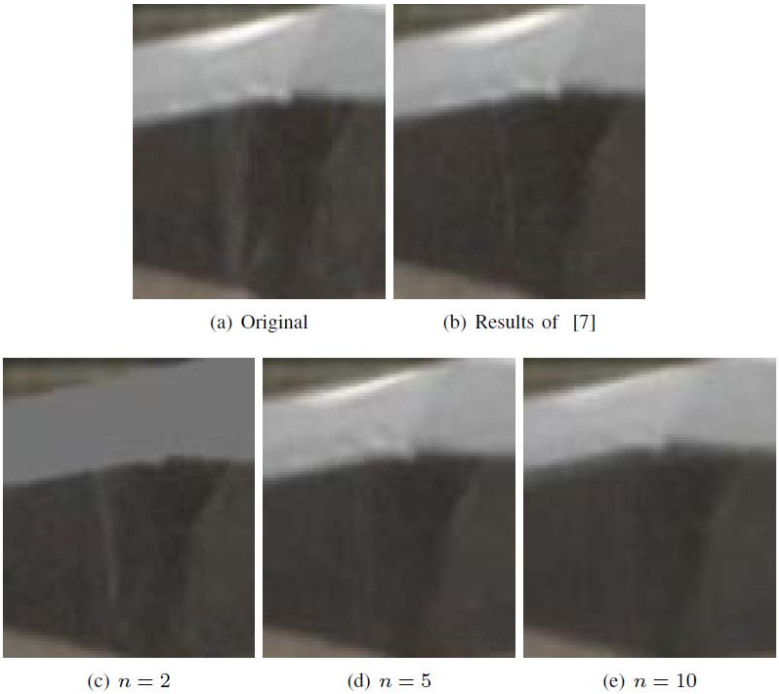

Figure 5. Qualitative results of exemplar frames of a static video [7]



(a) An original frame (b) Rain removal results [7]

(c) $n = 2$ (d) $n = 5$ (e) $n = 10$

Figure 6. Qualitative results of exemplar frames of a dynamic video [7]



(a) Original (b) Results of [7]

(c) $n = 2$ (d) $n = 5$ (e) $n = 10$

Figure 7. Qualitative results of exemplar frames of a dynamic video [7]