Natural Image and Video Decomposition with Applications to Single Image Denoising

Deepthi A J#1, Arun Pradeep*2

Dept of Electronics and Communication Engineering
Axis College of Engineering and Technology, Thrissur, India

***Abstract* — *Rain, fog and Gaussian noise removal from an image and video is a challenging problem and has been recently investigated extensively. In this paper, present a self-learning based image and video decomposition framework. Based on sparse representation, this method first learns an over-complete dictionary from the high spatial frequency parts of the input image for reconstruction purposes. An unsupervised clustering on the observed dictionary atoms has been performed in this work. And their corresponding reconstructed image versions via, affinity propagation, which allows to identify image-dependent components with similar context information. While applying this method for the applications of image and video denoising, it was able to automatically determine the undesirable patterns like rain streaks, Fog or Gaussian noise. From the derived image components directly from the input image or video, so that the task of single-image denoising can be addressed. DWT is used for better performance. Fog will degrade the quality of the preview image by reducing the saturation and contrast.. The objective of this method is to enhance the visibility, saturation, contrast and reduce the noise in a foggy image. Here single frame is used for enhancing foggy images using multi-level transmission map. The method is fast and free from noise. Comparison with the existing method shows that this method provides better processing time and quality. Experiments were conducted based on: single-image and video denoising with Gaussian, Fog and rain noises. The empirical results confirm the effectiveness and robustness of this approach, which is proved to outperform state-of-the-art image denoising algorithms.*

Keywords — *Denoising, rain removal, fog removal, image decomposition, Natural image, DWT*

I. Introduction

An image is a linear mixture of multiple source components and the image decomposition aims at determining such components and the associated weights [1]. For example, how to properly divide an image into texture and non-texture parts has been investigated in the applications of image compression, image inpainting, or related image analysis and synthesis tasks. Then 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 N × C unknown variables. While this problem is posed, image sparsity prior has been exploited to address this task [1]. As a result, an input image can be morphologically decomposed into the different patches based on such priors for a variety of image processing applications [6]. First briefly review the morphological component analysis (MCA) [6], which is a sparse representation based image decomposition algorithm [9], and has been successfully applied and extended to solve the problems of image denoising, image inpainting , and image deraining (i.e., rain removal) [8], [12]. Discrete wavelet transform is used to get a better performance. The benefit of DWT is that it will also extract the high frequency parts from the diagonal section. As a result more accurate image is obtained. If the noisy video is given as the input, the video is divided in to frames and then the process will continued [27].

Fog is one of the noise used in this method[18]. Sometimes it will affect the outdoor scenes. Reducing the effect of fog can be used by several steps. They are, Dark channel extraction [20], Airlight calculation, Transmission map and Radiance map. The noisy video and image are treated in one section and self-learning is carried out there. If the noisy input is a fog image, then the defogging is done on another way. Atlast the denoised output is obtained. Different types of filters, especially bilateral filter, KSVD, BM3D and SURELET are used in this method [7], [2]. The denoised noises are Fog, Gaussian and rain removal. Noise from the video and image can be denoised.

The scope of the project is to denoise the input noisy image and provide an output with maximum clarity. Noises like fog, Gaussian, and also rain will affect the clarity of the image. This method will helps to remove the noises by different filters like Bilateral, KSVD, BM3D and SURELET and display the denoised output [2]. Noise like Gaussian, fog and rain streaks are also removed from video. To remove Gaussian noise. It is important to add Gaussian noise to the noise free image and video [27]. That is Gaussian noise is added manually.

The scope of the paper is to denoise the input noisy image and provide an output with maximum clarity [23]. Noises like fog, Gaussian, and also rain will affect the clarity of the image. This method will helps to remove the noises by different filters like

Bilateral, KSVD, BM3D and SURELET and display the denoised output. Noise like Gaussian, fog and rain streaks are also removed from video [27]. To remove Gaussian noise. It is important to add Gaussian noise to the noise free image and video. That is Gaussian noise is added manually. The objectives of this paper is to identify image-dependent components with similar context information, does not need to collect training image data in advance, identifying image components which corresponds to undesired noise patterns, to minimize the processing period, to enhance the visibility and saturation of the fog affected image or video, to reduce the noise in a foggy image and video and to process the denoising in both noisy image and video [8].

A. *Image Decomposition by MCA*

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 [6]. Consider an image I of N pixels is a superposition of K components, called morphological components, denoted by

$$\mathrm{I} = \sum_{k=1}^{K} I_k \qquad (1)$$

Where I_k denotes the k-th component. ie, the geometric or textural component of the image I. To decompose I in to I_k, k=1,2,….. K, MCA iteratively minimizes the given energy equation

$$E(\{I_k\}_{k=1}^{K}, \{\theta_k\}_{k=1}^{K}) = \frac{1}{2}\|\mathrm{I} - \sum_{k=1}^{K} I_k\|_2^2 + \tau \sum_{k=1}^{K} E_k(I_k, \theta_k) \qquad (2)$$

Where θ_k denotes the sparse coefficients corresponding to I_k with respect to the dictionary D_k, τ is a regularization parameter and E_k is the energy function defined according to the type of D_k - global or local dictionary.

The MCA algorithms solve (2) by iteratively performing for each component I_k, the following two steps. (i) Update of the sparse coefficients : this step performs sparse coding sparse coding to solve θ_k or $\{\theta_k^p\}_p^P$, Where θ_k^p represents the sparse coefficints of the p-th patch y_k^p extracted from I_k and p is the total number of extracted patches, to minimize $E_k(I_k, \theta_k)$ while fixing I_k and (ii) Update of the components : this step updates I_k or $\{y_k^p\}_{p=1}^P$ while fixing θ_k or $\{\theta_k^p\}_{p=1}^P$.

B. *Overview and Contribution of Self-Learning*

Identifies the image components based on semantical similarity and thus can be easily applied to the applications of image denoising [14]. Unlike prior learning-based image decomposition or denoising works which requires the collection of training image data, eg : raw/noisy inputs Vs denoised outputs, or low-resolution Vs high-resolution output images. Here advocates the self-learning of the input noisy input image directly. After observing dictionary atoms with high spatial frequency, ie, potential noisy patterns, here advanced the unsupervised clustering algorithm of affinity propagation without any prior knowledge of the number of clusters, which allows to automatically identifies the dictionary atoms which corresponds to undesirable noise patterns. As a result, removing such noise from the input image can be achieved by performing image reconstruction without using the associated dictionary atoms. It is clear that this method doesn’t need any external training image or video data (eg : noisy and ground truth image pairs), and no user interaction or prior knowledge is needed either. Therefore, this method can be considered as an unsupervised approach. And according to the experiments, this method can be directly applied to a single input image or video and solve single-image or video problems of rain streaks, Gaussian and fog noise removal [11], [16]. The former type of noise can be considered as structured noise patterns and the latter as the unstructured ones.

Major contribution of the method is tri-fold : (i) Besides prior MCA based approaches, this method allows to decompose an input image or video and to observe its representation without the need to learn from pre-collected training data. Here does not assume any image priors such as the relationship between the input and desirable output images either. This makes the single-image and video based applications applicable in real-world scenarios. (ii) Here advance affinity propagation for identifying key image and video components which exhibit similar context information, so that those associated with noise or undesirable patterns can be discarded for automatic image and video denoising [27]. (iii) While this method can be applied to address the task of single-image or video denoising and rain and fog removal, further show that here do not limit the use of any specific preprocessing techniques when retrieving the high spatial frequency parts, eg : Bilateral filtering, K-SVD-based image and video denoising and BM3D filtering [2], [7].

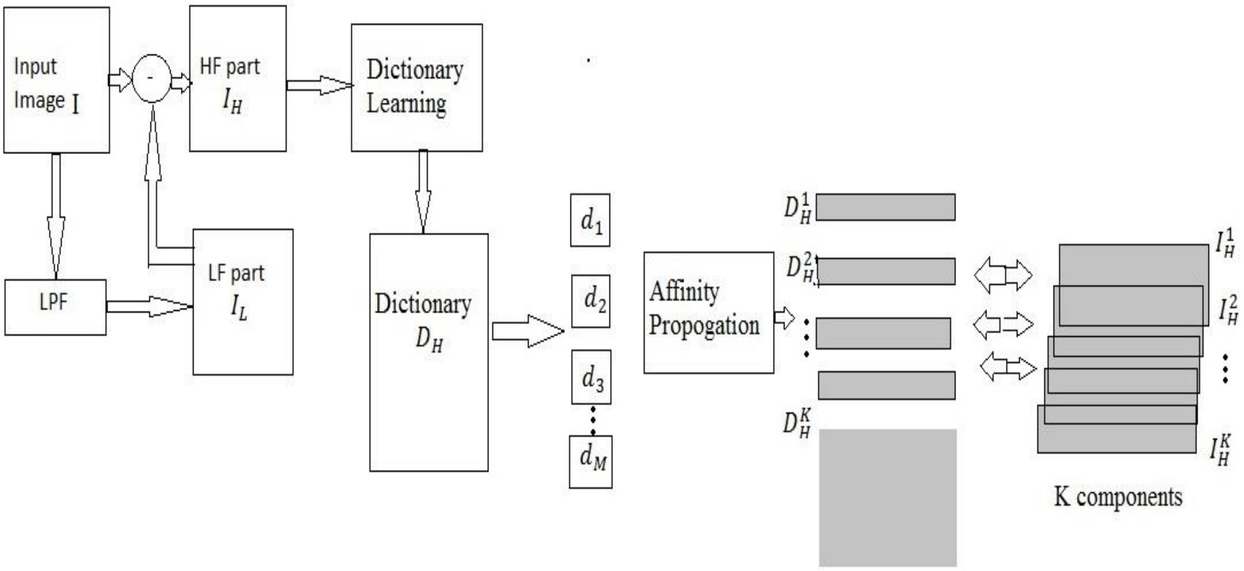


Fig 1: Illustration of image decomposition frame work

II. Sparse Representation And Dictionary Learning

A. Sparse Representation

Sparse coding is a technique of representing a signal in terms of a compact linear combination of a set of basis signals or atoms from a dictionary [3], [9]. A pioneering work in image sparse representation stated that the receptive fields of simple cells in mammalian primary visual cortex can be characterized as being spatially localized, oriented, and band passed. It was shown that a coding strategy that maximizes sparsity is sufficient to account for the above properties, and that a learning algorithm attempting to determine sparse linear codes for natural scenes will develop a complete family of localized, oriented, band passed and receptive fields.

For each image patch y^P extracted from an image I. The corresponding sparse coefficient vector θ^P with respect to a given dictionary D can be find by solving the following optimization problem.

$$\arg \min_{\theta^P} \left(\frac{1}{2}\|y^P - D\theta^P\|_2^2 + \lambda\|\theta^P\|_1\right) \qquad (3)$$

Where λ is regularization parameter. It has been shown that (3) can be efficiently solved using the orthogonal matching pursuit (OMP) algorithm.

B. Dictionary Learning

To construct a dictionary D to sparsely represent each patch extracted from an input image or video, one can use a set of training image patches y^P, p=1,2…..P, for learning purposes. To derive a dictionary D which satisfies above sparse coding scheme, the following optimization problem can be solved.

$$\min_{D,\theta^P} \sum_{p=1}^{P} \left(\frac{1}{2}\|y^P - D\theta^P\|_2^2 + \lambda\|\theta^P\|_1\right) \qquad (4)$$

Where θ^P denotes the sparse coefficient vector of y^P with respect to D and λ is a regularization parameter. Eqn (4) can be efficiently solved by performing a dictionary learning algorithm, such as online dictionary learning or K-SVD algorithm [2].

III. Proposed System

In the detailed block diagram Fig 2 given above, It is clear that the Gaussian, rain noises from the image and video are denoised in one section with the use of self-learning [16]. DWT section is attached to it, to denoise the noise also from the diagonal part. Defogging is done on another section using dark channel extraction and mask [20].

A. Dictionary Learning for Image Sparse Representation

First separate the high spatial frequency parts I_H from the low spatial frequency parts I_L for an input image I. This is because most undesirable noise patterns like rain streaks or Gaussian noise are of this type [12], [13], [15]. If the input is a noisy video, then it is divided into frames and the same process is done here. In order to achieve I= I_L + I_H, consider the use of three low-pass filtering (LPF) or denoising techniques : bilateral, KSVD and BM3D as the

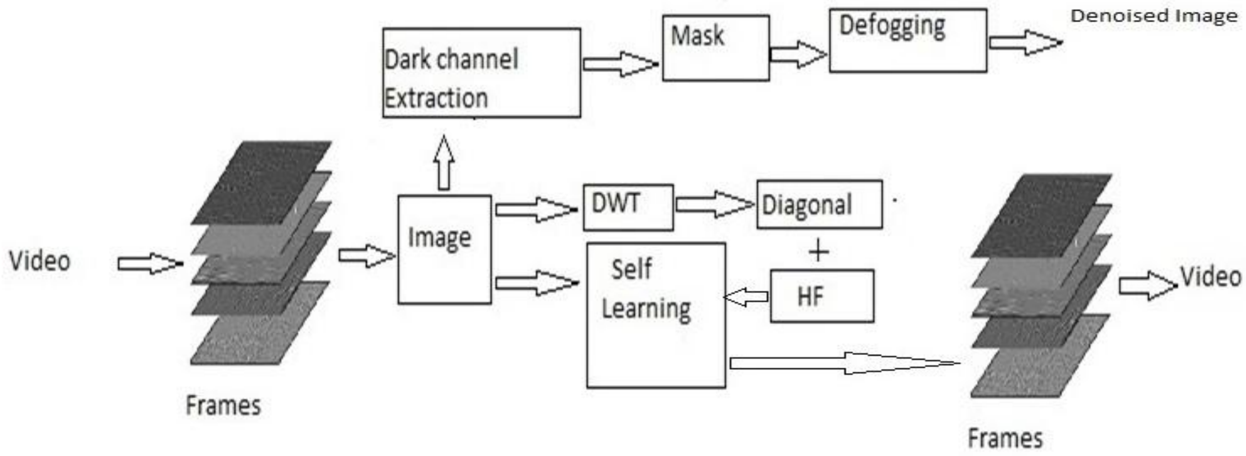


Fig 2: Block diagram of self-learning based image and video reconstruction

preprocessing stage [7], [2]. These denoising techniques can be replaced by band-pass filtering, only if the noise of interest is known to be associated with a particular frequency band [8]. Nevertheless, I_H can be produced by subtracting the resulting smoothed/filtered version I_L from I. However, since do not have prior knowledge or assumptions on the type of noise to be removed, it is not clear how to identify the image components of I_H which correspond to undesirable noise patterns.

MCA has been successfully applied to decompose an image in to different components/atoms [6]. However, traditional MCA approaches usually use a fixed dictionary, eg : discrete cosine transform (DCT), wavelet, or curvelet basis) to sparsely represent an image component. For these cases, the selection of dictionaries and parameters become heavily empirical [3], [9], and the results will be sensitive to the choice of dictionaries. While some advanced training image data to learn dictionaries for improved representation, how to select a proper image set in advance for training remains a challenging problem. Moreover, the collection of training data might not be practical in many real-world applications such as single-image based processing tasks.

Therefore, different from the traditional MCA using fixed dictionaries, here the learning of dictionary is directly from the input image. More precisely, here only learn a dictionary based on the high spatial frequency part of the input image, ie, I_H. When such a dictionary is observed, the next task is to automatically identify the undesirable components/patterns which correspond to noise, so that one can perform image reconstruction without using such components for achieving image denoising.

The patches y_H^P(p=1,2,… P) are extracted of size n × n from I_H for learning the dictionary D_H via solving (3). Here applying an online dictionary algorithm for solving the following problem:

$$\min_{D_H,\theta_H^P} \sum_{p=1}^{P} \left(\frac{1}{2} \|y_H^P - D_H\theta_H^P\|_2^2 + \lambda\|\theta_H^P\|_1 \right) \tag{5}$$

Where θ_H^P denotes the sparse coefficient vector of y_H^P with respect to D_H and λ is a regularization parameter. The learned dictionary D_H = [d1…….., d_M] contains M atoms and thus is of size n^2 × M (Having M > n^2).

B. Learning Of Context Aware Image Components

The atoms d_i of D_H are not necessarily distinct from each other in such a over-complete dictionary for sparse image representation. Therefore, it is not easy to estimate the undesirable image patterns in I_H using the observed dictionary atoms. Inspired by MCA, separated these atoms in to disjoint groups. ie, those within the same group are semantically similar to each other [5]. Thus, it will be possible to determine the group and their components associated with the noise of interest, and the task of image denoising can be achieved by performing image reconstruction without using those undesirable components.

This task is approached as solving an unsupervised clustering problem. Grouped the aforementioned M atoms d_m , m=1,2,…. M in to K different clusters, so that the atoms within the same group will share similar edge or texture information. Since the number of clusters K is not known, apply affinity propagation for solving this task. This minimizes the net similarity (NS) between atoms:

$$\text{NS} = \sum_{i=1}^{M}\sum_{j=1}^{M} c_{ij}\,\text{s}(d_{i,}d_{j}) - \gamma\sum_{i=1}^{M}(1-c_{ii})(\sum_{j=1}^{M} c_{ij}) - \gamma\sum_{i=1}^{M}|(\sum_{j=1}^{M} c\,ij)-1| \quad (6)$$

In (6), the function s(d_i, d_j) measures the similarity between atoms d_i and d_j . In order to group atoms share similar edge or texture information [21]. The similarity function is defined as

$$\text{s}(d_{i,}d_{j}) = \exp(-\|HOG(d_i)\text{-HOG}(d_j)\|^2) \quad (7)$$

Where HOG(.) extracts the features of Histogram of Oriented Gradients describing the shape/texture information of the atom [10]. The coefficient c_{ij} = 1 indicates that the atom d_i is the exemplar, that is, the cluster representative of the atom d_j , and thus d_j is categorized to cluster i. And c_{ii} equals 1 since d_i itself is the exemplar cluster i. The first term in eqn (6) is to calculate the similarity between atoms with each cluster, while the second term penalizes the case when atoms are assigned to an empty cluster (that is, c_{ii} = 0 but with $\sum_{j=1}^{M} c_{ij}$ >= 1). The third term in eqn (6), penalizes the condition when atoms belong to more than one cluster or no cluster label is assigned. Practically, the parameter γ is set to + ∞ to avoid the aforementioned problems. In addition to HOG, other features describing shape or textural information can also be considered here. The use of HOG features is sufficient to identifying and removing a dominant undesirable noise pattern from the input [10].

After grouping automatically the extracted M dictionary atoms in to K different image clusters, associated with each cluster image components I_H^k can be derived. That is, the P-th patch of I_H^k is computed from $D_H\,\delta_k$ (θ_H^P) where δ_k (θ_H^P) is a vector whose nonzero entries are only those associated with the atoms in the k-th cluster. Each image component I_H^k can considered as being associated to a particular type of context information, as depicted in figure 1. This completes the task of image decomposition [5].

C. *Single Image Rain Removal*

First decompose the input rain image I in to I_L and I_H using existing low-pass filtering techniques. If the input is a rain video [27], then divide it in to different frames and then apply filtering process. DWT is used to get diagonal and vertical frequencies of the noisy image and this result is added with high frequency to improve the final result. Once I_H is obtained, learn the dictionary D_H for representation purposes, and the dictionary atoms d_i will be grouped in to different clusters based on its HOG features via affinity propagation [11]. This clustering stage is for identifying dictionary atoms which are similar to each other in terms of their context information. Once this stage is complete, then obtain the multiple subsets of dictionary atoms D_H^k , where k=1,….,K. Each D_H^k contains dictionary atoms d with similar HOG features. As illustrated in figure 4.1, the image component I_H^k can be reconstructed using the corresponding dictionary set D_H^k.

For the task of image rain removal, from the observed K groups one of the images I_H^k would indicate the high spatial frequency rain streak pattern [26], [13]. To identify such patterns, considering the variance of gradients for each dictionary atoms associated with each group, that is, calculate the variance of HOG features of d_i^k in D_H^k. If the noise patterns of interest are the rain streaks, the edge directions of the rain streaks would be consistent throughout the patches in I_H and thus dominates one of resulting cluster k [4]. In this case, the variance of the atoms in that cluster would be the smallest among those across different clusters, and thus determine the cluster and its components corresponding to such noise patterns accordingly. When the components associated with noise are identified and removed, Use the remaining atoms for reconstructing the high frequency part of the image. Adding the low spatial frequency parts I_L back to this recovered output, the denoised version of I is produced.

D. *Application to Single Image Denoising*

1) Image Denoising: The goal of image denoising is to remove unstructured or structured noise from an image which is required in the presence of an additive noise. Numerous approaches have been proposed to address this problem. Extended from image denoising, algorithms have also been proposed for addressing particular image processing tasks. An example is bilateral filtering, which performs image denoising via Gaussian blur while being able preserves the edge information.

The use of sparse and redundant representations has been successfully applied to address this task. With a predetermined dictionary or the one learned from the input image itself, one can effectively recover the denoised version. A representative sparse-representation based denoising work is the K-SVD approach [2]. Another popular method is

BM3D (block-matching and 3D filtering), which is also on the image sparse representation in the transformed domain. Like K-SVD, BM3D also requires the prior knowledge of the standard deviation of the Gaussian noise.

2) Method for Removing Gaussian Noise: Besides rain removal, this method is also used for removing Gaussian noise from input images. Unlike K-SVD or BM3D, the standard deviation of noise patterns is not need in advance, which makes this method more practical for real-world applications.

Like rain removal, first decompose the input I in to I_L and I_H.Once $I_H$is obtained, learn the dictionary D_H and extract the HOG features for each atom d_i [10]. While HOG is not expected to describe the Gaussian noise, the presence of such noise would result in HOG features in which each bin/attribute is not distinguishable. For noise free dictionary atoms, observe dominant attributes in their HOG features. As a result, the use of HOG still allows us to perform clustering of dictionary atoms. In other words, even the standard deviation of the Gaussian noise is not given, it is still able to identify the image component which corresponds to the presence of such noise using this decomposition and clustering framework. Once this noise component is identified and disregarded, the image can be reconstructed using the remaining HF components and I_L.

E. Fog Noise Removal

Sometimes outdoor scenes are often affected by fog. A user may like to capture a clearer picture by reducing the effect of fog especially when the density of fog is too high. In the case of driving, dense fog reduces the visibility and may cause road accidents and also affects flight take-offs and landing [22].

Poor visibility due to fog is caused by the suspended particles in the atmosphere. The incoming light from scene or object is scattered due to these particles and attenuated till it reaches the camera. Therefore both saturation and contrast of the captured image reduces significantly. There are a number of methods, which do not use any physical model of degradation and in many cases produce over saturated outputs.

An enhanced image refinement technique is used in this method, which is based on a dark channel prior and airlight calculation [19], [20]. Multi-level transmission maps are used. The major contributions of fog noise removal method are :

Computaion of transmission map with multiple block sizes to avoid soft matting and hence reducing processing complexity, Computation of transmission map based on saturation map is for better compensation of saturation, Modified method to compute dark channel prior and Scheme of streaming fog, rain and snow free images

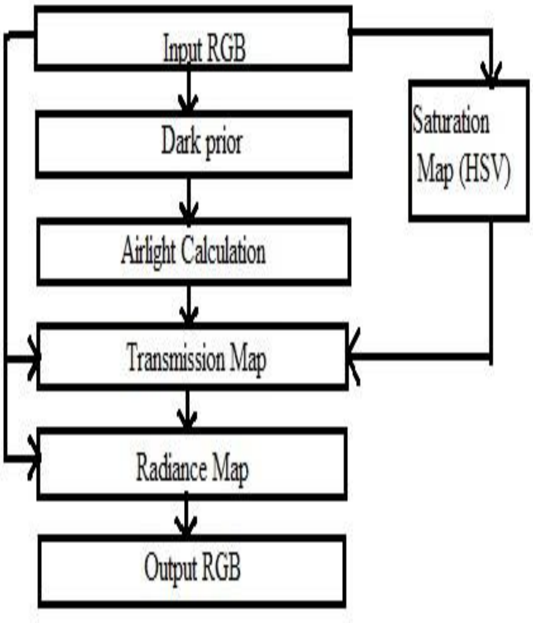


Fig 3: Block Diagram of Defogging

A foggy image can be represented as

$$I(x) = J(x)\, t(x) + A(1-t(x)) \qquad (8)$$

Where I is the intensity of the pixel, J is the original scene radiance, A is global airlight, and t is medium of transmission describing the portion of light that is not scattered and reaches the camera. ‘A’ is the Airlight, which is the result of scattering of light from scene or object [25]. A typical fog removal method involves estimating A, to find the original scene radiance. The block diagram of this method is shown in the figure below. The method computes dark channel prior and saturation map, to find transmission map for these different block sizes.

1) Calculating Dark Channel Prior and Airlight: Dark channel means the at least one color channel has some pixels whose intensity values are very low and close to zero [20]. In the current work it is computed as

$$J_d = \min(patch) \qquad (9)$$

Where Patch is a 7 × 7 matrix belongs to input image.

In this method, top 1% pixels are considering in computing airlight and find the pixel which has maximum value of J_d in its dark channel among the pixels based on equation (9) The value of I in that pixel is considered as airlight for this method [11].

2) Calculating Transmission Map and Radiance Map: The saturation of color in a foggy image decreases with the density of fog which in turn depends on depth or distance of the object. In this method saturation map and airlight is combined to get a more accurate transmission map and clearer image. The transmission map is calculated as

$$\text{Txn map} = 1 - \text{omega(dark channel)} \qquad (10)$$

Where omega is the chance to occur dark channel and the value of omega is

$$\text{omega} = 0.8 - (0.2 * S) \qquad (11)$$

Where S is saturation of the pixel (0 <= S <= 1). Saturation values of the pixels to omega are mapped to enhance the output and observed lesser saturation in object that are far away and have more fog. If saturation is less, omega will be more and vice-versa. This helps to remove inconsistent color patches in the output.

Estimate scene radiance as,

$$J(x) = \frac{I(x) - A}{\max(t(x), t0)} + A \qquad (12)$$

Where t0 is a factor to restrict the noise level

IV. Experiments

To evaluate the performance of this method, here conduct experiments for addressing two single-image and video denoising tasks: rain removal and denoising (with Gaussian noise) [12]. Considered the patch size of each image as 16 × 16 pixels, and the number of dictionary atoms M = 1024. The regularization parameter and the maximum sparsity value for the OMP algorithm are set as 0.15 and 10. For LPF preprocessing techniques, the spatial and intensity- domain standard deviations for bilateral filtering as 6 and 0.2, respectively [7]. All images are of size 256 × 256 pixels in this experiment.

A. *Performance Evaluation on Single Image Rain Removal*

Collect several synthetic rain images from the Internet and thus the ground-truth images without rain streaks presented for PSNR calculation [26]. To evaluate the performance of this method for rain removal, compared this method with bilateral filtering, K-SVD, and BM3D denoising algorithms [2], [7]. Set large standard deviation values $\sigma = 25$ and 35 for K-SVD and BM3D algorithms, respectively. During the preprocessing stage of this work, larger values are allows to remove high spatial frequency patterns including possible rain streaks from the low spatial frequency parts of the input image [26], [16]. Do not (and it is not possible) fine tune such parameters for removing the rain streaks only.

To better visualize and to compare the results, experiment conducted on grayscale rain images [12], [16]. Figure given below shows the 3 different rainy input images.

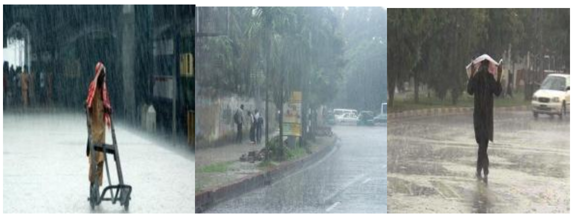

(a) (b) (c)

Fig 3: Rain Removal Input Figures

It is worth noting that, although the prior MCA-based approach successfully discarded most rain streaks without significantly degrading image quality [6], parts of non-rain components were also removed due to the heuristic dictionary partition by K-means clustering algorithm. While the prior context-based method produced comparable rain removal results, it requires one to perform context-constrained image segmentation on input images, and thus significantly increases the computational costs.

In particular, consider the image frames of the video data [27]. The videos were captured in real rainy scenes with static backgrounds, and adjust camera parameters for removing or enhancing the presence of rain streaks. Thus, using this video data, able to collect real-world rainy images and the corresponding ground truth versions.

Disctere Wavelet Transform is used for better performance. There are mainly four parts in DWT. One-low frequency and three-high frequency. They are mentioned as Diagonal, Horizontal and Vertical. Diagonal high frequency is used here and it is added with the high frequency parts of the input image for denoising.

B. *B. Performance Evaluation on Image Denoising*

To evaluate the performance of this approach for image denoising (with Gaussian noise), here collect and conduct experiments on several images [23], [14]. Manually add Gaussian noise with $\sigma = 25$ to the input noise-free images for addressing this task. Note that if the σ for the Gaussian function is known in advance then both K-SVD and BM3D algorithms will be expected to achieve excellent denoising results. However, this exact parameter choice is not known (which is practical), and here simply set large standard deviation values $\sigma = 25$ for both algorithms. Similar to the scenarios for rain removal [4], [26], this would allow to remove high spatial frequency patterns including possible Gaussian noise from the low spatial frequency parts of the input image without fine tuning the parameter. Also compare this algorithm with denoising methods not requiring the prior knowledge on for the Gaussian noise. Considered the SURE-LET algorithm, 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.

Furthermore, although the SURE-LET based approach was able to outperform the approaches using K-SVD for Gaussian noise removal [2], BM3D-based approaches still achieved the best denoising performance.

For the experimental purpose, collect several pictures and then add Gaussian noise manually in to the collected pictures. The performance is evaluated after denoising [14]. Same process is applied to denoising Gaussian noise from video [27]. Firstly Gaussian noise is added to a noise free video and then it is used as noised input. Video is divided into frames and the denoising process is done on each frame [23]. These frames are saved in to a distinct folder for evaluation purpose. Experiment conducted on 3 figures given below.

(a) (b) (c)

Fig 5 :- Input images for Gaussian noise removal

For denoising fog noise, dark channel extraction and mask processes are done [20]. The output images were enhanced significantly in terms of fog removal and saturation [24]. As the generation of transmission map is based on the saturation component, the degradation of the color saturation of image is appropriately compensated at most of the places. The advantage of this method is that, it is capable of working with multiple frames of preview or video as use temporal filtering instead of soft matting [27]. Then, the method is capable of enhancing the preview in presence of snow or rain as well. During the rain or snow fall, the distant objects become hazy and rain or snow droplets appear at different positions in different frames [25], temporal filtering helps reducing the visibility of such droplets in the output frame. Haze in the higher depth regions is reduced in the same way as removing fog in the present work [19].

Fog affected image is given as input image for defogging process [22]. After successfully performing this operation, fog affected video is given as input. For experimentation, given below figures are analysed.

(a) (b) (c)

Fig 6 :- Foggy input images

It is worth noting that, while this method quantitatively and qualitatively outperformed others, do not need to fine-tune this approach with or assume such parameters are known in advance (which might not be practical). From the above experiments, again confirm the effectiveness and robustness of this approach for image denoising [14], [23], which can be integrated with existing LPF/denoising techniques in the LPF preprocessing stage. In other words, do not limit the use of this proposed framework to any particular LPF or denoising algorithm.

Although real-time processing is not of concern of this method, provide the remarks on computation time for different learning stages of this proposed framework as follows. In this proposed method, it takes about 100 seconds to perform denoising for an input image of 256 x 256 pixels [14]. 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, 30 seconds 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 4G RAM.

V. Results

Table I lists the PSNR values of different filtering methods bilateral, KSVD and BM3D over three different rain images [2], [7], [17]. Because of the real time images, the noise free ground truth is not available. Therefore the PSNR is calculated by comparing result with the noisy ground truth. Because of this, the PSNR value will be as shown below.

Table I
PSNR Values Of Different Rain Images

Input Images	Bilateral	KSVD	BM3D	SURELET
Fig (a)	9.1735	9.1742	9.1741	9.1738
Fig (b)	9.7196	9.7188	9.7171	9.7221
Fig (c)	10.2928	10.2929	10.2927	10.2929

Similarly, Table II lists the PSNR values of different filtering methods over three different Gaussian images.

TABLE III
PSNR VALUES OF DIFFERENT GAUSSIAN IMAGES

Input Images	Bilateral	KSVD	BM3D	SURELET
Fig (a)	10.7236	10.7234	10.7241	10.7242
Fig (b)	8.1367	8.1349	8.1340	8.1324
Fig (c)	11.6446	11.6454	11.6454	11.6448

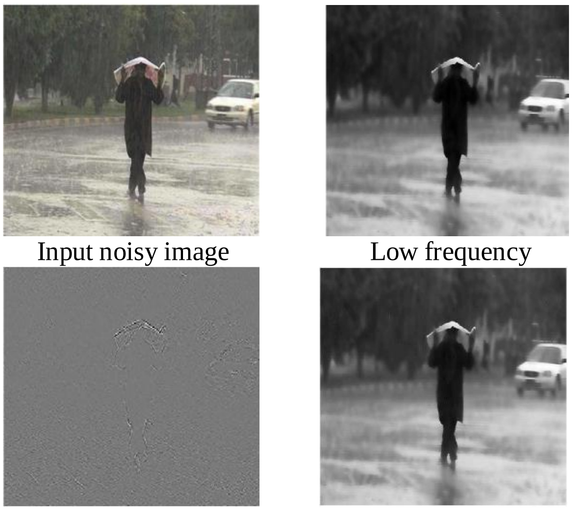
Table III lists the PSNR values of different filtering methods over three different Fog images. Here also the noise free ground truth is not available, and because of that the PSNR is calculated by comparing the noisy free output with the noisy input.

TABLE IIIII
PSNR VALUES OF DIFFERENT FOG IMAGES

Input Images	Bilateral	KSVD	BM3D	SURELET
Fig (a)	5.1556	5.1556	5.1556	5.1556
Fig (b)	2.8239	2.8239	2.8239	2.8239
Fig (c)	3.6352	3.6352	3.6352	3.6352

A. *Results of Denoising Rain Streaks from Image*

The figure given below shows the comparison of grayscale rain images denoised by different types of filters, Bilateral, KSVD, BM3D and SURELET [2], [7] [17].



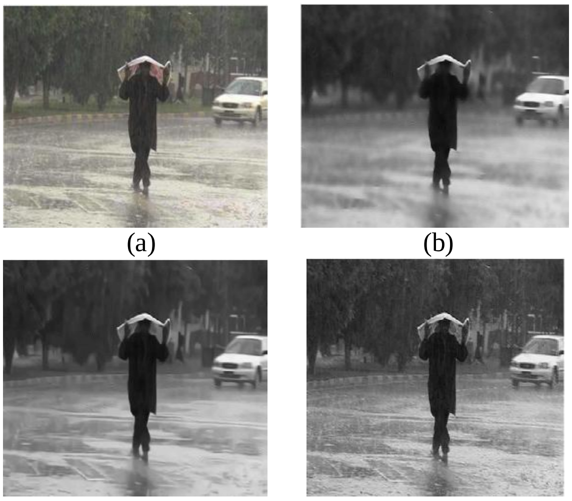
Input noisy image

Low frequency

High frequency

Rain removed output

Fig 6 :-Denoising result images obtained by using Bilateral filter



(a)

(b)

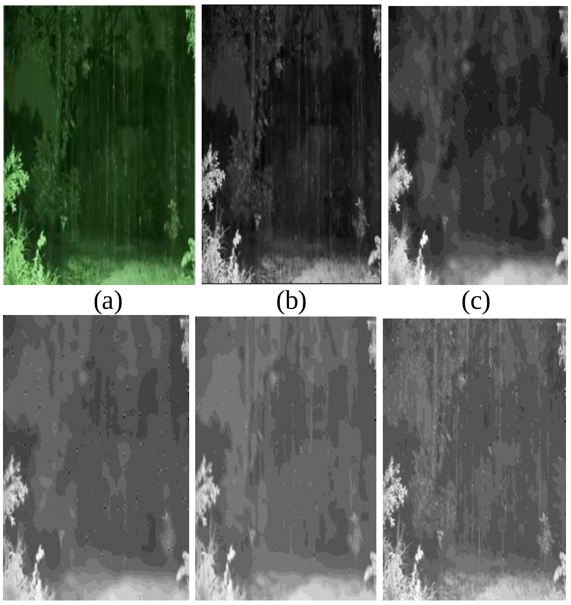
(c)

(d)

Fig 7 :- (a) Input noisy image and the Denoising result images obtained by using KSVD, BM3D and SURELET filter (b), (c), (d) respectively

From these figures, it can be observe that Bilateral, K-SVD, and BM3D methods were able to remove most rain streaks [16], [26], these denoising techniques inevitably disregarded image details (e.g., high spatial frequency parts). While applying these techniques in LPF preprocessing stage, then able to successfully identify/recover most non-rain image details and thus achieved improved visual quality.

B. *Results of Denoising Rain Streaks from Video*



(a)

(b)

(c)

(d)

(e)

(f)

Fig 8 :- (a) Ground truth frame (b) original frame and the denoised result frames obtained by using Bilateral, KSVD, BM3D and SURELET filter (c), (d), (e) and (f) respectively

From these results, Bilateral, KSVD and BM3D filters eliminate most of the noises from the input noisy image [17], [7], [2]. The noisy input image is taken as the ground truth image and the corresponding PSNR values based on different filters are calsulated. The comparison is made out using these values.

The table given below shows the PSNR values corresponding to each filters.

TABLE IVV
PERFORMANCE COMPARISONS(IN TERMS OF PSNR) OF DIFFERENT IMAGE DENOISING APPROACHES OF RAIN VIDEO

	Bilateral	KSVD	BM3D	SURELET
Rain video	11.4572 11.4576 11.4790 11.4786	11.4572 11.4576 11.4790 11.4786	11.4572 11.4576 11.4790 11.4786	11.4572 11.4576 11.4790 11.4786

The above table describes the PSNR values of rain video when denoised using different denoising filters like bilateral, KSVD, BM3D and SURELET [17], [7], [2]. The real time video is used here. Therefore the PSNR values are calculated by comparing the denoised output figure with the noisy input frame.

C. Results of Denoising Gaussian noise from Image

The given below figures shows the Gaussian noise input image and the noise eliminated output image by using bilateral, KSVD, BM3D and SURELET filters [2], [7], [17] and [27].

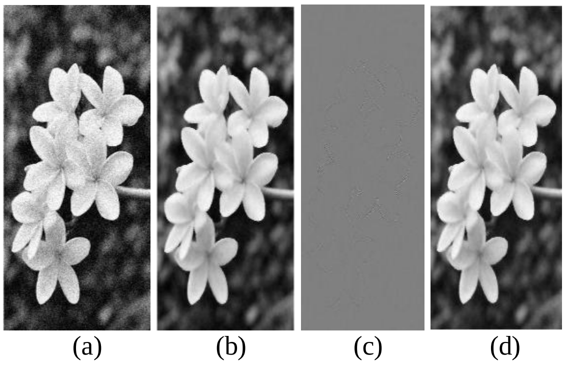


Fig 9 :- (a) Input noisy image and the Denoising result images obtained by using bilateral filter (b) low frequency, (c) high frequency, (d) rain removed output respectively

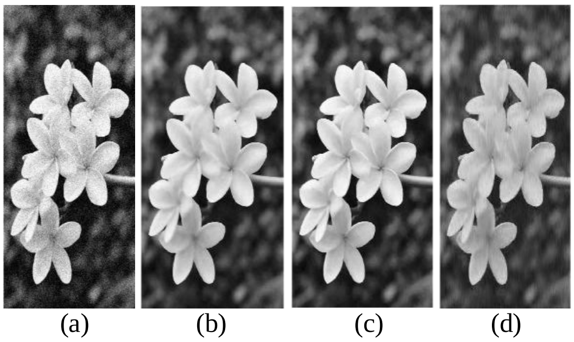


Fig 10 :- (a) Input noisy image and the Denoising result images obtained by using KSVD, BM3D and SURELET filter (b), (c), (d) respectively

D. Results of Denoising Gaussian noise from video

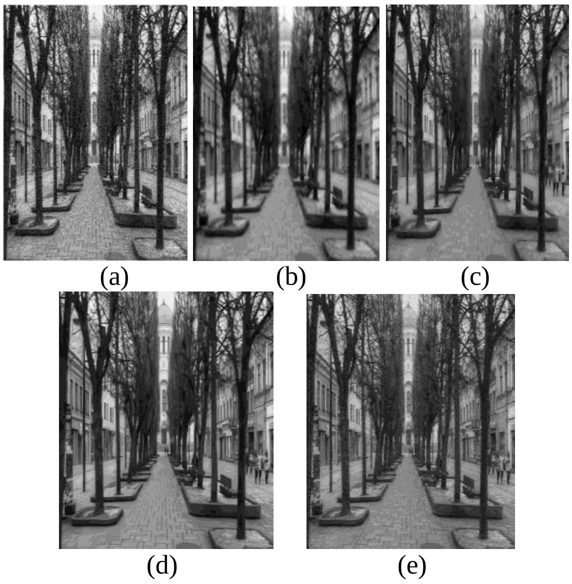


Fig 11 :- (a) Ground truth frame and the denoised result frames obtained by using Bilateral, KSVD, BM3D and SURELET filter (b), (c), (d) and (e) respectively

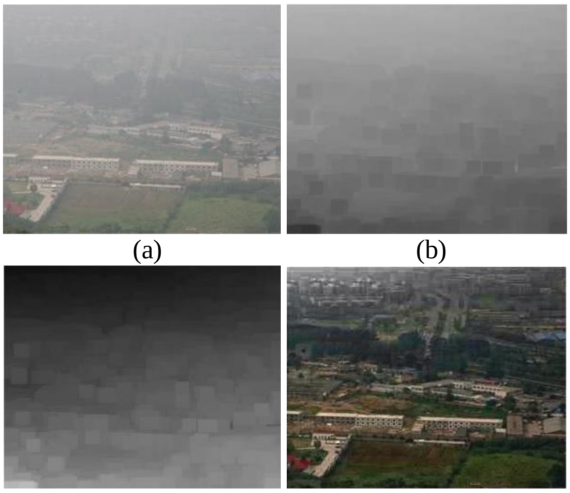
The table given below shows the PSNR values corresponding to each filters.

TABLE V
PERFORMANCE COMPARISONS(IN TERMS OF PSNR) OF DIFFERENT IMAGE DENOISING APPROACHES OF GAUSSIAN VIDEO

	Bilateral	KSVD	BM3D	SURELET
Gaussian Video	6.7130 6.6951 6.7051 6.7275	6.7130 6.6951 6.7051 6.7275	6.7130 6.6951 6.7051 6.7275	6.7130 6.6951 6.7051 6.7275

Gaussian noise is successfully removed from the video using bilateral, KSVD, BM3D and SURELET filters and it is clear from the above figures [27].

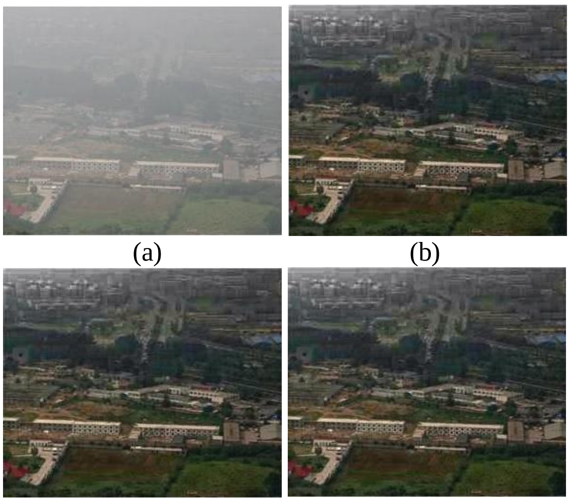
E. Results of Denoising Fog noise from Image



(a) (b)

(c) (d)

Fig 12 :- Defogging images obtained by bilateral filtering (a) foggy input image (b) Dark channel (c) Transmission map (d) Defogged output

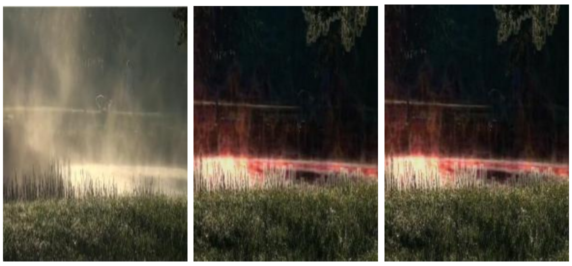


(a) (b)

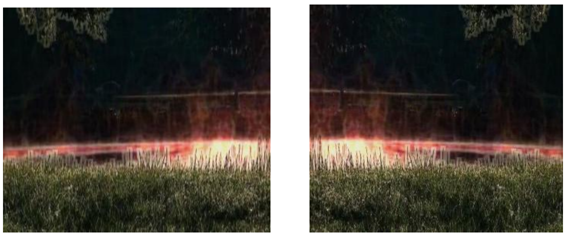
(c) (d)

Fig 13 :- (a) foggy input image and Defogging images obtained by using different filters (b) KSVD(c) BM3D and (d) SURELET

F. Results of Denoising Fog noise from Video



(a) (b) (c)



(d) (e)

Fig 11 :- (a) Ground truth foggy noise frame and the denoised result frames obtained by using Bilateral, KSVD, BM3D and SURELET filter (b), (c), (d) and (e) respectively

The above frames showing that the defogging is successfully done to the foggy input video using bilateral, KSVD, BM3D and SURELET filters [26], [11], [19].

The table given below shows the PSNR values corresponding to each filters.

TABLE VI
PERFORMANCE COMPARISONS(IN TERMS OF PSNR) OF DIFFERENT IMAGE DENOISING APPROACHES OF FOG VIDEO

	Blateral	KSVD	BM3D	SURELET
Fog video	6.2548 6.2316 6.2479 6.2465	6.2548 6.2316 6.2479 6.2465	6.2548 6.2316 6.2479 6.2465	6.2548 6.2316 6.2479 6.2465

VI. CONCLUSION

This paper presented a self-learning-based image decomposition work for single image and video denoising. This 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 the images nor the collection of training image data. To address the task of image denoising, this method is able to identify image components which correspond to undesired noise patterns. DWT is used here for better performance.

An enhanced fog removal technique is attached. The method is effective in fog removal so as to produce a better output as compared to existing fog removal techniques. Saturation of the image is enhanced without adding any noise or blocking artifacts in the image. This method uses transmission maps based on multiple blocks, which helps to remove some of the noise generated due to fog. Though this method is faster as compared to other existing techniques, real-time fog removal is still a challenge for HD or better quality preview frames.

Experiments on two types of single image denoising tasks (with structured and unstructured noise) confirmed the use of this method, which was shown to quantitatively and qualitatively outperform existing denoising approaches. The scope for future works are to denoise more noises in the images like salt and pepper, to improve the current processing time and also to add contrast enhancement to the defogging process.

REFERENCES

[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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," inProc. IEEE Int. Conf. Comput.Vis., Bombay, India, Jan. 1998, pp. 839–846.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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, Jan. 2010.

[14] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," Multiscale Model. Simulat., vol. 4, no. 2, pp. 490–530, 2005.

[15] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 4, pp. 791–804, Apr. 2012.

[16] 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.

[17] T.Blu and F. Luisier, "The sure-let approach to image denoising," IEEE Trans. Image Process., vol. 16, no. 11, pp. 2778–2786, Nov. 2007.

[18] J. Sahu, "Design a New Methodology for Removing Fog from the Image", International Journal of Advanced Computer Research, Vol.12 Number-4 Issue-7, pp. 62-65, 2012.

[19] R. Fattal, "Single image dehazing", SIGGRAPH, 2008.

[20] K. He, J. Sun, & X.Tan, "Single image haze removal using dark channel prior", IEEE Transaction on pattern Analysis and Machine Intelligence, Vol. 33, No. 12, 2011.

[21] R.T.Tan, "Visibility in bad weather from a single image", IEEE Conference on CVPR, 2008.

[22] Z.Tao , S. Changyan1 & W.Xinnian1, "Atmospheric scattering-based multiple images fog removal", 4th International Congress on Image and Signal Processing, 2011.

[23] A. Buades, Y. Lou, J. M. Morel, & Z. Tang, "Multi image noise estimation and denoising", (hal.archives-ouvertes.fr),2010.

[24] J. Mao, U. Phommasak, S. Watanabe, H. Shioya, "Detecting Foggy Images and Estimating the Haze Degree Factor", J Comput Sci Syst Biol, Vol, 7, pp.226-228,2014.

[25] V. Senthamilarasu, A. Baskaran, K. Kutty, "A new approach for removing haze from images," Processing of the International Conference on Image Processing, Computer Vision and Pattern Recognition (IPCV), 2014.

[26] Peter C, Barnum, Srinivasa Narasimhan, " Analysis of rain and snow in frequency space," Int J Comput Vis DOI 10.1007/s11263-008-0200-2

[27] K Garg, S K Nayar, "Detection and removal of rain from videos", Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004