Exploring Video Denoising using Matrix Completion

by

Hima Bindu Maguluri

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2013 by the
Graduate Supervisory Committee:

Baoxin Li, Chair
Pavan Turaga
Claude Claveau

ARIZONA STATE UNIVERSITY
August 2013
ABSTRACT

Video denoising has been an important task in many multimedia and computer vision applications. Recent developments in the matrix completion theory and emergence of new numerical methods which can efficiently solve the matrix completion problem have paved the way for exploration of new techniques for some classical image processing tasks. Recent literature shows that many computer vision and image processing problems can be solved by using the matrix completion theory. This thesis explores the application of matrix completion in video denoising. A state-of-the-art video denoising algorithm in which the denoising task is modeled as a matrix completion problem is chosen for detailed study. The contribution of this thesis lies in both providing extensive analysis to bridge the gap in existing literature on matrix completion framework for video denoising and also in proposing some novel techniques to improve the performance of the chosen denoising algorithm. The chosen algorithm is implemented for thorough analysis. Experiments and discussions are presented to enable better understanding of the problem. Instability shown by the algorithm at some parameter values in a particular case of low levels of pure Gaussian noise is identified. Artifacts introduced in such cases are analyzed.

A novel way of grouping structurally-relevant patches is proposed to improve the algorithm. Experiments show that this technique is useful, especially in videos containing high amounts of motion. Based on the observation that matrix completion is not suitable for denoising patches containing relatively low amount of image details, a framework is designed to separate patches corresponding to low structured regions from a noisy image. Experiments are conducted by not subjecting such patches to matrix completion,
instead denoising such patches in a different way. The resulting improvement in performance suggests that denoising low structured patches does not require a complex method like matrix completion and in fact it is counter-productive to subject such patches to matrix completion. These results also indicate the inherent limitation of matrix completion to deal with cases in which noise dominates the structural properties of an image. A novel method for introducing priorities to the ranked patches in matrix completion is also presented. Results showed that this method yields improved performance in general. It is observed that the artifacts in presence of low levels of pure Gaussian noise appear differently after introducing priorities to the patches and the artifacts occur at a wider range of parameter values. Results and discussion suggesting future ways to explore this problem are also presented.
To my parents and brother.
ACKNOWLEDGEMENTS

My deepest gratitude goes to my advisor, Prof. Baoxin Li for his unending support and guidance throughout my study at ASU. I learnt from him, how to approach, formulate and present research. His constant support has given me the chance to build my skills which will benefit me in my entire career. I am thankful for the opportunity of working with a person like Dr.Li.

My sincere thanks to the committee members, Prof. Pavan Turaga and Dr.David Claveau for spending their valuable time in reviewing my work.

I thank Parag Chandakkar for being there for me in good and bad times. I am very thankful to Ragav Venkatesan for his help in the completion of this draft. I thank Dr. Peng Zhang for all the motivating and interesting conversations. I thank Qiongjie Tian and Archana Paladugu for being very helpful persons as teammates. I thank all the members of my group for the interesting discussions. I thank my friends here and back in India who stood by me throughout this journey.

Most importantly, I thank my parents for their love and support which made me who I am today. I thank my brother for he has been my strength throughout my life. He had played every possible role to help me throughout my career. This would not have been possible without the support from my family.
<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.2 Discussion on speed performance.</td>
<td>61</td>
</tr>
<tr>
<td>5 CONCLUSIONS AND FUTURE WORK</td>
<td>66</td>
</tr>
<tr>
<td>5.1 Future work</td>
<td>67</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>72</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>PSNR values of denoised frames for CIF videos using the baseline approach</td>
<td>30</td>
</tr>
<tr>
<td>3.1</td>
<td>PSNR values of denoised frames for CIF videos by grouping structurally relevant patches in presence of motion</td>
<td>43</td>
</tr>
<tr>
<td>3.2</td>
<td>PSNR values of denoised frames for CIF videos by separating low-structured patches</td>
<td>44</td>
</tr>
<tr>
<td>4.1</td>
<td>PSNR values of denoised frames for CIF videos by prioritizing patches for MC</td>
<td>49</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Examples of Original, Noisy and Adaptive median filtered frames. (b) &amp; (c) show noisy and partial denoised images for dominant impulsive noise. (d) &amp; (e) correspond to dominant Poisson noise.</td>
<td>28</td>
</tr>
<tr>
<td>2.2 Example of matched patches from 20 neighboring frames at noise level ((\sigma = 10, \eta = 0.1, s = 0.2)).</td>
<td>29</td>
</tr>
<tr>
<td>2.3 Original, Noisy and denoised frames from Tempete video at noise level ((\sigma = 10, \eta = 0.2, s = 0.1)).</td>
<td>29</td>
</tr>
<tr>
<td>2.4 Original, Noisy and denoised frames from Mother &amp; Daughter video at noise level ((\sigma = 10, \eta = 0.2, s = 0.1)).</td>
<td>30</td>
</tr>
<tr>
<td>3.1 Matched patches using baseline approach in presence of high motion.</td>
<td>32</td>
</tr>
<tr>
<td>3.2 Matched patches over 20 frames by selecting fixed number of patches from each frame.</td>
<td>33</td>
</tr>
<tr>
<td>3.3 Matched patches using our approach in presence of high motion.</td>
<td>34</td>
</tr>
<tr>
<td>3.4 Patch matching on partially denoised data.</td>
<td>35</td>
</tr>
<tr>
<td>3.5 Visual quality of denoised images at same noise level ((\sigma = 10, \eta = 0.1, s = 0.2)) with different structural content.</td>
<td>36</td>
</tr>
<tr>
<td>3.6 Visual quality of denoised regions with high and low structural content.</td>
<td>36</td>
</tr>
<tr>
<td>3.7 Histogram of (\sigma_c) values in presence of pure Gaussian noise of standard deviation 30.</td>
<td>38</td>
</tr>
<tr>
<td>3.8 Histogram of (\sigma_c) values in presence of mixed noise obtained from the noisy data.</td>
<td>39</td>
</tr>
<tr>
<td>3.9 Histogram of (\sigma_c) values in presence of mixed noise obtained from the adaptive median filtered data.</td>
<td>41</td>
</tr>
<tr>
<td>3.10 Preserving structure by grouping relevant patches.</td>
<td>42</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>4.1 Comparison of all the four methods for Akiyo video.</td>
<td>50</td>
</tr>
<tr>
<td>4.2 Comparison of all the four methods for Bowing video.</td>
<td>51</td>
</tr>
<tr>
<td>4.3 Comparison of all the four methods for Bus video.</td>
<td>52</td>
</tr>
<tr>
<td>4.4 Comparison of all the four methods for Football video.</td>
<td>53</td>
</tr>
<tr>
<td>4.5 Comparison of all the four methods for Foreman video.</td>
<td>54</td>
</tr>
<tr>
<td>4.6 Comparison of all the four methods for Mobile video.</td>
<td>55</td>
</tr>
<tr>
<td>4.7 Comparison of all the four methods for Mother &amp; Daughter video.</td>
<td>56</td>
</tr>
<tr>
<td>4.8 Comparison of all the four methods for Tempete video.</td>
<td>57</td>
</tr>
<tr>
<td>4.9 Consistency of PSNR over 100 frames.</td>
<td>58</td>
</tr>
<tr>
<td>4.10 % of data constituted in regions away from the mean for a Gaussian distribution.</td>
<td>59</td>
</tr>
<tr>
<td>4.11 Variation in % of retained pixels with $\alpha$ for a set of 7 videos in presence of low and high levels of pure Gaussian noise.</td>
<td>62</td>
</tr>
<tr>
<td>4.12 Artifacts introduced by baseline method for $\alpha = 0.7$ in presence of Gaussian noise with $\sigma = 10$ on QCIF videos.</td>
<td>63</td>
</tr>
<tr>
<td>4.13 Artifacts introduced by method 3 of proposed approach in presence of Gaussian noise with $\sigma = 10$.</td>
<td>63</td>
</tr>
<tr>
<td>4.14 Visual comparison of reconstructed images.</td>
<td>65</td>
</tr>
<tr>
<td>5.1 Closeness plots for randomly chosen reconstructed patch matrix #1 with Adaptive median filtered output and ground truth.</td>
<td>69</td>
</tr>
<tr>
<td>5.2 Closeness plots for randomly chosen reconstructed patch matrix #2 with Adaptive median filtered output and ground truth.</td>
<td>70</td>
</tr>
<tr>
<td>5.3 Singular values plots of two randomly chosen patch matrices.</td>
<td>70</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

With the advancements made in computer vision and image understanding technologies, images/videos collected from a variety of sources have become an important part of real world data. These images/videos act as input for computer vision systems which are used in many fields ranging from common day to day applications such as image search, object/event detection, video tracking to more sophisticated applications such as defense, medicine, autonomous vehicles etc. The input to such systems comes from a variety of sources as high/low cost digital cameras, scanners, mobile phones, webcams, screen recorders etc. The quality of the input depends upon many factors such as capturing technology, lighting conditions, compression artifacts, transmission errors, motion of the object etc. The performance of many computer vision systems degrades in the presence of low quality input. Therefore, image or video restoration is a key task in any computer vision system. Denoising is one of the important tasks in image or video restoration as images collected under non-ideal conditions are easily prone to noise and many applications involving such images require their denoised versions for optimal performance.

1.1 Image noise and noise models

Noise in images arises due to a variety of reasons and based on the statistical properties of each type of noise, they are modeled using several probability distributions. Commonly occurring noise types and models are described in this section.

Pixels in an image are affected by overheated or faulty pixels in camera sensors, errors in digitization process and bit errors in data transmission. Noise
caused due to any of the above reasons will affect some number of pixels in the image as opposed to corrupting each pixel value in the image. In such noise, the signal strength of noise is much larger than the image and thus affected pixels tend to take extreme values from the allowed range of image intensity. So generally, affected pixels appear very different from their surroundings. In a gray scale image, affected pixels appear as randomly occurring white and black dots and thus this kind of noise is named as salt and pepper noise. This noise is modeled as impulse distribution typically with spikes at maximum and minimum values of the image. Basic low pass filters such as mean filter or Gaussian smoothing fail on impulsive noise because the corrupted pixel affects the mean significantly and thus mean can deviate by large amounts from the original value. Median filtering is effective in removing impulsive noise and preserving edges.

Noise occurs in images due to electronic circuits (such as amplifiers in cameras), sensors (for example, CCD read noise). Such noise is modeled as additive Gaussian noise, characterized by its mean and standard deviation. Gaussian distributions are the most frequently used noise models because of their mathematical tractability in spatial and frequency domains [1]. Another reason for the importance of Gaussian distributions is the central limit theorem which states that mean of large number of independent and identically distributed (i.i.d.) random variables will approximately follow normal distribution.

The variation in number of photons sensed at a given exposure level causes photon shot noise. This noise is caused due to random arrival of photons. The probability of arrival of a photon in a given time period follows Poisson distribution and thus shot noise follows a Poisson distribution. For
sufficiently large number of samples, this distribution can be approximated by a Gaussian distribution. This noise has its root mean square value proportional to square root of the image intensity. This noise is more dominant when collecting relatively less number of photons and thus it is dominant in lighter parts of the image. It can be reduced by increasing exposure time or by combining multiple frames.

There is an additional shot noise which is caused by dark leakage current in image sensors. This noise, called as dark-current shot noise can be removed by dark-frame subtraction.

The quantization noise caused by quantizing image pixels into discrete levels is approximated by additive signal independent uniform distribution. This is an assumption and in reality, quantization noise is related to the signal. The assumption is valid in some cases like when other noise sources cause dithering or one way to ensure the validity of the assumption is to apply dithering which helps in randomizing the quantization error and thus making it signal independent.

In addition to above mentioned noise sources, there can be other types of noise which can be modeled by different distributions. For example, Rayleigh probability distribution is used to model noise occurring in range imaging, exponential and gamma distributions are used in laser imaging.

In [2], various noise sources that corrupt pixel values have been quantified by studying the properties of a CCD camera system.

1.2 Denoising

Denoising aims at effectively removing noise (acquired during capturing and transmission) from the corrupted image and recovering the original clean im-
The performance of any denoising algorithm depends on its effectiveness in differentiating noise from original image data. The better its differentiating capability, the better would be the performance of the algorithm. In general, no algorithm can make this decision 100% accurately. So, there is always a trade-off between amount of noise removed and fine image detail that is preserved. Naive approaches for denoising or algorithms designed for aggressive noise removal generally smooth out fine details and blur the edges. Sophisticated algorithms try to study the noise characteristics and image models/structural details to make more accurate classification between noise and image detail. Thus, such algorithms effectively eliminate noise by preserving fine details. Understanding of human visual system also assists in effective denoising. A straight-forward example in which knowledge of image model and human vision is used for denoising is separation of chroma and luminance noise. Some algorithms separate the image into chroma and luminance components and perform more noise reduction on the chroma component due to the following two reasons: most of the image detail is concentrated in the luminance component and there is not much of important detail present in the chroma component. So, there is lower loss of information by performing more noise reduction on chroma component; generally, people find chroma noise more objectionable than luminance noise. Some of the fundamental techniques for image denoising are linear and nonlinear filters in spatial domain, anisotropic filtering [3] [4], frequency domain filtering etc.

1.3 Related work

There has been tremendous amount of research in image denoising and many impressive algorithms have been proposed over the past 30 years. Nevertheless, there is substantial focus on this task in present-day research aiming at
continually improving the state-of-the-art. In [5], Chatterjee et al. studied the performance bounds for the denoising problem. This work estimated a lower bound on the performance in terms of mean squared error (MSE) and compared the results from state-of-the-art methods with this bound. All the state-of-the-art methods compared in this work are patch-based methods, validating the fact that recently proposed effective denoising approaches are all patch-based [6], [7], [8] and [9]. In [5], performance limits are calculated by assuming the availability of noise-free image and deriving the bounds on how well the given image can be denoised. The approach in [8] is used to cluster geometrically similar patches together and the bias of the estimator for each cluster is approximately modeled as an affine function (estimator refers to the denoising approaches and each denoising method can be characterized by the parameters of the affine function). Experiments are presented to show that affine model is a reasonable model for the bias function. Experiments also show that the affine model is valid only when all the patches are geometrically similar. So the bias is modeled as different affine functions for different clusters. Based on this assumption, the bound on the conditional covariance of the biased estimate is formulated as Optimal Bias Bayesian Cramér-Rao Lower Bound (OB-CRLB) which is derived from the Bayesian version of the classical CRLB [10]. Then the bound on the MSE is derived by solving for the optimal parameters of the affine bias model. This framework developed in [5] can be used to derive bounds for any noise distribution, though only additive white Gaussian noise is discussed in [5]. The derived expression for the lower bound on MSE depends upon the covariance of the clean image patches. Therefore, bootstrapping is used to estimate the stochastic lower bound on MSE and associated confidence interval. In any general image, the above steps are repeated for each cluster containing geometrically similar
patches and the final MSE bound is obtained as an aggregate of bounds for each cluster. By using this framework, MSE bounds have been derived for several images and compared with performance of state-of-the-art algorithms on these images. This work in [5] concludes that though the performance of the state-of-the-art algorithms is impressive, there still exists some scope for improvement in some cases depending on content of images, noise levels etc.

Though denoising relatively smooth and simple images is easier than denoising high textured images, there is still scope for improvement in denoising relatively smooth images. Even though denoising high textured images is harder, the performance of the state-of-the-art is quite close to the predicted bounds (especially in images with repetitive patterns) and there is very little scope for improvement. In high textured images, some improvement is still possible at lower noise levels where many denoising methods show bias and artifacts.

Moreover, most of the existing denoising approaches assume the noise model is Gaussian. Such methods show drastic reduction in performance when applied on real world images which are generally corrupted by mixed noise from various sources [2]. So, in general there is still need for improvement in existing techniques to be effectively used in practical applications and thus the denoising problem draws attention of present-day researchers. In this section, some of the fundamental and influential methods proposed for image and video denoising are discussed.

In the initial stages, images were treated as two dimensional signals and many of the denoising approaches were derived from the field of signal processing. In much later years, computer vision techniques for denoising based on statistical models and learning theory were developed. One of the early techniques which served as basis for many denoising approaches is filtering
in frequency domain. A notable example of such methods is Discrete cosine transform (DCT) based noise filtering [11]. Then came the recursive filtering techniques based upon Kalman filtering [12] and Bayesian estimation such as [13] and [14]. In 1980, Lee noticed that though the existing techniques are impressive, they are computationally expensive and not suitable for real-time processing [15]. While the frequency domain transformations on large two dimensional arrays are computationally expensive, the recursive methods are not suitable for parallel processing. In [15], a spatial, non-recursive noise filtering technique based on local image statistics was introduced. The filtering algorithm is developed based on the local mean and variance at each pixel and then applying the minimum mean square error (MMSE) estimator. In this algorithm, each pixel can be processed independently making it suitable for parallel processing and thus real-time processing.

The emergence of wavelet transform [16] played a prominent role in solving image processing problems such as denoising, image compression etc. In [16], it has been shown that the difference of information between successive resolutions of an image can be extracted by decomposing the image using a wavelet orthonormal basis. It has been shown that wavelet transform facilitates in depth understanding of the statistical properties of images and applications of wavelets in image compression and texture discrimination have been discussed. In [17], Daubechies presented wavelet theory and different types of wavelet transforms in great detail. Donoho et al. proposed denoising by soft or hard thresholding on the coefficients in wavelet domain [18], [19]. These wavelet thresholding methods were later improved to achieve translation-invariance [20]. Though the method proposed in [16] and similar methods based on orthonormal basis are very useful in image coding appli-
cations, their basis functions are not steerable (rotation invariant) and thus such methods do not favor orientation analysis \cite{21}. In \cite{21}, steerable filters have been designed which form a basis set and for steerable image transform. Though the efficiency decreases (compared to orthonormal basis such as \cite{16}) as the filters are nonorthogonal and overcomplete, these steerable oriented filters are useful in many image processing tasks edge detection, enhancement, orientation analysis etc. One of the application discussed in \cite{21} shows that noise removal and enhancement of oriented structures can be done efficiently using this method. In 1996, Simoncelli et al. developed a Bayesian estimator that can incorporate the non-Gaussian higher-order statistics present in the subband decompositions of natural images \cite{22}. The Wiener solution for denoising uses second-order statistics from the Fourier decomposition. This estimator presented in \cite{22} can be viewed as an extension of Wiener solution which can also make use of higher-order statistics of subband representations which capture additional image information. In \cite{22}, the steerable pyramid transformation derived from \cite{21} has been used to generate the wavelet subbands. Thereafter, many image denoising algorithms have been developed based on the wavelet representation \cite{23} and \cite{24}. In 2003, Portilla et al. proposed a denoising method based on Bayesian least squares-Gaussian scale mixtures (BLS-GSM) estimation on subbands generated by using steerable pyramid transform.

Variational image processing is another important framework which provides solutions for many image processing problems such as image denoising, deblurring, segmentation etc. \cite{3}. Rudin et al. in 1992 developed a total variation based algorithm for denoising \cite{25}, which is also a partial differential equation (PDE) based nonlinear diffusion filter. A constrained optimization
problem is solved to minimize the total variation of an image and the constraints are obtained from the noise statistics. This method played a prominent role in image denoising and there after many methods were developed for image restoration based upon total variation minimization [26], [27]. The total variation method preserves edges but shows the “staircase effect”. Undesirable artifacts are often introduced by second order PDEs due to the transformation of smooth regions into piecewise constant regions. To eliminate this problem, many improved nonlinear filters have been proposed [28]. Many nonlinear methods based upon higher order PDEs were proposed to reduce the staircase effect [29], [30], [31]. Other alternative variational methods proposed to reduce the piecewise constant behavior of TV methods include the mean curvature models [32], [33]. An introduction to variational image processing along with discussions on the most recent literature on application of variational models to several image processing problems including image denoising can be found in [34].

In 2005, Buades et al. made a systematic comparison of the then existing primary denoising techniques by defining the method noise [35]. The selected techniques were compared using four different criteria to allow comparison of various aspects of the denoising methods. As there are several attributes (for ex: amount of noise removed, fine image detail preserved, artifacts introduced) to be considered in evaluating the performance of denoising algorithms, defining a single criterion will not provide a fair comparison. Therefore, four ways of comparison have been employed in [35] including both objective and subjective ways of comparison. The authors also introduced a new denoising algorithm called non-local means (NL-means) which explores the redundancy in natural images to denoise the images as well as preserve
the structures and detail [35] and [36]. This algorithm is based on the following idea: In predicting the value of a particular pixel \( i \), the center pixels of all other patches that are similar to patch centered at \( i \) are considered. This idea was for the first time used in texture synthesis in [37] which later became very useful in applications like image restoration, restoration of missing regions in 3D scenes etc. In case of NL-means algorithm, the denoised value at pixel \( i \) is calculated as the weighted average of the center pixels of all other patches that are similar to patch centered at \( i \). The weights are based on the similarity of a patch with the patch centered at \( i \).

Recently, many novel denoising algorithms have been proposed based on the patch-based non-local framework [6], [7], [8], [38] and [39]. Some of these recent methods show exceptionally high performance outperforming the previous methods. In [6], grouping gives 3D arrays of similar patches which are termed as “groups”. Each group is collectively filtered by the proposed collaborative filtering which explores similarity between grouped patches. Collaborative filtering is achieved by: 3D linear transformation of the groups, shrinkage (thresholding or Wiener filtering to remove noise) in the transform domain and inverse linear transformation. This procedure named as block-matching and 3-D filtering (BM3D) has outperformed the existing denoising methods. [7] is another patch-based non-local algorithm which gives excellent denoising results. The authors have used the K-SVD algorithm [40] to learn a dictionary which allows sparse representation of the image patches. As the learned dictionary can only represent small fixed size patches, the authors have proposed a global prior to deal with large arbitrary sized images. In the Bayesian reconstruction framework, the global prior has been defined in such a way that it forces sparsity over all the local image patches. [38] is another
recently proposed remarkable algorithm in which the ideas of non-local means and sparse coding are combined to overcome the flaws of each of these two methods. In this approach, similar patches are forced to have similar sparse decomposition by using simultaneous sparse coding [41]. This joint framework results in effective denoising.

Denoising of videos is favored when compared to image denoising due to the presence of temporal redundancy. Generally, adjacent frames in videos are similar to each other. This similarity provides much more information when denoising a frame compared to the case where each frame in the video is treated as a single disjoint image and denoised separately. For example considering the non local framework for video denoising case, each patch in a frame $k$ will now have many more similar patches which include patches from frame $k$ and from adjacent frames. This implies much more information exists which can be exploited to effectively remove noise while preserving video content. This is the general case and there may exist extreme cases where there is abrupt change between successive frames due to high motion in a video. Even in such a case, exploring temporal redundancy is advantageous because some patches in a frame will still have similar patches in adjacent frames. Imagining a worst case where two successive frames in a video happen to be two images which are completely independent of each other, this method reduces to single image denoising case for the particular frames. So, considering the denoising performance on over all video, exploring temporal similarity always gives better performance. Owing to this fact, in general all video denoising algorithms attempt to make effective use of spatial and temporal similarity. All the patch-based image denoising techniques discussed above can be extended to video denoising by searching for similar patches across several frames. Though each
of them adopted different ways to accurately find similar patches in presence of noise and motion, most of the above discussed methods have been extended to video denoising.

The NL-means algorithm \cite{35,36} has been extended to denoise image sequences in \cite{42}. The authors have shown that motion compensation is not necessary for image sequence denoising and it may also be disadvantageous as it discards useful information. Method presented in \cite{42} denoises image sequences yet preserving the main detail. Comparison presented in terms of “method noise” showed that this algorithm performs better than motion compensated methods like \cite{43,44}. The BM3D method \cite{6} has been extended to video denoising in \cite{15}. This method termed as V-BM3D groups similar patches in video by predictive-search block-matching. The authors propose predictive-search block-matching to search for similar patches in a data-adaptive spatio-temporal 3D subdomain. This method achieves state-of-the-art video denoising performance. The K-SVD based image denoising \cite{7} has been extended to image sequence denoising in \cite{46}. In \cite{47}, similar patches from images taken from multiple views are grouped by using depth estimation. Grouped patches are denoised using PCA and tensor analysis. Markov random field (MRF) based image \cite{48} and video \cite{49} denoising models are also among the prominent denoising methods proposed in the recent years. Lin et al. in 2012 introduced Switchable MRFs to solve several low level vision problems including image and video denoising \cite{50}. The authors propose an approach to control the underlying graphical structure of the MRFs by introducing switching variables. Different low level vision tasks share the same underlying graph and then task specific solutions are inferred based upon the graph. A variational inference algorithm is derived which provides both graphical structure
Most of the denoising methods mentioned above assume a fixed statistical model for noise, mostly additive Gaussian noise. Such methods attempt to attenuate noise based on the assumption that the corrupting noise follows the assumed model. This assumption is not valid on real world data. For example, [2] identifies most common sources of noise when CCD cameras are used and characterizes the noise properties. In general, noise in images/videos is caused by various sources and thus it is a combination of noise falling under different statistical models. It is not possible to define a single statistical model which can characterize this mixed noise. Algorithms based on single statistical noise model will fail to deal with mixed noise. In general, any denoising algorithm cannot categorize the mixed noise present in videos into different kinds of noise (each with unique statistical model). The algorithm has to deal with noise altogether, implying that in presence of other noise it cannot even effectively remove the part of the noise which comes from the statistical model which the algorithm was designed for. This rules out the possibility of applying different existing algorithms one after the other to remove different types of noise which constitute the mixed noise. Thus, though the above mentioned algorithms produce impressive results, most of them cannot be employed for practical denoising purposes. This shows that there is a strong need for a denoising algorithm which can deal with mixed noise as a whole. Identifying this need, Ji et al. in 2010 proposed a denoising algorithm with mild assumptions on the statistical model of the noise unlike the previously discussed methods [51]. This is also a patch-based non local algorithm based on grouping and collaborative filtering with the major difference being the collaborative filter-
ing part has been formulated as a low-rank matrix completion problem. This novel formulation combined with efficient algorithm to solve low-rank matrix completion problem yields a robust denoising algorithm capable of removing mixed noise from videos. In [52], removal of structured noise is achieved by using robust motion estimation.
Chapter 2

MATRIX COMPLETION FOR VIDEO DENOISING

This chapter presents the formulation of video denoising as a low-rank matrix completion problem [51]. Our implementation and analysis of the method presented in [51] is discussed in detail in this chapter. Before presenting this discussion, an introduction to the matrix completion problem and overview of the existing related algorithms to solve this problem is given.

2.1 Matrix completion

The problem of matrix completion (MC) is gaining enormous attention in the recent years with its application in solving practical problems and many efficient theoretical algorithms being proposed to recover the complete matrix. This problem can be described as recovering missing entries when only few data samples have been observed. The solution for this problem can be applied to many practical problems from various fields. One famous example for application of matrix completion in practice is the Netflix challenge. Though it is theoretically not possible to recover all matrices from any set of observed samples, Candès et al. in 2009 showed that most of the matrices can be perfectly recovered from most of the sample sets [53]. The authors have proved that for sufficiently low rank matrices, this recovery can be done from a nearly minimal number of entries by solving a simple convex optimization problem under suitable conditions. This theoretically strong yet simple solution has inspired many theoretical works [54], [55] and [56] as well practical applications of matrix completion.

Theoretically, the low-rank matrix completion problem can be solved by an optimization problem which minimizes rank of the recovered matrix.
subjected to the constraint that entries corresponding to observed elements in
the recovered matrix are equal to the observed values. Such an optimization
problem is NP-hard and solving such problems for exact solution involves
double exponential time complexity which prevents such solutions being used
in practice [53]. In [53], the authors have used nuclear norm instead of rank
in the optimization problem. Nuclear norm is a convex function and it can be
efficiently optimized which makes the solution practically appealing. This idea
has been stated in the authors’ words as, “for most problems, the nuclear norm
relaxation is formally equivalent to the combinatorially hard rank minimization
problem”. Many methods have been proposed to efficiently optimize such a
convex optimization problem [56], [54] and [57].

The matrix completion theory is related to the concept of sparsity.
Sparsity can be described as the property of the signals which allows represen-
tation by much smaller amount of data without loss of useful information.
In low-rank matrix completion, a signal can be recovered from small amount
of data under some assumptions. Inspite of the low-rank criteria, this the-
ory is very useful in many fields because most of the natural signals exhibit
information redundancy.

In [58], even more useful result which can be stated as matrix completion
is provably accurate even when the few observed entries are corrupted with a
small amount of noise. This result is extremely remarkable and useful in real
world applications. Examples of this remarkable result employed in various
fields can be seen in [59], [60] and [61].
2.2 Video denoising using low-rank matrix completion

The work presented in this thesis is based on the formulation proposed in [51]. So, before proceeding further, the video denoising algorithm presented in [51] is described in this section.

2.2.1 Formulation of denoising as matrix completion problem

A noisy video containing \( K \) frames can be represented as \( \mathcal{F} = \{f_k\}_{k=1}^{K} \) where \( f_k \) represents each frame in the given video. Considering the noise to be additive, each frame can be viewed as sum of clean image and \( g_k \) and noise \( n_k \) where the noise can be mixed noise as discussed in Section 1.3.

\[
f_k = g_k + n_k \quad (2.1)
\]

\[
P_{j,k} = (p_{1,j,k}, p_{2,j,k}, \ldots, p_{m,j,k}) \quad (2.2)
\]

\[
P_{j,k} = Q_{j,k} + N_{j,k} \quad (2.3)
\]

The problem to be solved is the recovery of \( \mathcal{G} = \{g_k\}_{k=1}^{K} \) by removing noise \( n_k \) from all frames \( f_k \). This is a patch-based non local scheme based on the grouping and collaborative filtering framework. As explained in Section 1.3 any video denoising algorithm attempts to effectively make use of the temporal redundancy. So, for a particular frame \( f_k \), search for similar patches is done in a set of neighboring \( M \) frames, which includes \( f_k \). Let \( p_{j,k} \) represent a patch of size \( n \times n \) centered at pixel \( j \) in frame \( k \). For this particular patch, \( m \) similar patches are found from each of the chosen \( M \) neighboring frames. As the set of neighboring frames also includes \( f_k \), this search gives similar patches both in spatial and temporal domain. The resulting set of similar patches consisting of \( N = Mm \) patches is represented as \( \{p_{i,j,k}\}_{i=1}^{N} \). Representing each
patch of size $n \times n$ as a vector of size $n^2$, the set of similar patches becomes set of vectors where each vector $p_{i,j,k} \in \mathcal{R}^{n^2}$. By representing this set of vectors as a matrix with each patch vector being the columns as in Equation 2.2 the columns of the matrix $P_{j,k}$ represents all the similar patches of the patch centered at pixel $j$ in frame $k$. From [2.1], each grouped patch matrix will be sum of matrix representing similar patches grouped from clean video $(Q_{j,k})$ and matrix representing noise at all corresponding locations $(N_{j,k})$. (as in Equation 2.3)

Considering the case of clean video, similar patches can be accurately found though this cannot be perfectly done in presence of noise. As $Q_{j,k}$ represents similar patches from clean video, the grouped patches will have almost identical structural content unlike the noisy case. This means that the columns of $Q_{j,k}$ are structurally similar to each other. Such a matrix has high amount of information redundancy. Relating back to the concept of sparsity, such data can be represented by much lesser information in a transform domain without significant loss of image content. Such a matrix $Q_{j,k}$ will have low-rank and according to matrix completion theory discussed in Section [2.1] complete version of such a matrix can be recovered from few observed values even if some of the observed entries are noise corrupted. Based on this idea, the problem of collaboratively filtering $P_{j,k}$ has been formulated as a low-rank matrix completion problem which can robustly estimate $Q_{j,k}$.

The problem is to estimate low-rank noiseless matrix $Q_{j,k}$ from its observed noisy version. Since, $Q_{j,k}$ can be reliably estimated from few observed values, entries from $P_{j,k}$ which are considered noisy are discarded and only reliable pixel values are retained from which an estimate of $Q_{j,k}$ can be recovered. When large number of samples are drawn independently from a zero
mean probability distribution function (pdf), their average value tends towards zero. For a particular scalar \( s \), if large number of its noisy versions are generated by adding noise samples drawn independently from a zero mean pdf, the average of all these noisy versions will tend towards true value \( s \). For the noise less version \( Q_{j,k} \), since the matched patches are very similar to each other, all the elements in each row of \( Q_{j,k} \) will be numerically close to each other. Assuming zero mean noise, each row in \( P_{j,k} \) corresponds to sum of corresponding elements from \( Q_{j,k} \) and randomly drawn samples from zero mean pdf. Thus, average of each row of \( P_{j,k} \) should be close to the true pixel value at position \( j \) in frame \( k \). This estimate will not be an accurate estimate due to two reasons: patch matching done in presence of noise is not accurate. So, patches forming \( P_{j,k} \) are not exactly identical to each other. The second reason is the averaging is done on finite (small) number of samples. Though this is not an accurate estimate, it is reasonable enough to judge the reliability of pixel values in \( P_{j,k} \).

Based on this idea, matrix entries of \( P_{j,k} \) which deviate by large amount from its corresponding row vector are discarded. The discarded entries could be pixels corrupted by large amplitude of noise. They could also be a result of incorrect grouping as similar patches cannot be accurately matched in presence of noise. Discarding of the noisy pixels results in an incomplete version of \( P_{j,k} \) with few reliable entries. In the next step, low-rank estimate of \( Q_{j,k} \) is obtained by solving a convex optimization problem.

### 2.2.2 Mathematical formulation

The following notations are useful in the subsequent sections.

\[
\|X\|_F := \left( \sum_{i,j} |x_{i,j}|^2 \right)^{\frac{1}{2}} \tag{2.4}
\]

\[
\|X\|_*_i := \sum_{i} (\sigma_i(X)) \tag{2.5}
\]
\[ D_\tau(X) = U\Sigma_\tau V^T \quad (2.6) \]

where \( \Sigma_\tau = \text{diag}(\max(\sigma_i - \tau, 0)) \)

The Frobenious norm of a matrix \( X \) is given by equation \[2.4\] and the nuclear norm of \( X \) is defined by equation \[2.5\] where \( \sigma_i(X) \) is the \( i \)th largest singular value of \( X \). If the factorization of \( X \) by singular value decomposition (SVD) is \( X = U\Sigma V^T \), then the soft shrinkage operator \( D_\tau(X) \) is defined as in equation \[2.6\]

Let \( \Omega \) be an index set, then \( \#(\Omega) \) represents the size of the set \( \Omega \) and \( X|_\Omega \) denotes the vector \( X \) with only those elements which are indexed in \( \Omega \).

\[
\min_Q ||Q||_* \quad \text{s.t.} \quad ||Q|_\Omega - P|_\Omega||_F^2 \leq \#(\Omega)\hat{\sigma}^2 \quad (2.7)
\]

Once the reliable elements of \( P_{j,k} \) are identified as mentioned in Section \[2.2.1\], the index set \( \Omega \) is formed by including indices of all the reliable pixels. \( P_{j,k}|_\Omega \) represents incomplete version of \( P_{j,k} \) obtained by discarding noisy pixels. The task of recovering the low-rank noiseless version \( Q_{j,k} \) from its noisy and incomplete observed matrix \( P_{j,k}|_\Omega \) is achieved by solving the optimization problem in \[2.7\] which is a low-rank matrix completion problem in presence of noise.

\( \hat{\sigma} \) is the standard deviation of noise present in \( P_{j,k}|_\Omega \). (Details on calculation of \( \hat{\sigma} \) are mentioned in \[2.3.2\]).

The recovery of complete low-rank data from corresponding noisy data with missing elements can be achieved by several other principal component analysis (PCA) based methods such as [62] and [63]. In [63], the problem has been formulated as principal component analysis with missing data (PCAMD) which has been generalized as a weighted least-squares (WLS) minimization problem. The minimization approach in \[2.7\] has the advantage of strong math-
2.3 Algorithm details

The main steps involved in the video denoising algorithm [51] are discussed in detail in this section.

2.3.1 Patch matching

Patch matching over several frames is an essential task to explore temporal redundancy in videos and it is required in many applications such as motion estimation, video coding etc. Many motion estimation algorithms have been proposed for efficient and accurate search of similar patches over frames [64], [65]. For a particular patch, the motion estimation algorithms search for the best match in the reference frame while our aim is to find the top $m$ matches. Since the main focus of this work is on denoising and as efficient patch matching is itself a wide research area, we employ an exhaustive block matching algorithm (EBMA) which is optimal in accuracy [66]. The EBMA is implemented with a search range $R$. Mean absolute difference (MAD) is used as the similarity measure between patches in the EBMA algorithm i.e. the patch with lowest MAD value is considered as the best match. For each patch in a frame $f_k$, exhaustive search for similar patches is done within the search window in all the $M$ neighboring frames (which include $f_k$). Best $m$ matches are selected from each frame and arranged as columns of a matrix, thus forming the noisy patch matrix $P_{j,k}$ with $N$ columns.

Since the problem formulation in Section 2.2.1 is based on the criterion that columns of $Q_{j,k}$ are similar to each other (thus making $Q_{j,k}$ a low-rank
matrix), it is very important to group similar patches in a reliable way. This task needs careful consideration as patch matching is done in presence of noise in our algorithm. Specifically, noise types such as impulsive noise corrupt the pixels with very high magnitude as such pixels take the value of either minimum or maximum intensity. This large deviation of the pixel values from the original affects the calculated similarity measure between patches to a great extent, thus resulting in unreliable grouping of similar patches. To avoid this, patch matching is not applied directly on the noisy video data. An intermediate estimate of the video data is obtained by removing the impulsive noise. Using such partially denoised data instead of original noisy video for patch matching, improves the accuracy of patch matching and similarity between the grouped patches. An adaptive median filter [67] is used to remove the impulsive noise and generate the partially denoised video data. The adaptive median filter finds the pixels corrupted by impulsive noise and such pixels are replaced by the median value of its neighborhood pixels. The advantage of the adaptive median filter over the standard median filter is that it can even handle large probabilities of impulsive noise corruption (unlike the standard median filter). Moreover, it preserves the detail and smooths non-impulsive noise.

2.3.2 Denoising patch matrix using matrix completion

$P_{j,k}$ constituting of similar patches in spatial and temporal domain is formed from the patch matching step discussed in section 2.3.1. As discussed in section 2.2.2, incomplete version of $P_{j,k}$ is formed by retaining only the reliable pixels. The set $\Omega$ contains indices of all the reliable pixels. This set $\Omega$ is formed by discarding pixels from two stages: Firstly, the pixels identified as impulsive noise corrupted by the adaptive median filter discussed in the section 2.3.1.
Secondly, the set of pixels whose value deviates largely from the mean of the corresponding row vector. A pre-defined threshold is set on the amount of allowed deviation of a pixel value from its corresponding row vector. The indices of all the remaining pixels after discarding the pixels from above two stages form $\Omega$. Then, $P_{j,k}|\Omega$ is formed by including only the elements in $\Omega$.

$$\min_Q ||Q||,$$

s.t.  $$||Q|_\Omega - P|_\Omega||^2_F \leq #(\Omega)\hat{\sigma}^2$$

As discussed in section 2.2.2, $Q_{j,k}$ is recovered from $P_{j,k}|\Omega$ by solving the minimization problem in 2.8. $\hat{\sigma}$ is the estimate of standard deviation of noise present in $P_{j,k}|\Omega$. Variances of all elements $\in \Omega$ are calculated across each row. $\hat{\sigma}$ is obtained from $\hat{\sigma}^2$ which is calculated as the average of all such row variances.

$$\min_Q \frac{1}{2}||Q|_\Omega - P|_\Omega||^2_F + \mu||Q||,$$  

By using the standard duality theory, the Lagrangian version of 2.8 shown in 2.9 is equivalent to 2.8 for some value of $\mu$.

$$||Q|_\Omega - P|_\Omega||^2_F \approx #(\Omega)\hat{\sigma}^2$$

The parameter $\mu$ in the unconstrained formulation 2.9 should be chosen such that the solution of 2.9 satisfies 2.10.

$$\mu = (\sqrt{n_1} + \sqrt{n_2})\sqrt{p}\hat{\sigma}$$

From the heuristics presented in 58, $\mu$ is chosen as shown in 2.11. $n1$ and $n2$ are the dimensions of the patch matrix i.e. $n1 = n^2$ and $n2 = N$. $p$ is the fraction of pixels retained in $P|_\Omega$ (ratio of the number of entries in $\Omega$ to the total number of pixels in the patch matrix).

$$p = #(\Omega)/(n1 \times n2)$$
Solution of 2.9 gives the denoised patch matrix. 2.9 can be efficiently solved by using many existing algorithms. The fixed point iterative algorithm ([56] and [54]) is used in this work owing to its implementation simplicity. Algorithm 1 shows the implementation details for solving 2.9 using the fixed point iterative method.

**Algorithm 1** Fixed point iteration for solving the minimization (2.9)

\[
Q^{(0)} \leftarrow 0 \\
\textbf{While } ||Q^{(k)} - Q^{(k-1)}||_F \leq \epsilon, \text{ iterate on } k \\
\quad \left\{ \begin{array}{l}
R^{(k)} = Q^{(k)} - \tau \mathcal{P}_\Omega(Q^{(k)} - P), \\
Q^{k+1} = D_{\tau \mu}(R^{(k)}),
\end{array} \right. \quad (2.13a) \tag{2.13a} \\
\text{end While}
\]

where \( \mu \) and \( 1 \leq \tau \leq 2 \) are pre-defined parameters, \( D \) is the shrinkage operator defined in equation 2.6 and \( \mathcal{P}_\Omega \) is the projection operator of \( \Omega \) defined by

\[
\mathcal{P}_\Omega(Q)(i,j) = \begin{cases} 
Q(i,j), & \text{if } (i,j) \in \Omega; \\
0, & \text{otherwise}
\end{cases} \quad (2.14) \tag{2.14}
\]

### 2.3.3 Reconstructing denoised video frame

All patches in a video frame \( f_k \) are denoised by following the two stage grouping and collaborative filtering framework described in sections 2.3.1 and 2.3.2. The patches are sampled such that there is overlap between neighboring patches. The last step of this algorithm is to reconstruct the video frame \( f_k \) from all the overlapping patches. The spatial sampling interval is chosen in such a way that almost all the pixels (except the pixels at corners of the frame) are included in several patches. In reconstructing \( f_k \), the value at each pixel \( j \) is calculated
as the average of pixel values at that particular location obtained from all the patches that cover the pixel $j$. This reconstruction avoids potential artifacts and removes blocky appearance at the boundaries of patches.

2.4 Experiments and analysis

This section presents results from our own implementation of method proposed in [51]. We also present our analysis on this method for gaining profound understanding of the problem. Good insights on the framework presented in [51] help in identifying possible scope for improving [51] and also analyzing inherent limitations of using matrix completion for video denoising, which are presented in subsequent chapters. We choose to proceed with our implementation of [51] and use this as the baseline for any comparisons in subsequent chapters. This choice is made as there is no publicly available implementation of [51]. This choice is justified as detailed analysis has been done to resolve any ambiguity in the presentation of the algorithm in [51] and implementation has been done in such a way to ensure optimal performance of [51].

All the experiments and analysis in this work are done on a test set of videos downloaded from [68]. As discussed in section 2.2.1, the temporal search for similar patches is done over a set of $M$ neighboring frames. In [51], $M = 50$ frames have been used. In our observation we found that using 20 frames will not affect the performance much but saves greatly on computation. So we set $M = 20$ frames. The patch dimension $n$ is 8 pixels and spatial sampling interval is $4 \times 4$ pixels. For a frame $f_k$, each of the sampled patch is taken as the reference patch and $m = 5$ most similar patches from all $M$ frames are picked to form the patch matrix. Thus, patch matrix has the dimensions of $(n^2) \times (Mm)$ which is $64 \times 100$. The threshold to identify reliable pixels
is set as $\alpha \bar{\sigma}$ where $\bar{\sigma}$ is the estimate of standard deviation of noise present in $P_{j,k}$ and $\alpha = 2$ is a constant. $\bar{\sigma}$ is obtained from $\bar{\sigma}^2$ which is the average of all variances calculated across each row of $P_{j,k}$. Detailed discussion on how the performance of the algorithm is affected by varying the $\alpha$ value is presented in Section 4.2.1. The stopping criterion for algorithm 1 is either 30 iterations or $\epsilon \leq 10^{-5}$, whichever is achieved first. The recovered results are evaluated by computing PSNR with respect to the ground truth data. All the results in subsequent chapters are presented by denoising single frame (G frame) from the color (RGB) videos. Brief discussion on extension to color videos is presented in Section 5.1.

The noisy video data is synthesized by corrupting the original videos with mixed noise constituting of Gaussian, Poisson and impulsive noise. The Poisson noise is pixel dependent with variance proportional to the pixel values. The Poisson noise is added as shown in equation 2.15 [69].

\[
g^p_{k}(i,j) = \frac{1}{\eta} \text{poisson}(\eta g_{k}(i,j))
\]  

(2.15)

$g_{k}(i,j)$ represents the clean pixel from frame $k$ and $g^p_{k}(i,j)$ represents the corresponding pixel degraded by Poisson noise. The function $\text{poisson}()$ generates random samples from Poisson distribution with variance $\eta g_{k}(i,j)$. The amount of noise added is controlled by varying the parameter $\eta$. The scaling factor $\frac{1}{\eta}$ ensures that mean of the degraded frame is same as the mean of the original video frame. The additive pixel independent Gaussian noise is represented by $n^q_{k}$. $n^q_{k} \sim \mathcal{N}(0, \sigma^2 I)$ follows a normal distribution with zero mean and variance $\sigma^2 I$. Bipolar impulsive noise with total probability of corruption $s$ is synthesized and the noisy frame $f_k$ degraded by mixed noise is generated as shown.
in equation \(2.16\)

\[
f_k(i, j) = \begin{cases} 
0, & \text{with probability } \frac{s}{2}, \\
255, & \text{with probability } \frac{2}{s}, \\
(g_k^p + n_k^p)(i, j), & \text{with probability } 1 - s.
\end{cases}
\] (2.16)

The parameter set \((\sigma, \eta, s)\) controls the amount of each type of noise in the videos. As discussed in Section 2.3.1, partial denoised data is generated using the adaptive median filtering for the purpose of patch matching. Figure 2.1 shows an Example of noisy and partial denoised frames corresponding to dominant impulsive and dominant Poisson noise. As adaptive median filter is suitable for impulsive noise removal, it can be seen that in presence of dominant Poisson noise the quality of partial denoised image is poor. But such images are acceptable because the partial denoised data is used only for the purpose of patch matching, where as the denoising algorithm works on the corresponding noisy data at the matched locations. When other types of non-impulsive noise are dominant, the quality of partial denoised images by adaptive median filtering is poor, but adequate enough for patch matching. Figure 2.2 shows an example of matched patches within a search range \(R\) from all the 20 neighboring frames. 2.2a shows adaptive median filtered output at noise level \((\sigma = 10, \eta = 0.1, s = 0.2)\). The smaller and larger white square regions show the reference patch and the search range respectively. 2.2b shows the matched patches within the search region from all the 20 neighboring frames.

The resulting noisy patch matrices are denoised using the algorithm discussed in Section 2.3.2. Finally, the video frames are reconstructed from the denoised patch matrices. The PSNR values of one denoised frame from each test sequence using our implementation of [51] are shown in Table 2.1. \(\sigma\) is set as 10 and \(\eta \& s\) are varied as shown in Table 2.1.
Figure 2.1: Examples of Original, Noisy and Adaptive median filtered frames. (b) & (c) show noisy and partial denoised images for dominant impulsive noise. (d) & (e) correspond to dominant Poisson noise.

Figures 2.3 and 2.4 show the denoised output frames from Tempete and Mother & Daughter videos at a particular noise level ($\sigma = 10$, $\eta = 0.2$, $s = 0.1$). It is observed that at a fixed noise level, the performance of this algorithm varies with several factors such as structural content in the frames, amount of motion present in the video etc. More detailed exploration into these factors and possible ways to improve this algorithm are presented in subsequent chapters.
Figure 2.2: Example of matched patches from 20 neighboring frames at noise level ($\sigma = 10, \eta = 0.1, s = 0.2$).

Figure 2.3: Original, Noisy and denoised frames from Tempete video at noise level ($\sigma = 10, \eta = 0.2, s = 0.1$).
Table 2.1: PSNR values of denoised frames for CIF videos using the baseline approach.

<table>
<thead>
<tr>
<th>s / η</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
<th>s / η</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>29.66</td>
<td>27.37</td>
<td>25.07</td>
<td>0.1</td>
<td>27.43</td>
<td>25.89</td>
<td>24.01</td>
</tr>
<tr>
<td>0.2</td>
<td>29.22</td>
<td>26.78</td>
<td>24.48</td>
<td>0.2</td>
<td>26.98</td>
<td>25.64</td>
<td>23.58</td>
</tr>
<tr>
<td>0.3</td>
<td>28.46</td>
<td>26.32</td>
<td>23.94</td>
<td>0.3</td>
<td>26.63</td>
<td>25.17</td>
<td>23.30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>s / η</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>23.72</td>
<td>22.89</td>
<td>21.73</td>
</tr>
<tr>
<td>0.2</td>
<td>23.46</td>
<td>22.65</td>
<td>21.47</td>
</tr>
<tr>
<td>0.3</td>
<td>23.09</td>
<td>22.30</td>
<td>21.18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>s / η</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>26.90</td>
<td>24.79</td>
<td>22.37</td>
</tr>
<tr>
<td>0.2</td>
<td>26.51</td>
<td>24.30</td>
<td>21.80</td>
</tr>
<tr>
<td>0.3</td>
<td>25.94</td>
<td>23.90</td>
<td>21.36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>s / η</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>28.81</td>
<td>26.66</td>
<td>24.12</td>
</tr>
<tr>
<td>0.2</td>
<td>28.32</td>
<td>26.10</td>
<td>23.59</td>
</tr>
<tr>
<td>0.3</td>
<td>27.77</td>
<td>25.59</td>
<td>23.20</td>
</tr>
</tbody>
</table>

Figure 2.4: Original, Noisy and denoised frames from Mother & Daughter video at noise level (σ = 10, η = 0.2, s = 0.1).
Chapter 3

PATCH BASED STUDY FOR MATRIX COMPLETION

3.1 Grouping relevant patches in presence of high motion

As discussed in the problem formulation in section 2.2.1, matrix completion for denoising is based on the idea that the columns of $Q_{j,k}$ are structurally similar to each other. So for optimal performance of this algorithm, it is very important to group structurally similar patches together for matrix completion. As mentioned in Section 2.2.1, $m$ best matches are selected from each of the $M$ frames to form $Q_{j,k}$. i.e. fixed number of patches are selected from each frame. Though this method groups reliable patches in videos with low motion, it is disadvantageous in cases where the videos contain high motion. One example which can be easily visualized is the case when a particular object suddenly moves out or into the search region. Figure 3.1 shows such an example. 3.1b shows the best 5 matches picked from each of the 20 neighboring frames. By selecting fixed number of patches from each frame, the algorithm is forced to pick patches even from frames in which the reference object is not present in the search region. As seen in 3.1b, the algorithm picks closest possible matches from each frame even if the object in reference patch is not present in the search region. This way, the algorithm is forced to pick structurally irrelevant patches for matrix completion. Increasing the search range is not a solution for this problem. In figure 3.2 matched patches are shown when the entire frame is searched with no limit on the search range. The black patch represents the reference patch. It can be clearly seen that even now, many irrelevant patches are picked from the first few frames where the reference object is not present. Moreover when patch matching is done in presence of noise, increasing search range by large amount is counter productive. In presence of
noise, patch matching is not accurate. In such a case, searching in an excessive range increases the probability of mismatched patches.

Instead, if it was allowed to pick $mM$ (100 in this case) number of patches over all from all the $M$ frames without the limit of $m$ patches per frame, all the grouped patches would be more structurally relevant to each other. We propose to achieve this by selecting $l > m$ (say 10) patches from each frame and then picking the best $mM$ patches from the pool of already selected $lM$ patches. This gives the algorithm the freedom to select 0 to $l$ number of patches from each frame based upon the relevance in content to the reference patch. For every $r^{th} (r > m)$ patch selected from a particular frame, a less relevant patch is dropped from a different frame, thus still maintaining the same number of $mM$ patches overall from all the frames. Patches grouped using this proposed way are shown in Figure 3.3. It can be seen that no
patches are selected from the frames in which the reference object is not present the search region. Instead, more number of relevant patches are picked from other frames. This gives better groups of patches for the matrix completion algorithm, especially in videos with high amount of motion.

The above discussion is shown on clean images for the ease of understanding. Figure 3.4 shows patch matching on partially denoised data in presence of noise. 3.4b shows matched patches by using baseline approach and 3.4c shows matched patches using our approach. It can be noticed that the
above discussion is valid even in presence of noise.

3.2 Separating patches with low image variance

By observing the denoised images from obtained from the baseline algorithm, it was noticed that the plain regions such as background still contained noticeable amounts of noise compared with image regions containing structural details. Figure 3.5 shows the visual quality of denoised images from mobile and foreman videos at same noise level. It can be noticed that visual quality of mobile image is much better than the foreman image. Mobile image is high in structural content whereas foreman image contains more plain background region. Similar trend can be noticed in Figures 2.3c and 2.4. Tempete image looks much cleaner than Mother & daughter image at same noise level. Figure 3.6 shows two different denoised regions from a same image. Noise is attenuated to greater extent in the region containing eye and hair compared to the
plain region. From these observations, it was identified that regions with low structural content are not denoised as much as regions with high structural content. The reason for this could be: in the matrix completion algorithm, the algorithm tries to recover the underlying structure of the reference patch based upon the reliable pixels present in $P|_\Omega$. During the process, the elements in $P|_\Omega$ are also modified to some extent as they are not perfectly noise-less pixels. In essence, the algorithm tries to recover the underlying structure while attenuating the noise as much as possible. In image regions like background where the structural content of image is very low, there is a possibility that the noise dominates over the image structure and thus the algorithm retains noise. This could be the reason for noticeable amounts of noise in low structured regions of the denoised image. This can be investigated more if one can separate the image patches corresponding to low structured regions from the noisy image.

To understand the problem, first a simple case where the image is corrupted only by additive Gaussian noise $n^*_k$ is considered. As the variance of the Gaussian noise is pixel independent, the noise and the image are uncorrelated.
Figure 3.5: Visual quality of denoised images at same noise level \((\sigma = 10, \eta = 0.1, s = 0.2)\) with different structural content.

Figure 3.6: Visual quality of denoised regions with high and low structural content.
In this case of only Gaussian noise, \( f_k = g_k + n_k^g \). Where \( f_k \) is the noisy image and \( g_k \) is the clean image. Then, the variance of the noisy image is given by the expression shown in equation 3.1.

\[
Var(f_k) = Var(g_k) + Var(n_k^g) + 2Cov(g_k, n_k^g)
\]  

(3.1)

\[
Var(f_k) = Var(g_k) + Var(n_k^g)
\]  

(3.2)

In this case, as the image and noise are uncorrelated, \( Cov(g_k, n_k^g) = 0 \) and equation 3.1 reduces to 3.2. Extending the same relation to the grouped patch matrix, variance of the patch matrix would be as shown in equation 3.3.

\[
Var(P_{j,k}) = Var(Q_{j,k}) + Var(N_{j,k})
\]  

(3.3)

Considering infinite number of matrix entries, \( Var(N_{j,k}) = Var(n_k^g) \). For matrices of finite size, \( Var(N_{j,k}) \sim Var(n_k^g) \). Let \( \sigma_c \) denote the variance of the patch matrix calculated as the average of variances across each column. Average of column variances is considered as each column represents a patch and thus column variance is an indication of amount of image structure present in a particular patch. From equation 3.3 value of \( \sigma_c \) will approximately be equal to sum of variance of clean image patch and standard deviation of the noise added. Gaussian noise of standard deviation (say 30) is added to the video frame and the distribution \( \sigma_c \) corresponding to all patch matrices is observed. Figure 3.7 shows the histogram of \( \sigma_c \) corresponding to all patch matrices in a particular frame. For patches with very low image variation such as plain regions, \( Var(Q_{j,k}) \) in equation 3.3 tends towards zero and \( \sigma_c \) will be approximately equal to the noise standard deviation (30 in this case). This means all the patches falling in bins close to the noise standard deviation value (30 in this case) and in bins less than 30 correspond to patches with low image content. Considering such histograms over several images and several noise
standard deviation values, it was observed that the peak of the histogram (bin containing maximum number of patches) occurs close to the noise standard deviation value. Figure 3.7 shows an example. This implies that most of the patch matrices in an image correspond to low variance/(low structured) patches. (This observation is supported by natural image statistics). So given a noisy video frame, all the patch matrices that fall in bins located at $\sigma_c$ less than $\sigma_c$ corresponding to the histogram’s peak are classified as patches with low image variance. i.e the patch matrices falling on left side of the histogram’s peak are classified as low structured patches. Matrix completion is not done on such patch matrices. Matrix completion is done only on patch matrices with considerable image structure. To investigate the usefulness of separating low structured patches, the following is done: Patches with low
image variance are recovered by taking average of reliable pixels in $P|_{\Omega}$ on each row. Patches containing reasonable amount of image details are recovered by subjecting the grouped patch matrices to matrix completion. The video frame is reconstructed from all the recovered patches. Repeating this experiment for various levels of Gaussian noise showed consistent improvement in PSNR value of the recovered images compared to the case when all the patch matrices are subjected to matrix completion. This implies that even a simple averaging of reliable pixels is better than matrix completion for low structured patches. This suggests the inherent limitation of matrix completion in recovering images or image regions where noise dominates the structural properties of the image.

![Histogram of $\sigma_c$ values in presence of mixed noise obtained from the noisy data.](image)

Figure 3.8: Histogram of $\sigma_c$ values in presence of mixed noise obtained from the noisy data.

In the above discussion, videos corrupted by pure Gaussian noise are
considered. Direct extension of this discussion to mixed noise case is not appropriate and it will not yield meaningful results. In mixed noise discussed in section 2.4, noise constitutes of Gaussian, Poisson and impulsive noise. Poisson noise as defined in equation (2.15) is pixel dependent noise. So in presence of mixed noise, the image and noise are correlated and thus equation (3.1) will not reduce to equation (3.2). Moreover, the presence of impulsive noise affects the $\sigma_c$ values of the patch matrix to great extent. In such a case, it is difficult to draw any conclusion on the image variance from the values of $\sigma_c$. An example of histogram of $\sigma_c$ values in the presence of mixed noise is shown in Figure 3.8. It can be noticed from Figure 3.8 that unlike the previous case of pure Gaussian noise, histogram of $\sigma_c$ does not represent the distribution of variances of clean image patches. It is difficult to even approximately classify low structured patches based upon such a histogram.

As discussed in section 2.3.1, adaptive median filter reduces the affect of large distortion of pixel values caused due to impulsive noise on the calculated similarity measures. Similarly at this stage, adaptive median filter is employed to reduce the large deviation caused in $\sigma_c$ values due to impulsive noise. The adaptive median filtered output is not clean in the presence of mixed noise. Adaptive median filtering reduces the distortion caused due to impulsive noise and smooths non-impulsive noise. Though the adaptive median filtered output is still noisy in the presence of high amounts of mixed noise, we expect to obtain a reasonably approximate distribution of $\sigma_c$ values by using the adaptive median filtered data. Instead of patch matrices containing noisy data, we use patch matrices with corresponding adaptive median filtered data to compute the histogram of $\sigma_c$ values. Figure 3.9 shows the histogram of $\sigma_c$ values computed from adaptive median filtered data. Once such histogram is computed,
similar technique discussed in pure Gaussian noise case is used to classify low structured patches. Only the patches with considerable image details are subjected to matrix completion and the recovered video frame is reconstructed. Experimental results for different levels of mixed noise and comparison with baseline approach are presented in section 3.3.

3.3 Results and discussion

Experiments are carried out by introducing both the proposed techniques discussed in Section 3.1 and Section 3.2 to improve the baseline algorithm. Table 3.1 shows the results for the 8 test videos using our approach of grouping relevant patches discussed in Section 3.1. This approach is useful to group structurally relevant patches, especially in presence of high motion and hence
preserves structure better than the baseline approach. Figure 3.10 shows an example.

Then separating low structured patches discussed in section 3.2 is implemented and Table 3.2 shows results on the 8 test videos for various levels of noise.
Table 3.1: PSNR values of denoised frames for CIF videos by grouping structurally relevant patches in presence of motion.

(a) Akiyo

<table>
<thead>
<tr>
<th>s / k</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>29.72</td>
<td>27.44</td>
<td>25.14</td>
</tr>
<tr>
<td>0.2</td>
<td>29.29</td>
<td>26.86</td>
<td>24.54</td>
</tr>
<tr>
<td>0.3</td>
<td>28.54</td>
<td>26.40</td>
<td>24.01</td>
</tr>
</tbody>
</table>

(b) Bowing

<table>
<thead>
<tr>
<th>s / k</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>28.27</td>
<td>26.43</td>
<td>24.14</td>
</tr>
<tr>
<td>0.2</td>
<td>27.80</td>
<td>26.04</td>
<td>23.79</td>
</tr>
<tr>
<td>0.3</td>
<td>27.42</td>
<td>25.70</td>
<td>23.38</td>
</tr>
</tbody>
</table>

(c) Bus

<table>
<thead>
<tr>
<th>s / k</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>24.28</td>
<td>23.33</td>
<td>22.05</td>
</tr>
<tr>
<td>0.2</td>
<td>23.97</td>
<td>23.05</td>
<td>21.75</td>
</tr>
<tr>
<td>0.3</td>
<td>23.58</td>
<td>22.67</td>
<td>21.42</td>
</tr>
</tbody>
</table>

(d) Football

<table>
<thead>
<tr>
<th>s / k</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>24.34</td>
<td>23.19</td>
<td>21.66</td>
</tr>
<tr>
<td>0.2</td>
<td>24.06</td>
<td>22.85</td>
<td>21.35</td>
</tr>
<tr>
<td>0.3</td>
<td>23.68</td>
<td>22.49</td>
<td>21.03</td>
</tr>
</tbody>
</table>

(e) Foreman

<table>
<thead>
<tr>
<th>s / k</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>27.06</td>
<td>24.91</td>
<td>22.40</td>
</tr>
<tr>
<td>0.2</td>
<td>26.67</td>
<td>24.42</td>
<td>21.93</td>
</tr>
<tr>
<td>0.3</td>
<td>26.09</td>
<td>24.00</td>
<td>21.50</td>
</tr>
</tbody>
</table>

(f) Mobile

<table>
<thead>
<tr>
<th>s / k</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>21.79</td>
<td>20.47</td>
<td>18.86</td>
</tr>
<tr>
<td>0.2</td>
<td>21.37</td>
<td>20.09</td>
<td>18.54</td>
</tr>
<tr>
<td>0.3</td>
<td>20.72</td>
<td>19.51</td>
<td>18.05</td>
</tr>
</tbody>
</table>

(g) Mother & Daughter

<table>
<thead>
<tr>
<th>s / k</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>28.91</td>
<td>26.66</td>
<td>24.15</td>
</tr>
<tr>
<td>0.2</td>
<td>28.35</td>
<td>26.16</td>
<td>23.62</td>
</tr>
<tr>
<td>0.3</td>
<td>27.90</td>
<td>25.79</td>
<td>23.18</td>
</tr>
</tbody>
</table>

(h) Tempete

<table>
<thead>
<tr>
<th>s / k</th>
<th>0.2</th>
<th>0.1</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>25.24</td>
<td>24.14</td>
<td>22.62</td>
</tr>
<tr>
<td>0.2</td>
<td>24.86</td>
<td>23.74</td>
<td>22.33</td>
</tr>
<tr>
<td>0.3</td>
<td>24.38</td>
<td>23.32</td>
<td>21.98</td>
</tr>
</tbody>
</table>
Table 3.2: PSNR values of denoised frames for CIF videos by separating low-structured patches.

<table>
<thead>
<tr>
<th></th>
<th>Akiyo</th>
<th></th>
<th>Bowing</th>
<th></th>
<th>Football</th>
<th></th>
<th>Mobile</th>
<th></th>
<th>Mother &amp; Daughter</th>
<th></th>
<th>Tempete</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s / k</td>
<td>0.2</td>
<td>0.1</td>
<td>0.05</td>
<td></td>
<td>0.2</td>
<td>0.1</td>
<td>0.05</td>
<td></td>
<td></td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>0.1</td>
<td>29.84</td>
<td>27.56</td>
<td>25.22</td>
<td>0.1</td>
<td>28.70</td>
<td>26.79</td>
<td>24.64</td>
<td></td>
<td></td>
<td>0.1</td>
<td>25.24</td>
</tr>
<tr>
<td>0.2</td>
<td>29.36</td>
<td>26.96</td>
<td>24.57</td>
<td>0.2</td>
<td>28.14</td>
<td>26.47</td>
<td>24.22</td>
<td></td>
<td></td>
<td>0.2</td>
<td>24.86</td>
</tr>
<tr>
<td>0.3</td>
<td>28.66</td>
<td>26.52</td>
<td>24.06</td>
<td>0.3</td>
<td>27.77</td>
<td>26.04</td>
<td>23.87</td>
<td></td>
<td></td>
<td>0.3</td>
<td>24.37</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
As discussed in section 2.3.2, the fixed point iterative algorithm shown in Algorithm 1 is used to recover \( Q \) by solving the equation 2.9. In algorithm 1, the two steps shown in Equations 2.13a and 2.13b are iteratively repeated until the stopping criterion is met. To discuss the implications of these equations, 2.13b imposes the low rank criterion on \( Q \) by thresholding on the singular values. The threshold \( \tau \mu \) controls the amount of information discarded in this step. From equation 2.11, as \( \mu \) directly depends on \( \hat{\sigma} \), the estimate of amount of noise present in \( P|_{\Omega} \), this step discards the noise present in the retained pixels and recovers noiseless values of missing elements such that the low rank property of the recovered \( Q \) is maintained. As the elements in \( P|_{\Omega} \) are considered as reliable pixels, the deviation of elements in \( Q|_{\Omega} \) from the elements in \( P|_{\Omega} \) is not allowed to be too large. This condition is imposed as the inequality constraint in equation 2.8 which is later solved by using its unconstrained Lagrangian version in equation 2.9. This condition is introduced to ensure that the recovered patch matrix does not drift away from the original image information in the process of recovering missing elements and denoising the patch matrix by nuclear norm minimization. Since the pixels in \( P|_{\Omega} \) are not perfectly noise less, change in pixel values of \( P|_{\Omega} \) is allowed during the process of recovery. But the amount of allowed change is limited by a threshold as they are considered as the reliable pixels meaning that they are corrupted by relatively low amplitudes of noise. This condition appears in the iterative algorithm as Equation 2.13a. In every iteration \( k \), the difference between \( Q^{(k)} \) and \( P \) for pixels \( \in \Omega \) is penalized in equation 2.13a. Iteratively repeating
the equations 2.13a and 2.13b converges the algorithm to the optimal $Q$. To ensure convergence of the algorithm, the $\tau$ value is selected from a particular range $[a,b]$.

From the patch matching technique introduced in section 3.1, the top $mM$ patches overall from all the $M$ frames are grouped for matrix completion. By doing so, unlike the previous case, the columns of $P$ are in the order of their structural relevance to the reference patch. This can be explained as when moving from first column to the $mM^{th}$ column in the matrix $P$, any particular column is more relevant and important in recovering the reference patch than the columns occurring later. In equation 2.13a $\tau$ is a scalar value employed in penalizing $Q^{(k)} - P$ for pixels $\in \Omega$ and $\tau$ lies in the range $[1,2]$ to ensure convergence. By scaling all columns of $P_{\Omega}(Q^{(k)} - P)$ with $\tau$, the deviation of recovered patch from all columns of $P|_{\Omega}$ is penalized equally. i.e. all columns of $P|_{\Omega}$ are treated equally important and relevant in recovering the reference patch. But, by using the patch matching method discussed in section 3.1, the columns of $P$ are in the order of their relevance to the reference patch. So by scaling each column of $P_{\Omega}(Q^{(k)} - P)$ with a different $\tau$ value in equation 2.13a, it is possible to penalize deviation from different columns by different amounts according to their importance in recovering the reference patch. This means priority is introduced to the columns of $P|_{\Omega}$ by varying the scaling factor $\tau$ for each column of the projection, $P_{\Omega}(Q^{(k)} - P)$ in equation 2.13a.

In the algorithm discussed in section 2.3.2 $\tau$ is a pre-defined constant chosen from a particular range. We propose to dynamically update $\tau$ for each patch matrix instead of choosing a constant value for all the groups of patches. In the presence of noise, it is hard to guarantee that the patch matrix
constitutes the most similar patches to the reference patch. Considering the random nature of the noise, it is understandable that the reliability on the grouped patches varies for each reference patch. So, it is useful to dynamically update $\tau$ for each patch matrix i.e. in recovering each reference patch. For each patch matrix, we also propose to use variable scaling factor for each column of the projection $P_{\Omega}(Q^{(k)} - P)$ in equation 2.13a in order to introduce priorities to the columns of $P_{\Omega}$. The scaling factor for each column depends upon the MAD value of the particular column in $P$ from the reference patch. As MAD values indicate similarity between the patches, the scaling factor is decided by the MAD values so as to prioritize the columns according to their similarity.

Let $s$ be a $1 \times mM$ row vector with each element $s_i$ representing MAD between each column of $P$ and the reference patch. Equation 4.1 shows the computation of the proposed variable scaling factor for each column.

\[
\begin{align*}
    s_1 &= s_2 \\
    \tau &= b1 - \left( \frac{b - a}{s_{mM}} [s - s_1 1]\right) \\
    \text{where } 1 &= [1, 1, \ldots, 1]_{1 \times mM}
\end{align*}
\]  

Instead of the scalar $\tau$ in equations 2.13a and 2.13b we compute a vector $\tau$ where each element of $\tau$ represents scaling factor for each column of $P_{\Omega}(Q^{(k)} - P)$. It can be seen from equation 4.1 that $\tau$ is computed in such a way that all of its elements lie in the range $[a,b]$. $\tau$ in 4.1 is designed such that the column corresponding to least MAD value is assigned the highest scaling factor and as the MAD values increase, the corresponding columns of $P_{\Omega}$ are assigned decreasing values of the scaling factor. This implies that deviation from columns with low MAD values is penalized more compared to deviation from columns with high MAD values. So, high priority is given to columns which are more similar. We are aiming to recover the reference patch in such a
way that it is more similar to the columns with lower MAD values. In equation 4.1, the adjustment \( s_1 = s_2 \) is made for the following reason: the first entry of the \( s \) vector will always be 0 since it represents the MAD of the noisy patch to be denoised with itself. By having 0 as the first entry of \( s \), very high priority will be given to the first column and all other columns will be given relatively very low priorities. This is not intuitive as the actual patch itself is noisy. To overcome this the adjustment \( s_1 = s_2 \) is made which means that for the purpose of calculating priorities, the first column is treated as equally similar and equally important as the next closest patch.

Since we vary the scaling factor for each column, the equations 2.13a and 2.13b in Algorithm 1 are modified as shown in 4.2a and 4.2b.

\[
\begin{align*}
R^{(k)} &= Q^{(k)} - ((1^T \tau) \circ P_{\Omega}(Q^{(k)} - P)), \\
Q^{k+1} &= D_{\tau_a \mu}(R^{(k)}), \\
\tau_a &= \frac{1}{MM} \sum_i \tau_i
\end{align*}
\]

(4.2a, 4.2b, 4.2c)

The \( \circ \) in equation 4.2a represents Hadamard product which is element-wise multiplication of two matrices. As all the elements of \( \tau \) lie in the range \([a, b]\), \( \tau_a \in [a, b] \).

4.2 Results and discussion

Experimental results after introducing the proposed technique discussed in Section 4.1 are shown in Table 4.1.

Figures 4.1 to 4.8 show the comparison of results obtained from the proposed methods with baseline approach for all the 8 videos. For each video the results from the proposed methods are presented at 9 different noise levels to show the consistency of results at several noise levels. It can be observed
Table 4.1: PSNR values of denoised frames for CIF videos by prioritizing patches for MC.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR Values (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s / k 0.2 0.1 0.05</td>
</tr>
<tr>
<td></td>
<td>0.1   0.2   0.3</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
</tr>
<tr>
<td>(a) Akiyo</td>
<td>30.42 29.93 29.27</td>
</tr>
<tr>
<td></td>
<td>25.51 24.93 24.35</td>
</tr>
<tr>
<td>(b) Bowing</td>
<td>29.09 28.63 28.17</td>
</tr>
<tr>
<td></td>
<td>24.84 24.37 24.00</td>
</tr>
<tr>
<td>(c) Bus</td>
<td>24.99 24.73 24.18</td>
</tr>
<tr>
<td></td>
<td>22.57 22.21 21.83</td>
</tr>
<tr>
<td>(d) Football</td>
<td>28.19 27.71 27.22</td>
</tr>
<tr>
<td></td>
<td>23.65 22.89 22.58</td>
</tr>
<tr>
<td>(e) Foreman</td>
<td>22.77 22.31 21.54</td>
</tr>
<tr>
<td></td>
<td>19.57 19.20 18.74</td>
</tr>
<tr>
<td>(f) Mobile</td>
<td>29.75 29.26 28.92</td>
</tr>
<tr>
<td></td>
<td>24.94 24.40 23.89</td>
</tr>
<tr>
<td>(g) Mother &amp; Daughter</td>
<td>25.91 25.57 25.13</td>
</tr>
<tr>
<td></td>
<td>23.10 22.72 22.34</td>
</tr>
<tr>
<td>(h) Tempete</td>
<td>20.09 20.21 20.34</td>
</tr>
<tr>
<td></td>
<td>19.20 18.74 18.47</td>
</tr>
</tbody>
</table>

that the results are completely in coherence with the motivations behind the proposed algorithm. Let method 1 denote the denoising algorithm after introducing the technique of grouping structurally relevant patches. Method 2 denotes results after introducing both the techniques of grouping structurally relevant patches and separating low structured patches. Method 3 denotes the results after prioritizing the patches for matrix completion in method 2. In Akiyo video as the motion content is very low, improvement of method 1 over the baseline approach is also low as expected (Figure 4.1). As Bowing video has high motion content, it can be seen from Figure 4.2 that PSNR val-
ues from method 1 are much higher than the values obtained by using the baseline approach. Also, as there is considerable amount of low structured region in Bowing video, it can be seen that method 2 results in considerable improvement in performance over method 1.

As there is high motion content in Bus video, it can be noticed from Figure 4.3 that method 1 gives high performance improvement over the baseline approach. Method 2 is not expected to show improvement over method 1 for this video and it can be seen that the results are consistent with the underlying ideas of algorithm.

As football video has reasonably high motion, method 1 shows considerable improvement in performance over the baseline approach. AS the football
video also has some amount of low structured region, method 2 consistently shows some performance improvement over method 1 as shown in Figure 4.4.

As the foreman video contains high amount of low structured region or plain region, it can be seen from Figure 4.5 that method 2 shows high performance improvement over method 1 consistently at all noise levels.

The mobile video does not contain high motion content. Also, it does not contain high amount of low structured or plain region. Therefore, as expected improvement is performance of method 1 and method 2 is not very high as shown in Figure 4.6.

The mother & Daughter video has almost no motion and thus, method 1 is not expected to show increase in PSNR over the baseline approach. It can
be seen from Figure 4.7 that the same is the case. This video has considerable amount of plain background region and thus, method 2 shows consistent improvement over method 1 at all noise levels.

The Tempete video has almost no motion and no low structured region. So, method 1 and method 2 are not expected to show significant improvement in performance. It can be seen from Figure 4.8 that the experimental results show the same. For this video, only method 3 contributes for all the increase in PSNR values.

From the results of all the 8 videos, several observations can be made.

- The results for all the videos are consistent across several noise levels.
Figure 4.4: Comparison of all the four methods for Football video.

- The results for different videos (with varying motion content & structural content) are in perfect agreement with the underlying ideas behind the proposed methods.

- Method 3 gives improvement in performance consistently for all videos irrespective of motion content or structural content.

- The improvement shown by method 3 over method 2 is inversely related to improvement shown by method 2 over method 1. This can be observed from results for Foreman and Football videos. In Foreman video, improvement by method 2 over method 1 is high. In this case, improvement by method 3 over method 2 is not very high. Whereas in Football video, improvement by method 2 over method 1 is not very high and im-
Figure 4.5: Comparison of all the four methods for Foreman video.

provement by method 3 over method 2 is high. This behavior is also in accordance with the underlying idea. If method 2 shows high improvement in performance (Foreman video), it indicates that there is large amount of low structured region in the video frame implying that large number of patch matrices are not subjected to matrix completion. In this case prioritizing of patch matrices for matrix completion is applicable to less number of patch matrices compared to the case of Football video. Hence performance improvement by method 3 will not be as high as the improvement for Football video. Therefore, it can be noticed that all the experimental results are in exact agreement with the underlying ideas of the proposed methods.
The results presented above correspond to single frame from each video under the assumption that the same trend would extend over many frames. To verify the consistency, the algorithm was tested for Foreman video over 100 frames by sampling one frame per every 5 frames. Figure 4.9 shows the plot of PSNR values over 100 frames. It can be seen that PSNR values over several frames are consistent.

Figure 4.14 shows an example of reconstructed images from baseline approach and from the proposed approach (method 3) for visual comparison. It can be noticed that the structural content is preserved by the proposed method compared to baseline approach (especially in face region). The baseline method produces blocky nature for objects under high motion (The person in Figure 4.14). This has been reduced by the proposed algorithm.
4.2.1 Artifacts in a particular case.

Though the baseline method and the proposed approach are robust in general for any level of mixed noise or high amounts of Gaussian noise, it is noticed from our experiments that both the algorithms are unstable in presence of Gaussian noise with low standard deviation values. Both the methods introduce artifacts in some particular cases in presence of pure Gaussian noise with low $\sigma$ values. The nature of the artifacts and the cases in which artifacts are introduced vary for both the methods. This section presents detailed discussion on such particular cases.

In the baseline algorithm the incomplete version of the noisy patch matrix, $P_{j,k}$, is formed by retaining only the reliable pixels from the noisy matrix.
The pixels which deviate from the mean of the corresponding row vector by an amount larger than a predefined threshold are considered as unreliable pixels. Such elements represent pixels corrupted by high amount of noise. The threshold to identify reliable pixels is chosen as $\alpha \bar{\sigma}$ where $\bar{\sigma}$ is the estimate of standard deviation of noise present in $P_{j,k}$. $\bar{\sigma}$ is obtained from $\bar{\sigma}^2$ which is the average of all variances calculated across each row of $P_{j,k}$. The constant $\alpha$ is chosen as 2 in the baseline algorithm. Therefore, if the difference between any pixel value and the mean of the corresponding row vector is greater than $2\bar{\sigma}$, the pixel is discarded as unreliable pixel. In our experiments with the baseline algorithm we varied $\alpha$ to control the threshold $\alpha \bar{\sigma}$ which identifies the reliable pixels. Variation of this threshold influences the percentage of pixels retained in $P|_{\Omega}$ for the matrix completion algorithm. As a part of extensive analysis,
PSNR values of denoised video frames are observed for several threshold values by varying $\alpha$ (i.e. performance variance with percentage of retained pixels in $P|_{\Omega}$ is observed). In case of pure Gaussian noise, each row of $P_{j,k}$ approximately follows a Gaussian distribution. Each row of $P_{j,k}$ would exactly follow Gaussian distribution under the assumption that variance of the corresponding row in the clean patch matrix $Q_{j,k}$ is zero. In reality, variance of each row of $Q_{j,k}$ would be close to zero, but it might not be equal to zero due to mismatched patches in presence of noise. Therefore, each row of $P_{j,k}$ approximately follows a normal distribution. In such case, the percentage of pixels retained for different values of threshold (in terms of multiples of standard deviation) on the difference from the mean is shown in the Figure 4.10. So choosing $\alpha = 2$ as in baseline algorithm implies that almost 95% of the pixels
in the noisy patch matrix are retained as reliable pixels. Retaining 95% of the pixels as reliable pixels is counter-intuitive, especially in presence of noise with higher standard deviation values. Only 5% of elements from the noisy patch matrix are discarded as highly noise corrupted pixels and the remaining 95% of pixels are used in recovering the clean low rank patch matrix. This is counter-intuitive because most of the retained reliable pixels are still corrupted by high amplitudes of noise values, especially in presence of noise with high standard deviation. For low rank matrices, the matrix completion algorithm can recover the complete noise less matrix with much fewer entries. Also if most of the retained pixels are corrupted by high amplitudes of noise, the recovered patch matrix will also be noisy. Based on these insights, the performance of the algorithm is observed at various threshold values $\alpha \bar{\sigma}$ and several noise levels to
understand how these parameters affect the algorithm. Figure 4.11 shows the variation in % of retained pixels with $\alpha$ for a set of 7 QCIF videos. During this analysis, it is noticed that there is some instability in the baseline algorithm in presence of low levels of Gaussian noise. Noticeable artifacts are introduced at lower values of $\alpha$ in presence of Gaussian noise with low standard deviation values. It can be seen from Figures 4.11a and 4.11b that almost same amount of pixels are retained for a particular $\alpha$ value in presence of Gaussian noise with standard deviations 10 and 30. Artifacts are noticed for some $\alpha$ values in presence of Gaussian noise with $\sigma = 10$ while they are not noticed for the same $\alpha$ values in presence of noise with $\sigma = 30$. Precisely, for some particular $\alpha$ values artifacts introduced are dominant in presence of noise with $\sigma = 10$, less dominant when $\sigma = 20$ and not noticeable when $\sigma = 30$. This shows that discarding more pixels from the patch matrix is not the sole reason for the artifacts introduced. This instability exists only in presence of low levels of noise. Also, it has been observed that the artifacts are introduced mostly in plain, bright regions of the video. Figure 4.12 shows examples of artifacts introduced by the baseline method for $\alpha = 0.7$ in presence of Gaussian noise with $\sigma = 10$. It can be noticed that the artifacts are mostly present in plain, bright regions of the video. In Figure 4.12a for the Akiyo video, 53.91% of pixels are retained on an average over all the patch matrices and in Figure 4.12b for the Foreman case, 53.47% of pixels are retained on an average. The nature and dominance of the artifacts also depends upon the content of the video. In presence of Gaussian noise with high standard deviation values where there are no artifacts introduced, the performance of baseline algorithm is better at lower $\alpha$ values such as 0.6 or 0.7 for some videos and it is better at $\alpha = 2$ for some videos. So it is difficult to choose a common threshold which gives best possible performance for all videos. For all the experimental results presented
in previous chapters and sections, $\alpha$ is chosen as 2 since this value of $\alpha$ gives stability in performance of the algorithm at any noise level (at higher $\alpha$ values such as 2, no artifacts are introduced even at low levels of Gaussian noise).

After introducing method 1 and method 2 of the proposed algorithm, the behavior of the artifacts is same as the case of baseline algorithm which is discussed above. It is noticed that the behavior of the artifacts changes with the introduction of method 3 of the proposed approach. Even in this case artifacts are introduced in presence of low levels of Gaussian noise, but the appearance of artifacts is different from the case of baseline approach. Figure 4.13 shows the artifacts introduced by the proposed approach in presence of Gaussian noise of standard deviation 10 and $\alpha = 0.85$. Another important difference noticed is that method 3 of proposed approach introduces artifacts at wider ranges of $\alpha$ values compared to baseline method. In presence of low levels of Gaussian noise, baseline method introduces artifacts at lower $\alpha$ values and the artifacts gradually disappear as $\alpha$ is increased. But in case of method 3 of proposed approach at low levels of Gaussian noise, artifacts occur at lower $\alpha$ values, the dominance of artifacts decreases with increase in $\alpha$ value and after some extent the dominance of artifacts increases with further increase in $\alpha$ value. Also, the artifacts are highly dominant at higher $\alpha$ values. From overall range of $\alpha$ values, it can be noticed that method 3 of proposed approach introduces artifacts at much wider range of $\alpha$ values compared to the baseline method.

4.2.2 Discussion on speed performance.

Currently the proposed algorithm is not suitable for real time applications and it is not optimal in terms of speed performance. The focus of this work is
Figure 4.11: Variation in % of retained pixels with $\alpha$ for a set of 7 videos in presence of low and high levels of pure Gaussian noise.
mainly on providing extensive analysis on application of matrix completion to video denoising as the existing literature lacks the details and analysis which can provide deep insights into the matrix completion framework for video denoising. In the process, some shortcomings of the baseline algorithm are identified and novel techniques are introduced to overcome the shortcomings and improve the performance of the baseline approach. As the main focus is on the three challenges listed below, this algorithm is not designed in a computationally optimal way.
• Providing extensive analysis to enable in depth understanding of the framework.

• Improving the noise attenuation capability of the algorithm.

• Preserving structural content of the original video data.

Therefore, there exists scope to improve the speed performance of the algorithm in future. In the current design of the algorithm, the most time consuming part is the patch matching step. EBMA algorithm is used to identify similar patches in spatial and temporal domain. Though EBMA is optimal in identifying similar patches, it is computationally expensive. Replacing EBMA by computationally efficient block matching methods such as [70], [64] and [65] will reduce the execution time taken by current algorithm by large amount.
(a) Denoised result from the baseline approach.

(b) Denoised result from the proposed approach.

Figure 4.14: Visual comparison of reconstructed images.
Chapter 5

CONCLUSIONS AND FUTURE WORK

In this thesis, the application of matrix completion to the video denoising problem is studied in detail. Video denoising is explored in the patch-based, grouping and collaborative framework. An existing state-of-the-art algorithm in which the collaborative filtering task is achieved by using matrix completion is implemented. Thorough analysis is done to better understand the problem and the implications of applying matrix completion on the problem. Results on videos containing different structural content and different amount of motion were examined carefully. The contributions of this work are:

- Analysis and discussions are presented to provide deep insights into the matrix completion framework for video denoising. As the existing literature lacks such details, this work helps in bridging the gap and facilitating further exploration. In the process, some shortcomings of the baseline approach are identified and novel techniques are introduced to overcome the shortcomings and improve the baseline method.

- A new technique for grouping structurally-relevant patches in presence of high motion is proposed. Experimental results showed that this technique improves the performance, especially in presence of high motion. Visual comparison of the results showed that this technique preserves the structural content in the video in presence of motion.

- A framework is designed to classify low structured patches from noisy video data. Visual inspection of denoised frames from the baseline approach suggested that relatively plain regions are not denoised as much as other image regions. To investigate this observation, low structured
patches are separated using the designed framework. Denoising of such patches is dealt in a different way instead of using matrix completion. Experiments showed improvement in performance indicating that matrix completion is counter-productive on low structured patch matrices.

- Results suggesting inherent limitation of matrix completion are presented. The limitation of matrix completion lies in attenuating noise when noise dominates structural properties of the image.

- The patches grouped for matrix completion are ranked and prioritized based on their similarity to the reference patch. Experimental results showed that this method gives improvement in performance in general.

- Instabilities found in both baseline algorithm and proposed approach in specific cases are discussed in detail.

5.1 Future work

In equation 4.1 in section 4.1, $\tau$ is designed based upon the MAD values. Equation 4.1 shows one way of computing $\tau$ such that high priority is given to columns with low MAD values. There exits many number of functions that can map MAD values to $\tau_i$ to achieve the same purpose. i.e. $\tau_i$ can be designed based on MAD to introduce priority to columns in large number of ways. One direction for future exploration on this work is to explore if it is possible to select the best set of scaling factors given a noisy patch matrix (say by using a learning algorithm).

In the method 2 described in Section 3.2, simple averaging over reliable pixels is used to denoise the low structured patches. This is done to explore the behavior of matrix completion algorithm with respect to structural content
of patches. It is expected that employing a more sophisticated approach for denoising low structured patches will improve the performance compared to simple averaging case. Also, the behavior and limitations of matrix completion with respect to relative structure between noise and image can be further explored by conducting experiments in presence of structured noise such as periodic noise.

Figures 5.1 and 5.2 show the similarity plots for two randomly chosen reconstructed patch matrices. The plots on the left side show the similarity of each reconstructed column in the patch matrix with adaptive median filtered reference patch. The X axis represents the column number and the Y axis represents the MAD values. Plots on the right side show the similarity of each reconstructed column with ground truth reference patch. The reconstructed patch matrix has $Mm$ columns which are similar to the reference patch and any of these columns can be picked to generate the reconstructed video frame. The question is which column should be picked to obtain the best possible performance for the algorithm? In our experiments with the method 1 of the proposed algorithm, it was observed that instead of picking a particular column from all patch matrices, picking the column from each patch matrix which is closest to ground truth reference patch yields considerable improvement in PSNR values of the reconstructed frames. But this is not practically possible as ground truth is not available in real world scenario. Therefore, one direction for future work is to explore if it is possible to pick best reconstructed column without the use of ground truth. It is important to note that in method 1 of proposed approach though the patches are arranged in order according to their similarity with the adaptive median filtered reference patch, picking the first reconstructed column (patch most similar to adaptive median
filtered reference patch) from each patch matrix will not result in best possible performance. This is because Figures 5.1 and 5.2 show that the patch which is most similar to adaptive median filtered reference patch need not necessarily be the most similar one to the ground truth reference patch. It would also be interesting to observe similar plots after introducing method 3 of the proposed approach to understand how prioritizing of columns affects the similarity of the reconstructed columns from the adaptive median filtered reference patch and ground truth reference patch. Such understanding might also give insights to design the priorities in a better way.

![Similarity plots for randomly chosen reconstructed patch matrix and adaptive median filtered output](image)

**Figure 5.1:** Closeness plots for randomly chosen reconstructed patch matrix # 1 with Adaptive median filtered output and ground truth.

**Figure 5.3** shows singular value plots for two randomly chosen patch matrices. Green curve indicates singular values of the denoised patch matrix by using the matrix completion framework, blue curve indicates the singular values of the corresponding patch matrix formed from ground truth and the red curve shows the singular values of the patch matrix formed from adaptive median filtered data. It can be seen that the patch matrix formed from adaptive median filtered data has high singular values indicating the presence of noise. It was observed that in most of the cases, the denoised patch matrices
Figure 5.2: Closeness plots for randomly chosen reconstructed patch matrix # 2 with Adaptive median filtered output and ground truth.

Figure 5.3: Singular values plots of two randomly chosen patch matrices.

had lower singular values compared with the patch matrices formed from the ground truth. This might indicate loss of useful image details during the denoising process. For future work it is useful to extend this analysis on singular values of patch matrices to get better insights into possible improvements for the algorithm.

Few experiments were done to explore the extension of this algorithm to color videos. Patch matching can be done either by picking patches from same locations from all the three RGB channels by computing single similarity score over all the three channels or patch matching can be done by treating the
three RGB channels independent of each other. Our observation is that the former case yielded better performance, though more number of experiments are needed to strongly conclude on this observation. On the other hand, in the BM3D [6] approach for color images the RGB image is transformed into YUV space and only the Y channel is used for patch matching and the patches from corresponding locations are picked from U and V channels. This method is followed in [6] for the following reasons: The luminance channel contains most of the structural information of the image and it is assumed that if two patches are similar in the Y channel, then their corresponding chrominance values will also be similar. Moreover, performing patch matching only on one of the three channels results in high amount of computational savings. In future work, it is useful to explore color extension for the proposed algorithm based on the above discussion.
REFERENCES


[68] \url{http://media.xiph.org/video/derf/}.
