UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

IMAGE AND VIDEO ENHANCEMENT USING SPARSE CODING, BELIEF PROPAGATION AND MATRIX COMPLETION

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

NAFISE BARZIGAR
Norman, Oklahoma
2014

IMAGE AND VIDEO ENHANCEMENT USING SPARSE CODING, BELIEF PROPAGATION AND MATRIX COMPLETION

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

Dr. Samuel Cheng, Chair

Dr. Pramode Verma, Co-chair

Dr. Jerzy Bodurka

Dr. James Sluss

Dr. William Ray

© Copyright by NAFISE BARZIGAR 2014
All rights reserved.

Acknowledgements

I would never have been able to finish my dissertation without the guidance of my committee members, help from friends, and support from my family and husband.

I would like to express my deepest gratitude to my advisor, Dr. Samuel Cheng, for his great guidance, caring, patience, and providing me with an excellent atmosphere for doing research. The joy and enthusiasm he has for his research was contagious and motivational for me, even during tough times in the Ph.D. pursuit. And special thanks to Dr. Pramode Verma for his trust, mentoring, support, and providing me the opportunities to assist in teaching telecommunications classes and managing the telecommunication lab. Thank you. I appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating.

To the members of my doctoral committee Dr. James Sluss, Dr. Jerzy Bodurka, and Dr. William Ray, I express my gratitude for their advice and suggestions at the beginning of my research work. Particular thanks to Dr. Jerzy Bodurka for his kind support, encouragement, and his faith in me. I would like to thank the OU Tulsa Graduate College Dean, Dr. William Ray, for providing me the opportunities to assist in teaching the "Stochastic Processes" course.

I would like to thank Mrs. Renee Wagenblatt for her kind support, help and patiently editing my writing. I am grateful for the opportunity to meet talented and supportive researchers and students at OU-Tulsa. I extend a big "thank you" to our OUSAM lab group, specially Lijuan, Shuang, Feng, and Anusha, and to Dr. Bodurka's expert lab members at Laureate Institute for Brain Research (LIBR).

I express my deepest gratitude to many friends who have meant so much to me and who have provided much encouragement, especially Atie, Somaye, Saeed, Alireza, Mohammad, Mahnaz, Soode, Amirreza, Aziz, Hajagha, Parisa, Athena, Sepideh, Mahsa, Fahime, my "Safora" classmates, Nilo, Andy, Mrs. Borbor, Mrs. Fallah, Mr. Ahmadinia, Mr. Steve Trinkle, and Dr. Salimian.

I especially thank my beloved mom, dad, sisters, and brothers. My hard-working parents have sacrificed their lives for my siblings and myself and provided unconditional love and care. I love them so much, and I would not have made it this far without them. Special thanks to Amin, my husband as well as his wonderful family who all have been supportive and caring.

The best outcome from these past five years is finding my best friend, soul-mate, and husband, Amin. I married the best person out there for me. There are no words to convey how much I love him. Amin has been a true and great supporter and has unconditionally loved me during my good and bad times.

Table of Contents

List of Tables

List of Figures

Abstract

Super resolution as an exciting application in image processing was studied widely in the literature. This dessertion presents new approaches to video super resolution, based on sparse coding and belief propagation. First, find candidate match pixels on multiple frames using sparse coding and belief propagation. Second, incorporate information from these candidate pixels with weights computed using the Nonlocal-Means (NLM) method in the first approach or using SCoBeP method in the second approach. The effectiveness of the proposed methods is demonstrated for both synthetic and real video sequences in the experiment section. In addition, the experimental results show that my models are naturally robust in handling super resolution on video sequences affected by scene motions and/or small camera motions.

Moreover, in this dissertation, I describe a denoising method using low-rank matrix completion. In the proposed denoising approach, I present a patch-based video denoising algorithm by grouping similar patches and then formulating the problem of removing noise using a decomposition approach for low-rank matrix completion. Experiments show that the proposed approach robustly removes mixed noise such as impulsive noise, Poisson noise, and Gaussian noise from any natural noisy video. Moreover, my approach outperforms state-of-the-art denoising techniques such as VBM3D and 3DWTF in terms of both time and quality. My technique also achieves significant improvement over time against other matrix completion methods.

CHAPTER 1

INTRODUCTION

Today, with advances in sensor design, the given image is relatively clean for digital cameras, but it remains noisy and blurry for low-grade and mobile phone cameras. Thus the super resolution problem is still of acute importance, and I present a novel learned image model method, which outperforms the-state-of-the-art, super resolution task on real and synthetic low sequences. Sparse representation techniques are beginning to show significant impact on image processing [17–20]. Also, the results of these methods illustrate that the sparse representation can be correctly recovered from the downsampled signals.

In addition, Images from various modalities need to be denoised as a pre-processing step for many planning, navigation, detection, data-fusion and visualization tasks in medical applications [21, 22]. Video sequences are often corrupted by noise during acquisition or transmission. CT slices are often corrupted by noise during acquisition or transmission. Noises are added in the CT slices during acquisition by CT scanner sensors [23]. Some noise sources located in camera hardware became active during image acquisition under some lighting conditions. Other noise sources are over transmission channels. Most video denoising algorithms proposed in the literature assume additive white Gaussian noise, which can be categorized into pixel domain and transform domain methods. However, I consider Impulsive/Poisson/Gaussian noise in my work and will show how robust my denoising method is. The goal of my denoising method is to keep only the reliable pixels and get rid of all other un-

reliable pixels I find as noise. For each patch in the reference frame, I find the similar patches in the other frames using a block matching algorithm. The found matches will be vectorized and then stacked into a matrix. The reliable pixel values in the matrix are between the mean ± standard deviation of all elements in the same row. The main step will be done by applying the matrix completion approach [24] on the incomplete matrix. The output of matrix completion is a noise free full matrix. Then, the average value of each row in the full matrix can recover the denoised patch. Repeating the same procedure for all blocks of reference frame can build a denoised frame.

1.1 Classic Super Resolution

Super resolution reconstruction attempts to estimate one high quality result $\mathcal{X}$ out of several lower resolution and potentially noisy images $\left\{\mathcal{Y}_t\right\}_{t=1}^{T}$. A popular way to model LR images $\left\{\mathcal{Y}_t\right\}_{t=1}^{T}$ from a pseudo HR image $\mathcal{X}$ is through a sequence of operations including geometrical wrapping F_t, linear space-invariant blurring H, spatial decimation D_t, and zero-mean white Gaussian noise ε_t. The model can be summarized with the following equation:

$$\mathcal{Y}_t = DHF_t\mathcal{X} + \varepsilon_t, \quad t = 1, 2, 3, ..., T, \tag{1.1}$$

where T is the number of available LR frames. Note that I assume H and D are identical for all frames in the sequence.

The recovery of $\mathcal{X}$ from $\left\{\mathcal{Y}_t\right\}_{t=1}^{T}$ using the above mentioned model requires us to solve an inverse problem. The maximum *a posteriori* probability estimate of $\mathcal{X}$

can be obtained by minimizing the following objective function with respect to $\mathcal{X}$:

$$\epsilon^2_{MAP}(\mathcal{X}) = \frac{1}{2}\sum_{t=1}^{T} \|DHF_t\mathcal{X} - \mathcal{Y}_t\|_2^2 + \lambda \cdot TV(\mathcal{X}), \tag{1.2}$$

where the first summation term ensures that the projections of the estimate $\mathcal{X}$ looks similar to the LR images and the second term, $\lambda \cdot TV(\mathcal{X})$, acts as a prior and helps to remove artifacts from the final solution and improves the rate of convergence [25].

Since H and F_t are space-invariant operators in (1.2), they can be considered as block circulant matrices (assuming a cyclic boundary treatment) that they commute [7, 26]. This allows one to solve (1.2) in the following two steps [2, 7, 26, 27]. First, minimize the following penalty function with respect to $\mathcal{Z}$:

$$\epsilon^2_{ML}(\mathcal{Z}) = \frac{1}{2}\sum_{t=1}^{T} \|DF_t\mathcal{Z} - \mathcal{Y}_t\|_2^2 , \tag{1.3}$$

where $\mathcal{Z}$ can be interpreted as a blurred version of the HR frame $\mathcal{X}$ and thus should be approximately equal to $H\mathcal{X}$. This step estimates the blurry high-resolution image $\mathcal{Z}$ from the collection of the low resolution images $\mathcal{Y}$. For a more general case with multiple input patches, I will modify ϵ^2_{ML} in (1.3) to (2.7) as shown in Section 1.3.

Then, impose the constraint of the closeness of $\mathcal{Z}$ and $H\mathcal{X}$ and incorporate back the regularization term to obtain the following objective function:

$$\epsilon^2_{MAP}(\mathcal{X}) = \|H\mathcal{X} - \mathcal{Z}\|_2^2 + \lambda \cdot TV(\mathcal{X}), \tag{1.4}$$

where $\mathcal{X}$ can be obtained through minimizing (2.8) in Section 2.3.3. Since H is usually singular, this stage is an under-determined problem and needs regularization (see [28, 29] for more detail).

In summary, one can break the minimization problem in (1.2) in two steps:

1. compute a blurred version of HR $\mathcal{Z}$ by minimizing (1.3).

2. estimate the deblurred frame $\mathcal{X}$ from the found blurred HR $\mathcal{Z}$ in step 1.

As the second step only involves the classic deblurring problem, many potential techniques can be applied here. In my proposed approaches, I adopt the *Adaptive Kernel Total Variation* (AKTV) regularized locally-adaptive kernel regression in a variational approach developed by Takeda *et al.* [30], which can simultaneously interpolate and deblur in one integrated step. However, one can generally incorporate any deblurring techniques into the proposed method.

1.2 Background of NLM Filter

The whole entity of a self-similar object is exactly like or similar to a part of itself. As a consequence, parts of it can show the same statistical properties at many scales. Based on this presumption, non-local self-similarity techniques have been widely used in areas such as image denoising [13, 31], texture synthesis [32], and super resolution [2, 5, 27]. For example, the NLM filter, which is based on the assumption that image content is likely to repeat itself within its neighborhood, is applied successfully to image denoising. Its key idea is that one can denoise a pixel $[i, j]$ by performing weighted average around its neighborhood [31]. More precisely, denote $\mathcal{Y}[i, j]$ as the intensity of pixel $[i, j]$, then the intensity of denoised pixel $[q, l]$, $\mathcal{X}[q, l]$, can be written as

$$\mathcal{X}[q,l] = \frac{\sum_{(i,j)\in\mathcal{N}(q,l)} \mathcal{W}[i,j,q,l]\mathcal{Y}[i,j]}{\sum_{(i,j)\in\mathcal{N}(q,l)} \mathcal{W}[i,j,q,l]}, \tag{1.5}$$

where $\mathcal{N}(q, l)$ denotes a neighborhood around pixel $[q, l]$, and $\mathcal{W}[i, j, q, l]$ is a weight that is decreased with the distances between pixels $[i, j]$ and $[q, l]$, and increased with the similarity of the patches centering at the two pixels. The formula in (1.5) describes the NLM filter where denoising each pixel is done by averaging all pixels

in its neighborhood. However, this averaging is not performed blindly and instead each pixel in the relevant neighborhood is assigned a weight which corresponds to the probability that the pixel $\mathcal{Y}[i,j]$ and the pixel $\mathcal{X}[q,l]$, prior to the additive noise degradation, had the same value.

NLM filter computes the weight based on both radiometric proximity and geometric proximity between the pixels. The radiometric part is estimated by computing the Euclidean distance between two image patches centered around these two included pixels. Let us consider $R_{q,l}$ as the matrix that extracts a patch with fixed and predefined size of $g \times g$ pixels at its position $[q,l]$ in the image. Hence, $R_{q,l}\mathcal{Y}$ is equivalent to the $g \times g$ matrix representing the extracted patch of $\mathcal{Y}$ at position of $[q,l]$. As NLM estimation is a zero-order regression, only the zero-order basis is used for estimation. Therefore, the NLM weights look like

$$\begin{aligned}\mathcal{W}[i,j,q,l] =& exp\left\{-\frac{\|R_{q,l}\mathcal{Y} - R_{i,j}\mathcal{Y}\|_2^2}{2\sigma^2}\right\} \\ &\times f(\sqrt{(q-i)^2+(l-j)^2}),\end{aligned} \tag{1.6}$$

where σ manages the effects of radiometric differences between two patches and when the intensities of the two patches are far away, the weight becomes very small and thus can be ignored. Whereas the function f is in charge of the geometric distance, and it may have many forms such as a Gaussian, a box function, or a constant [2, 27]. Since there are various other ways to choose the weights in (1.5), in this chapter I will restrict my choice to SCoBeP [1] and NLM as described in Sections 2.3.2 and 2.3.3.

1.3 NLM for Super Resolution

Since self-similarities exist in most natural images, one can also use the NLM algorithm to take advantage the non-local similarity property of natural images in the

superresolution problem.

In essence, one may extract a target patch information from multiple patches instead of one patch per each LR frame. This allows us to modify ϵ^2_{ML} in (1.3) instead to [2, 27]

$$\begin{aligned}\epsilon^2_{ML}(\mathcal{Z}) &= \frac{1}{2}\sum_{t=1}^{T}\sum_{[q,l]\in\mathcal{I}}\sum_{[ri,rj]\in\mathcal{N}(q,l)}\mathcal{W}[i,j,q,l,t] \\ &\quad\times\left\|DR^H_{q,l}\mathcal{Z}-R^L_{i,j}\mathcal{Y}_t\right\|^2_2,\end{aligned} \tag{1.7}$$

where $\mathcal{I}$ is the set of pixel coordinates of the entire frame $\mathcal{X}$, $\mathcal{N}(q,l)$ is a neighborhood of the pixel $[q,l]$, and $\mathcal{W}[i,j,q,l,t]$ can be interpreted as a weight that the pixel $[q,l]$ in the reference frame should be mapped to the pixel $[i,j]$ in the t^{th} LR frame $\mathcal{Y}_t$. $R^H_{q,l}$ and $R^L_{i,j}$ are defined as the HR and LR patch extraction operators respectively, where the size of the extracted patches are related to the resolution ratio r as follows. Let the size of patches extracted by $R^L_{i,j}$ and $R^H_{q,l}$ be $g\times g$ and $k\times k$, respectively. I have $k=r(g-1)+1$. Note that k is not set precisely as rg to avoid the need of extrapolation. The detail in computing $\mathcal{W}$ will be deferred to Section 2.3.2 and 2.3.3.

As for the first step, one can show that the optimum $\mathcal{Z}$ can be computed as [2]

$$\mathcal{Z}[q,l]=\frac{\sum_{t=1}^{T}\sum_{[ri,rj]\in\mathcal{N}(q,l)}\mathcal{W}[i,j,q,l,t]\mathcal{Y}_t[i,j]}{\sum_{t=1}^{T}\sum_{[ri,rj]\in\mathcal{N}(q,l)}\mathcal{W}[i,j,q,l,t]}. \tag{1.8}$$

1.4 Contribution of My Dissertation

The contributions of the proposed approach can be summarized as follows:

(i) Decomposing each reference image patch into a linear combination of a few

elements from a basis set, which increases the expressiveness of basis patches that allows the reconstruction of edges with high fidelity, and avoids the need of accurate motion estimation.

(*ii*) Identifying the best similar matches based on the local and the geometric characteristics using factor graph, which efficiently trade off both characteristics optimally.

(*iii*) Incorporating SCoBeP to efficiently sift through pixels from all LR images and thus resulting much more accurate non-local candidates for subsequent estimation.

(*iv*) Demonstrating competitive performance of the proposed methods with both synthetic and real video sequences.

(*v*) Breaking the locality constraint in the conventional restoration methods using NLM method, which provides similarity by exploiting the redundancy between overlapping patches.

(*vi*) Exploiting both SCoBeP output and the NLM technique in calculating weights to facilitate tradeoff between computational complexity and performance.

(*vii*) Incorporating belief propagation into sparse coding, which increases the effect of each neighboring LR pixel on the candidate points.

(*viii*) relying on basic vector operations instead of utilizing SVD in video denoising, which the approach is immediately applicable to matrices of any field (including finite field matrices).

The rest of my dissertation is structured as follows. I give a brief summary of related work and review the background of super resolution and NLM filter in the next chapter. In Section 2.3, I introduce my proposed methods: SCoBeP-SR and SCoBeP-NLM. Implementation issues are also presented in detail in this section. Section 3.3.3 presents the experimental results and compares my results with that of the existing super resolution methods. Moreover, I will introduce the concept

of my video deoising method using block matching filtering and matrix completion method in Chapter 3. I further show my simulation results in Section 3.3.1. Finally, Chapter 4 concludes the work has been done throughout this dissertation.

CHAPTER 2

A SUPER RESOLUTION METHOD FOR VIDEO WITH CAMERA AND OBJECT MOTION USING SCoBeP

2.1 Introduction

Super resolution tries to combine several low resolution (LR) images from a scene and produces one higher resolution image with better optical resolution. This is an inverse problem that is commonly tackled by integrating denoising, deblurring, and upsampling.

Fig. 2.1 illustrates this inverse process and presents how the LR sequence may be modeled using an original higher resolution frame. During imaging, the blurring effect can be modeled by the optical point spread function (PSF). The scene may then be warped due to camera or object motion. Moreover, the motion effect might not be the same for all frames in the sequence. A fixed decimation operator is typically used to model the effect of sampling by the image sensor. The operator is characterized by the resolution ratio between the original higher resolution frame and the LR sequence. The noise, which in most applications assumed to be white i.i.d. Gaussian, is added to the LR frames. The outcome of the super resolution reconstruction problem depends on the involved operators and noise characteristics of the above mentioned model.

A wide variety of super resolution methods have been studied in the last two decades [2,4,5,7,27,33–37]. Huang and Tsai were the first to address the multiframe

super resolution problem using a frequency domain approach that works for band limited and noise-free images [38]. Later, it was extended by others, such as Kim *et al.* who proposed a super resolution method on noisy and blurred images [39]. Pleg and Irani [33] also suggested a different approach for the super resolution problem based on the iterative backprojection (IBP) method adopted from computer aided tomography (CAT). Recently, an iterative multiframe super resolution method was presented in [4] that relied on extending the steerable kernel method in space-time. However, the approach assumes the input frames only contain smooth textures. Also, it has difficulties to estimate a pixel-wise motion in regions with the larger motion [40].

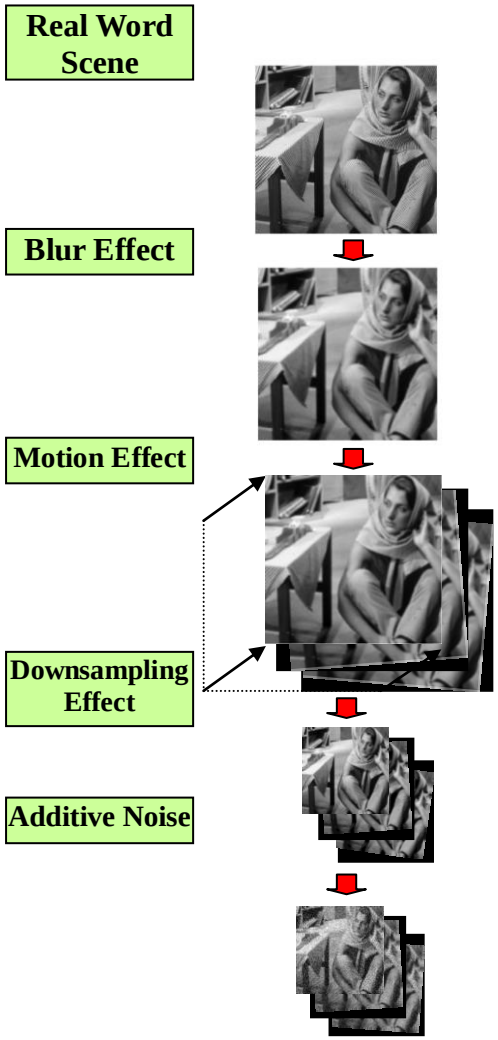


Figure 2.1: A general model of multi-frame super resolution.

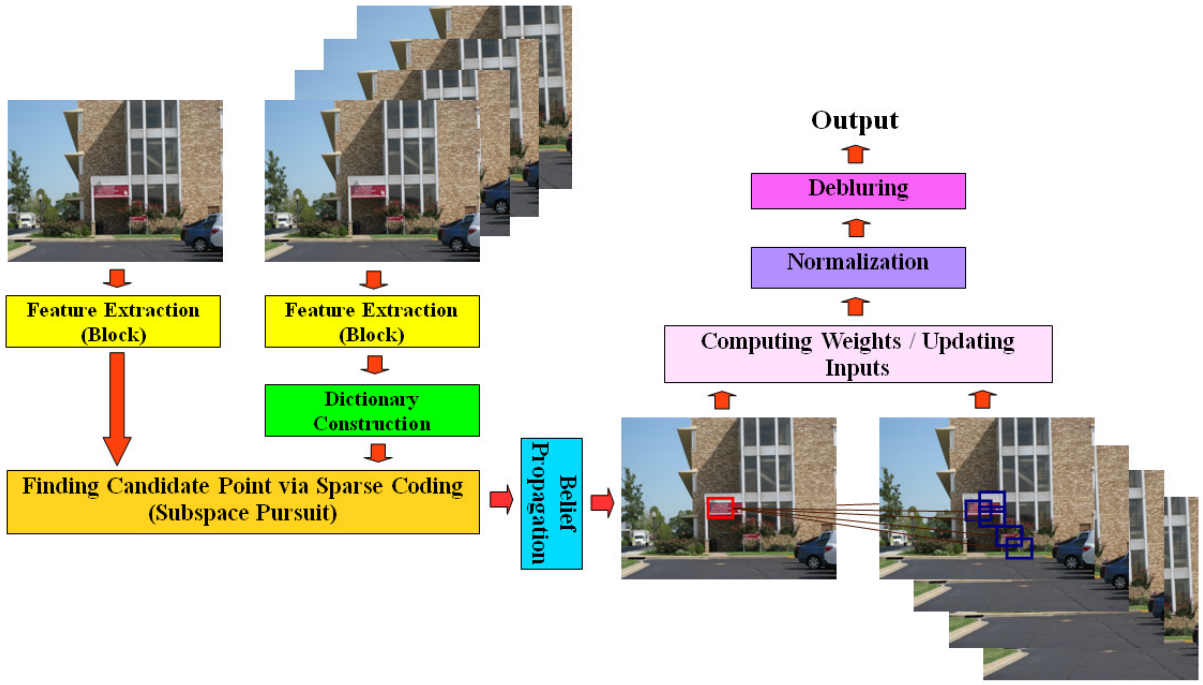


Figure 2.2: Block diagram representation of my models.

In dealing with camera position variation, a few attempts have been made through a global motion model [7, 35, 38, 41]. Bonchev and Alexiev suggested a method of super resolution that used the information from several LR frames by controlling the camera position in frequency domain when taking frames [35]. Also, in [41] a maximum a posteriori (MAP) was adopted to provide coarse estimates of rotation and translation between images. The authors claimed that such estimation step provided enough accuracy to effectively remove the effect of the rotational and coarse (super-pixel) translational motion between the images. Although that algorithm incorporates smoothness priors as a constraint to reconstruct the HR images, using these smoothness priors might not lead to smooth results [42]. A number of super resolution approaches using Total Variation (TV) regularization terms have been explored in the last decade, e.g., the approach by Mitzel *et al.* in [36], where their method is not restricted to any particular motion model and they do not assume that the motion is known. In another work, Farsiu *et al.* suggested a multiframe super resolution method by applying constraints on the L_1 norm of both the bilateral TV regularization term and the data fusion term to produce a sharp, high resolution (HR) image [7]. The researchers also registered the LR images with respect to a reference frame before starting the super resolution process. Liu and Sun in [40] proposed a Bayesian framework for adaptive video super resolution that deals with video super resolution by also simultaneously estimating underlying large motion. Moreover, they jointly estimated the flow field and the noise level in a coarse-to-fine manner on a Gaussian image pyramid using the HR image and the blur kernel.

In this work, I focus on video frames that suffer from non-homogeneous noise, atmosphere or camera blur, motion and down-sampling effects. Also, as real videos can be taken from both fixed or movable cameras, I also consider frames affected by scene motions and/or small camera motions. Note that as I will see in the coming

sections, the approach I introduced here works well for such scenario. My method is complementary to approaches such as [40], which considers the superresolution of sequences with larger camera motions but little or no scene motions.

I propose to solve the super resolution problem using a novel framework taking advantage of two recently developed techniques: SCoBeP [1] and Nonlocal-Means (NLM) [31]. My approaches are based on the concept of Sparse Coding and Belief Propagation (SCoBeP) which is earlier introduced in [1] for 2-D signals (images). It turns out that the technique is well-suited for super resolution of video (a 3-D signal) as I explore in this chapter.

As a summary of my approaches, I first build an overcomplete dictionary out of all block features of LR frames as shown in Fig. 2.2. Different from [43], I am not generating HR/LR patch pairs from the frames by exploiting self-similarities. For each pixel of the initial estimate of the HR frame, I then select a set of candidate pixels out of the constructed dictionary using sparse coding [44]. The match score of each candidate pixel will be evaluated taking both local and neighboring information into account using belief propagation [45]. The best matches will be selected as the candidates with the highest scores. An occluded pixel or any pixel not covered by the LR frames is likely to be identified since the match scores in this case will be significantly smaller than a typical maximum score when a match pixel actually exists. Finally, in my first proposed method, the NLM approach exploits similarity in patches around candidate pixels to average out the noise among similar patches [2] and in my second proposed method, a pixel is reconstruct from multiple candidate pixels with the weights extracted directly from the output of SCoBeP.

In the experiment section, I also illustrate that the proposed methods can perform well on real LR videos (besides "phantom" LR videos generated artificially) and can reconstruct image edges with high fidelity. Although the NLM filtering has shown great potentials for image denoising and superresolution [31], it is only

Table 2.1: Summary of Notation

Notation	Description
$\mathcal{X}$	the reconstructed HR frame
$\mathcal{Y}_t$	the LR frame at t
y_t	the interpolated LR frame at t
T	the number of available LR frames
$\mathcal{Z}$	the blurred version of the reconstructed HR frame
$\mathcal{N}(q,l)$	a neighborhood around pixel $[q,l]$
R_{ql}	the matrix that extracts a patch centered around pixel $[q,l]$
r	the resolution (magnification) ratio
n	the number of candidate pixels
X_{ql}	the vectorized patch at pixel $[q,l]$ in the reference frame
D_t	the dictionary constructed from the t^{th} LR frame $\mathcal{Y}_t$
α_{qlt}	the sparse representation vector of a patch centered around pixel $[q,l]$ in t^{th} LR frame
Y_{qlt}	the vectorized patch at pixel $[q,l]$ in the t^{th} LR frame $\mathcal{Y}_t$
$\mathcal{W}[i,j,q,l,b,t]$	the weight mapping from a pixel in the reference frame to b^{th} candidate pixel in the t^{th} interpolated LR frame y_t
cp	an $n \times 2$ matrix storing the locations of the candidate pixels
ρ_{qlt}	the prior probability of pixel $[q,l]$ in the reference frame mapping to the b^{th} candidate pixel in t^{th} interpolated LR frame

effective when a reference patch can be identified to accurately represent the targeted patch. Moreover, NLM approaches generally have very high computational complexity. I will show in this chapter reference patches can be effectively found by SCoBeP. Further, I will also show that the NLM step may be skipped completely (as demonstrated in the second method) with only a small performance penalty but a significant (about three times) speed-up.

2.2 Related Work And Background

Sparse representation [19] and self-similar-based techniques [5,27] have been used in super resolution in recent years. In this section, I will review how some of the recent works use these techniques in recovering the downsampled signals and computing the similarity of image patches.

In [19,20], a large set (of the order of a hundred thousand) of patches randomly sampled from natural images to train an LR and a HR dictionaries. The main idea consists of seeking in the database for a sparse representation of each patch of the LR input, followed by using this representation to generate the HR output. Yang *et al.* [43] proposed a super-resolution method that exploits self-similarities and group structural constraints of image patches using only one single input frame. In this algorithm, the patch self-similarity within the image is exploited and the group sparsity then will be introduced for better regularization in the reconstruction process. Another recent example based on an enhanced sparse representation in transform domain is block-matching 3-D filter (BM3D) [18], which uses a block matching technique to find a set of similar 2D blocks. Danielyan *et al.* have extended (BM3D) in [37] for image and video super resolution. They produce a sparse representation of the true signal in the transform domain to exploit the similarity among the blocks. In contrast to the sparse representation approaches discussed above where they use information from only one corresponding pixel per LR frame to reconstruct a target pixel, my first approach incorporates the NLM method to take advantage information from multiple matched pixels for the reconstruction.

I now turn to a discussion of certain works associated with self-similar-based technique. Plenty of works have emerged lately based on self-similarity for natural image and video processing. The self-similarity property shows that the image content desires to repeat itself within some neighborhoods. Non local self-similarity

has been effectively applied to many aspects of image processing [17, 27, 32]. Following this insight, Buades *et al.* used this approach in image denoising, which is known as the NLM method [31]. The NLM method was used also in image restoration explicitly exploits self-similarities in natural images [27, 31]. Liu and Freeman in [46] proposed a video denoising approach to use an approximate k-nearest neighbor (AKNN) algorithm to approximately but rapidly seek the most similar patches for a given video. As pointed out in [47], [46] takes into account only similar blocks for a given video and thus could be classified as a "closest structure" method, while one can call the original NLM method [31] a "closest space" method in the sense that it uses only closest blocks in a small window. Moreover, Marial *et al.* in [17] extended the NLM method in denoising and demosaicking using the idea that similar patches have similar sparsity patterns. Also, in [5], Zhang *et al.* proposed a non-local kernel regression method for image and video super resolution, which exploits both non-local self-similarity and local structural regularity in a single model. Distinct from the local kernel regression, the NLM method estimates the value of a pixel from all possible patches collected from a search area, and breaks the locality constraint in the restoration algorithms. Protter *et al.* [2] generalized this denoising method to perform multiframe super resolution reconstruction with no explicit motion estimation. In that work, computing the similarity of video frame patches resulted in probabilistic estimates of motion.

Prior works have been limited to block matching in restricted neighborhoods. These neighborhoods determine the candidate matches of target pixels and thus have a significant impact on SR performance. However, they have always been assigned with limited sizes and regular shapes (e.g., as rectangular blocks) in prior works and hence often do not include the best match patches. Due to this poor block matching, the prior techniques could suffer from block artifacts in some test cases [2, 48].

In contrast, the advantage of SCoBeP registration is that the chosen candidates will have better "diversity" when compared with the AKNN or even the *exact* K-nearest neighobor (KNN) approach (see Fig. 2.3). This originates from the induced orthogonality of the patches when sparsity is imposed in the solution. In the KNN case, when a smooth patch is incorrectly matched to a patch, the next best "matches" are likely around the neighborhood of the wrong patch and it ends up incorrect matching for all patches. The better "diversity" actually affords SCoBeP a larger search window compared with other registration without sacrificing the robustness of the approach [1]. In this chapter, I take advantage of SCoBeP to select from each LR frame a set of candidate pixels which are likely to be most similar to the target pixel. As a result, for each pixel and per LR frame, I have an irregular neighborhood that can include any pixel in the frame. This significantly improves the block matching performance that directly links to the overall SR performance.

2.3 Proposed Method

The key to apply NLM to super resolution efficiently depends on how I can identify the appropriate neighboring set ($\mathcal{N}(q,l)$) for each pixel and also how I can choose the appropriate weighting function. In particular, the neighborhood $\mathcal{N}(q,l)$ has significant effect on the performance of the NLM filter. The neighborhood should be sufficiently large to take advantage "non-local" benefit of the algorithm. However, this also significantly increases the complexity of the algorithm.

Ideally, I would like the neighborhood set $\mathcal{N}(q,l)$ to cover the entire frame. That is, to allow each pixel to take into account information from any pixel of every LR frame and let the weight variable $\mathcal{W}[i,j,q,l,t]$ to take care of the significance of the contribution. This, of course, will lead to unrealistic computational load if I blindly look into every pixel of every LR frame. What I need is an intelligent preprocessing step to identify pixels that are likely to provide useful information to the target pixel

no matter where the formers locate. The described problem above is closely related to image registration, and I want to look for multiple matches from each reference frame (i.e., an LR frame in this case).

While many registration methods can be used, I chose to use SCoBeP [1] for the aforementioned purpose as SCoBeP naturally identifies multiple matched pixels and returns the corresponding match scores as needed in this application. In summary, for each pixel in the initial estimate of the HR frame, I use SCoBeP to select from each LR frame a set of n candidate pixels which are likely to be most similar to the target pixel. The similarity between a target pixel and a candidate pixel, which will be characterized by a weight, will be used by the SCoBeP method as to be described in Section 2.3.2 or the NLM filter as to be described in Section 2.3.3.

For the rest of this section, I will review the SCoBeP registration technique in the context of superresolution and provide the implementation details of my proposed super resolution methods, SCoBeP-SR and SCoBeP-NLM, which are based on sparse coding, belief propagation and NLM. I divide the super resolution process into two steps as shown in Sections 2.3.1 and 2.3.2 or 2.3.3.

2.3.1 Use SCoBeP [1] to compute the locations and prior probabilities of candidate pixels

The proposed method described here is inspired by my recent work, SCoBeP [1]. First, I extract the features from all interpolated LR frames $\left\{ y_t \right\}_{t=1}^{T}$ and the reference frame $\mathcal{X}$. To extract the features, I consider a patch of size $(2h+1)^2$ containing neighboring pixels around each pixel on the reference and LR frames, where h is a positive integer. For each pixel $[p, q]$ in the reference frame $\mathcal{X}$, I vectorized the patch centered around the pixel $[p, q]$ to a feature vector $X_{ql} \in \mathbb{R}^{S\times 1}$, where $S = (2h+1)^2$.

In this dissertation, I focus myself on only using block features even though the proposed approach can generally be applied to other features (such as SIFT-

features). Thus, each feature considered here is essentially a vectorized block centered around a pixel in a frame.

Second, to match the extracted features of the reference frame to the corresponding extracted features of the t^{th} interpolated LR frame y_t, I create a dictionary which contains all feature vectors of y_t. More precisely, a dictionary $D_t \in \mathbb{R}^{S \times MN}$ (M and N are the height and width of the interpolated LR frame y_t minus h from each side) is constructed with all possible vector $Y_{qlt} \in \mathbb{R}^{S \times 1}$ as D_t's column vectors, where Y_{qlt} is created in the same manner as X_{ql} but from the t^{th} interpolated LR frame y_t instead. Thus, I can write D_t as

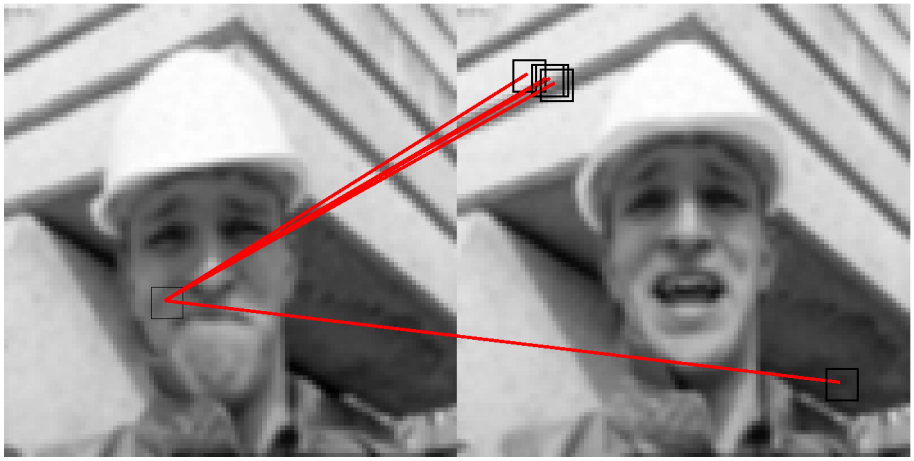
$$D_t = [Y_{1,1,t} Y_{1,2,t} \cdots Y_{1,N,t} Y_{2,1,t} \cdots Y_{M,N,t}]. \tag{2.1}$$

I then normalize dictionary D_t to guarantee the norm of each feature vector to be 1.

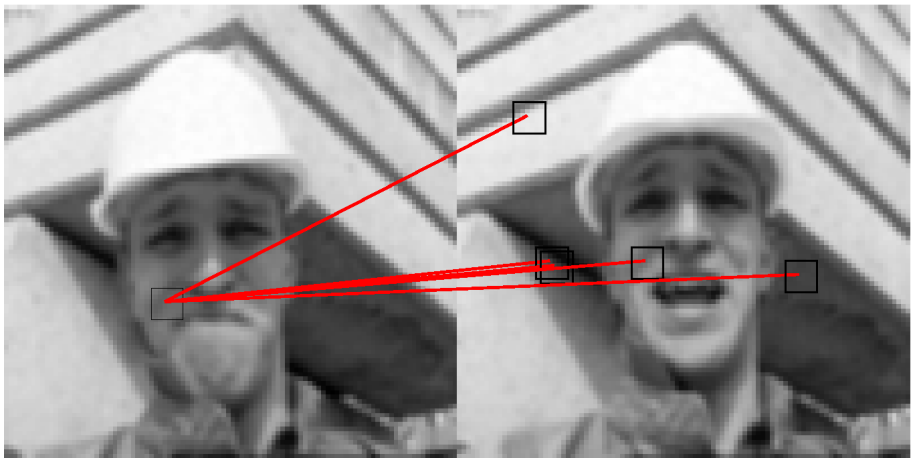
Third, to identify the candidate Y_{qlt} that looks most similar to the input X_{ql} in the reference frame, I apply sparse coding to each extracted features of the reference frame. Sparse coding will reconstruct a reference patch at pixel $[q, l]$ as a linear combination of LR patches. Denote α_{qlt} as the sparse vector where each element corresponds to a coefficient in this combination. Note that α_{qlt} should be sparse, i.e., it should be 0 for most coefficients.

Mathematically, I try to solve the following sparse coding problem of finding the most sparse coefficient vector α_{qlt} such that

$$X_{ql} = D_t \alpha_{qlt}. \tag{2.2}$$



(a)



(b)

Figure 2.3: Candidate points obtained by KNN and sparse coding. The images in (a) shows that KNN tends to result in candidate points with poor diversity. And thus it can easily miss including the true corresponding point as one of its candidate points. In contrast, the images in (b) show that the candidate points of sparse coding tend to diversify and thus is more likely to include the true corresponding point.

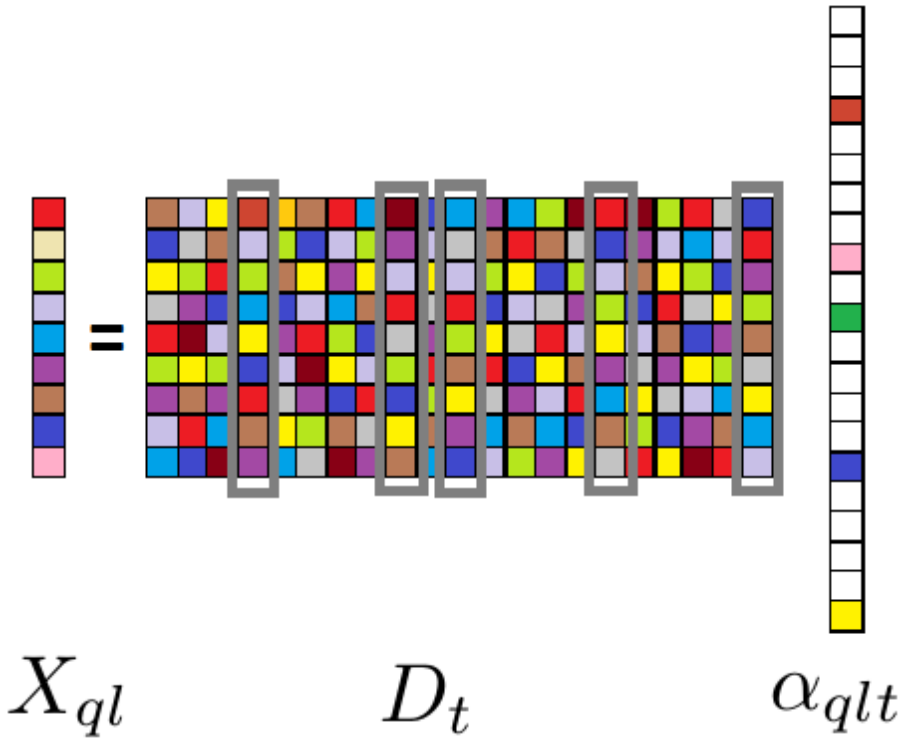


Figure 2.4: Sparse representation of a feature vector X_{ql} with a dictionary D_t: α_{qlt} as a sparse vector constructs the feature vector X_{ql} using a few columns (highlighted in gray) of dictionary D_t.

The sparse vector α_{qlt} is the representation of X_{ql}, which has few number of non-zeros coefficients. Thus, α_{qlt} describes how to construct X_{ql} as a linear combination of a few columns (also referred to as atoms) in D_t. The locations of the nonzero coefficients in α_{qlt} specifically point out which Y_{ql} in the dictionary D_t is used to build X_{ql} and the values of the non-zero coefficients in α_{qlt} show what "portions" thereof are used for its construction. As shown in Fig. 2.4, one expected that most of the coefficients in α_{qlt} obtained by sparse coding are zero, and the bases of those non-zero coefficients correspond to the highlighted gray columns in D_t. Thus, X_{ql} can be written as a sparse linear combination of those highlighted gray columns.

To solve (2.2), besides linear programming, many other suboptimal techniques have been proposed including orthogonal matching pursuit [49], Subspace Pursuit (SP) [50] and gradient projection [51]. In this work, I employed Subspace Pursuit (SP) [50]. After finding the sparse representation vector α_{qlt}, to select the n candidate pixels, I simply pick those corresponding to n largest absolute value of coefficients in α_{qlt}. I denote cp_{qlt} as an $n \times 2$ matrix storing the locations of these candidate pixels and ρ_{qlt} as the length-n vector storing the corresponding values of α_{qlt}. I will take the normalized $|\rho_{qlt}|$ as a prior probability of matching the reference patch at $[q, l]$ to a patch of the interpolated LR frame y_t taking only local characteristics into account but ignoring geometric characteristics of the matches.

Finally, to incorporate geometric characteristics, I model the problem by a factor graph and apply belief propagation to update probabilities ρ_{qlt} (for more details, see [1]).

I assume the operations such as warping and blurring in the maximum *a posteriori* probability equation (1.2) are known. However, this is not true in practice. In particular, while the blurring operation can be approximated to be more or less constant over the entire scene, the warping operation could vary from pixels to pixels. One way to handle the unknown warping problem is to adopt the Bayesian formu-

lation and integrate all possible warping operations, this is of course too expensive to compute. Fortunately, the SCoBeP step has already provided us some candidate match locations. So (1.8) becomes

$$\mathcal{Z}[q,l] = \frac{\sum_{t=1}^{T}\sum_{b=1}^{n}\sum_{[i,j]|[ri,rj]\in\mathcal{N}(cp_{qlt}^{(b)})}\mathcal{W}[i,j,q,l,b,t]\mathcal{Y}_t[i,j]}{\sum_{t=1}^{T}\sum_{b=1}^{n}\sum_{[i,j]|[ri,rj]\in\mathcal{N}(cp_{qlt}^{(b)})}\mathcal{W}[i,j,q,l,b,t]}, \tag{2.3}$$

where $\mathcal{W}[i,j,q,l,b,t]$ can be interpreted as the weight mapping from a pixel in the reference frame to b^{th} candidate pixel in the t^{th} interpolated LR frame y_t. Since there are various other ways to choose the weights in (2.3), in this chapter I will restrict my choice to SCoBeP-SR weights and SCoBeP-NLM weights as described in the next subsections.

2.3.2 Calculate Weights for SCoBeP-SR

As SCoBeP has naturally identified pixels that are most likely to be relevant to a target pixel and also output the corresponding "weight" of the relevant pixels. Thus, I have introduced and implemented a new SCoBeP based SR algorithm, SCoBeP-SR, where "mixing" weights and candidates are extracted from the SCoBeP step only.

The method for calculating $\mathcal{W}[i,j,q,l,b,t]$ for SCoBeP-SR is based on the materials that have been developed in section 2.3.1. As some candidate locations and the corresponding belief are available from the SCoBeP output. I will simply assign the weights as zero except the candidate locations and the weights precisely as the beliefs output from SCoBeP. More precisely, I define

$$\mathcal{W}[i,j,q,l,b,t] = I(cp_{qlt}^{b} = [i,j])\rho_{qlt}^{(b)}, \tag{2.4}$$

where $I(cp^b_{qlt} = [i, j])$ is an indicator function which is equal to 1 if $cp^b_{qlt} = [i, j]$ and 0 otherwise. To maintain the original formulation, the neighborhood function of a patch $\mathcal{P}$ will just equal to the patch itself. That is, $\mathcal{N}(\mathcal{P}) = \mathcal{P}$.

2.3.3 Calculate Weights for SCoBeP-NLM

In this subsection, the method for estimating $\mathcal{W}[i, j, q, l, b, t]$, based on the materials that have been developed in Sections 1.2 and 2.3.1, is proposed as follows:

Algorithm 1 Super Resolution framework using SCoBeP - estimate version of HR frame $\mathcal{X}$

Inputs : LR and noisy frames $\{\mathcal{Y}_t\}_{t=1}^{T}$, resolution ratio r, weight patch R, frame number t_c and the maximum number of iterations

Initialize :

- Set $\mathcal{Z}$ as the bicubic interpolated frame of $\mathcal{Y}_{t_c}$

Iterate : while the maximum number of iterations is not reached

Use SCoBeP to find candidate pixels : For each increased resolution frame y_t

- Extract dense feature
- Construct dictionary D_t
- Find the initial estimates of candidate pixel probabilities ρ_{qlt} and candidate pixel locations cp_{qlt}
- Apply belief propagation to refine ρ_{qlt} and cp_{qlt}

Find the blurred HR frame : For each pixel location $[q, l]$ on the HR frame $\mathcal{Z}$ and for $b \in \{1, 2, \cdots, n\}$, for each pixel location $[i, j]$ such that $[ri, rj] \in \mathcal{N}(cp_{qlt}^{(b)})$

- Compute weights:

  1. SCoBeP-SR weights: $\mathcal{W}[i, j, q, l, b, t] = I(cp_{qlt}^{b} = [i, j])\rho_{qlt}^{(b)}$ OR
  2. SCoBeP-NLM weights:

$$\mathcal{W}[i, j, q, l, b, t] = \rho_{qlt}^{(b)} exp\left\{ -\frac{\left\| R_{q,l}\mathcal{Z} - R_{cp_{qlt}^{(b)}} y_t \right\|_2^2}{2\sigma^2} \right\}$$
$$\times f(\sqrt{(q - ri)^2 + (l - rj)^2 + \xi(t - t_c)^2})$$

- Compute $\mathcal{Z}$, the blurred version of reconstructed HR frame: $\mathcal{Z}[q, l] = \frac{\sum_{t=1}^{T}\sum_{b=1}^{n}\sum_{[i,j]|[ri,rj]\in\mathcal{N}(cp_{qlt}^{(b)})} \mathcal{W}[i,j,q,l,b,t]\mathcal{Y}_t[i,j]}{\sum_{t=1}^{T}\sum_{b=1}^{n}\sum_{[i,j]|[ri,rj]\in\mathcal{N}(cp_{qlt}^{(b)})} \mathcal{W}[i,j,q,l,b,t]}$

End of Iteration

Perform deblurring : $\mathcal{X}$ =TV$deblur(\mathcal{Z})$

Output : a HR frame $\mathcal{X}$

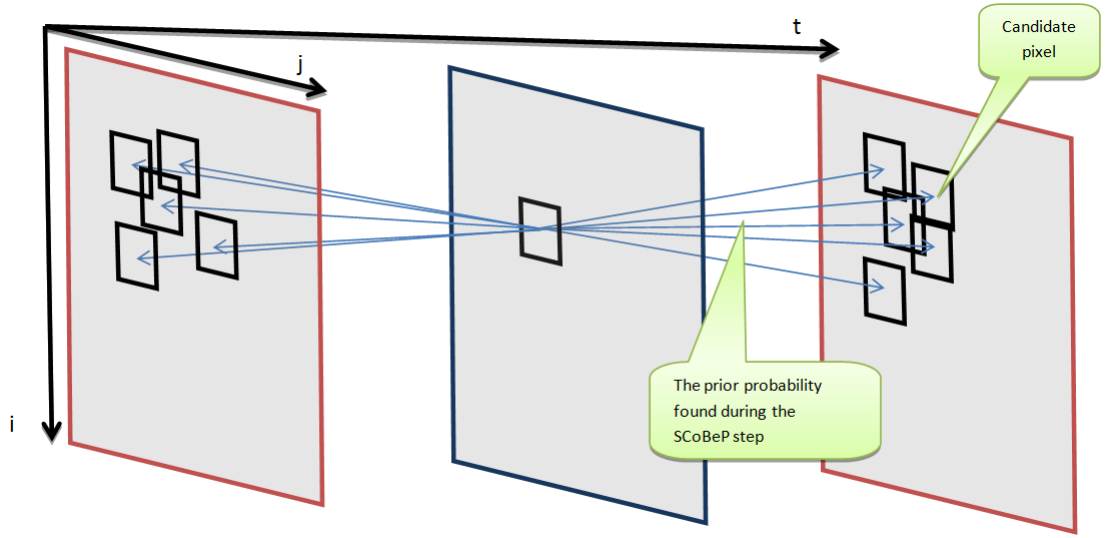


Figure 2.5: Candidate pixel and weight computation in SCoBeP. For the patch in the middle frame, SCoBeP weights the found candidate pixels along the space-time.

$$\mathcal{W}[i,j,q,l,b,t] = \rho_{qlt}^{(b)} exp\left\{ -\frac{\left\| R_{q,l}\mathcal{Z} - R_{cp_{qlt}^{(b)}} y_t \right\|_2^2}{2\sigma^2} \right\} \times f(\sqrt{(q-ri)^2 + (l-rj)^2 + \xi(t-t_c)^2}), \quad (2.5)$$

where t_c is the frame number of the output frame (see Fig. 2.5), and ξ is a scaling factor taking into account the difference in scale along the temporal and spatial dimensions. Note that I denote here $\rho_{qlt}^{(b)}$ as the b^{th} element of ρ_{qlt} (the probability of pixel $[q,l]$ in the reference frame mapping to the b^{th} candidate pixel in t^{th} interpolated LR frame), and $cp_{qlt}^{(b)}$ as the b^{th} row of cp_{qlt} (the b^{th} candidate location described by cp_{qlt}). Hence, $R_{cp_{qlt}^{(b)}}$ in (2.5) extracts a patch at the position $cp_{qlt}^{(b)}$ from frame y_t. To follow the notation easily, I summarized them in Table 2.1.

Note that computing weights involves the knowledge of the unknown frame $\mathcal{Z}$. For first iteration, the weights are computed by using an estimated version of $\mathcal{Z}$, which is a scaled-up frame generated by a conventional image interpolation algorithm such as bicubic, bilinear, or the lanczos method [3, 52, 53]. For the remaining iterations, the weights are computed using the estimated $\mathcal{Z}$ obtained in the previous iteration. The main procedure for my proposed methods are summarized in Algorithm 1, and also graphically depicted in Fig. 2.2. Note that in Algorithm 1, one can either pick SCoBeP-SR weights or SCoBeP-NLM weights for computing the weights.

As a summary of my approaches, I was able to write and minimize my cost

function which has two terms:

$$\epsilon^2_{MAP}(\mathcal{X}) = \frac{1}{2}\sum_{t=1}^{T}\sum_{[q,l]\in\mathcal{I}}\sum_{b=1}^{n}\sum_{[i,j]|[ri,rj]\in\mathcal{N}(cp^{(b)}_{qlt})}\mathcal{W}[i,j,q,l,b,t] \times \left\|DR^H_{q,l}H\mathcal{X} - R^L_{i,j}\mathcal{Y}_t\right\|_2^2 + \lambda\cdot TV(\mathcal{X}). \tag{2.6}$$

As described in Section 1.1, I followed [2, 7, 27] and decomposed (2.6) into two steps:

1. compute a blurred version of HR $\mathcal{Z}$ which $\mathcal{Z} = H\mathcal{X}$ by minimizing

$$\epsilon^2_{ML}(\mathcal{Z}) = \frac{1}{2}\sum_{t=1}^{T}\sum_{[q,l]\in\mathcal{I}}\sum_{b=1}^{n}\sum_{[i,j]|[ri,rj]\in\mathcal{N}(cp^{(b)}_{qlt})}\mathcal{W}[i,j,q,l,b,t] \times \left\|DR^H_{q,l}\mathcal{Z} - R^L_{i,j}\mathcal{Y}_t\right\|_2^2, \tag{2.7}$$

2. estimate the deblurred frame $\mathcal{X}$ from the found blurred HR $\mathcal{Z}$ in step 1:

$$\epsilon^2_{MAP}(\mathcal{X}) = \|H\mathcal{X} - \mathcal{Z}\|_2^2 + \lambda\cdot TV(\mathcal{X}), \tag{2.8}$$

I introduced the first step in Sections 2.3.1, 2.3.2, and 2.3.3, and as the second step is the conventional deblurring problem, many works can be applied here, which I simply adopted (AKTV) regularized locally-adaptive kernel regression in a variational approach developed by Takeda *et al.* [30].

I denote cp_{qlv} as an $n\times 2$ matrix storing the locations of these candidate pixels and ρ_{qlv} as the length-n vector storing the corresponding values of α_{qlv}. Each coefficient in ρ_{qlv} serves as a prior probability of matching the reference patch at $[q,l]$ to a LR patch of $\mathcal{Y}_v$ taking only local characteristic into accounts but ignoring geometric

characteristics of the matches. Finally, to incorporate geometric characteristics, I model the problem by a factor graph and apply belief propagation to update probabilities ρ_{qlv} (for more details, see [1]).

2.4 Experimental Results

In this section, I consider in two separated subsections with two different sets of experiments[1]. I utilize four test sequences (Miss America, Foreman, Suzie and Stefan) in Section 2.4.1 to compare the performance of my proposed methods with the state-of-the-art methods [2, 4, 6]. In that section I will first generate synthetic LR sequences and next apply super resolution methods to the degraded sequences. I will then compare the results to the ground truth (the original sequences). Also, in Section 2.4.2, I will illustrate additional examples that will assess my super resolution methods for real video sequences. Comparison will be made against the multi-image super resolution method proposed by Farsui *et al.* [7], 3-D ISKR method [4], super resolution Using TV prior method [6] and a single image up-sampling using the Lanczos algorithm [3], which were implemented using the software provided by their authors.

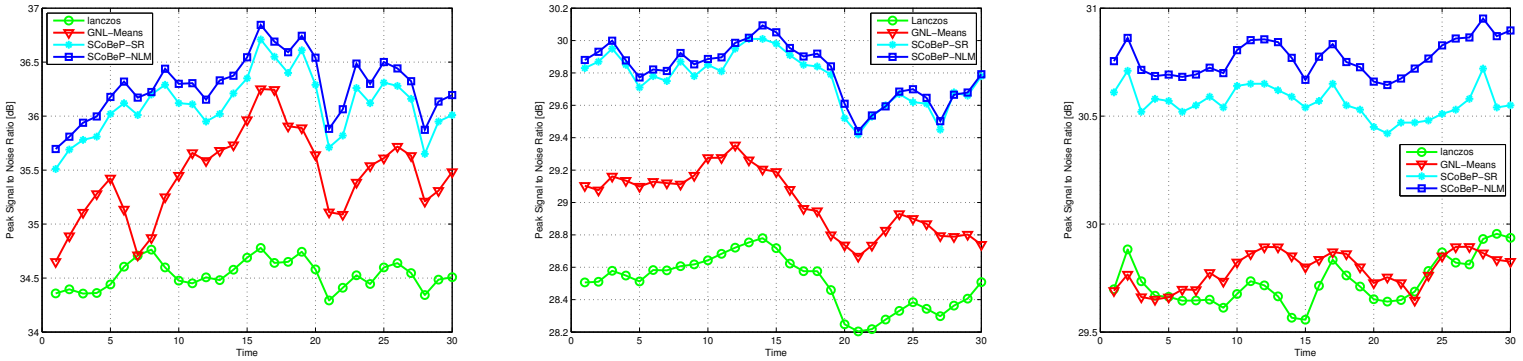
[1]The image frames of the result sequences using SCoBeP-NLM and SCoBeP-SR are available at http://students.ou.edu/B/Nafise.Barzigar-1/software/SCoBeP-NLM.html.

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

Figure 2.6: Results for the 8^{th}, 13^{th} and 23^{th} frame from the "Foreman" sequence. From Left column to Right column: LR frame; GNL-Means [2]; Lanczos interpolation [3]; result of the proposed SCoBeP-NLM; result of the proposed SCoBeP-SR. Also, the PSNR values for all the frames are shown in Fig. 2.7(b).

2.4.1 Evaluation On Synthetic Sequences

In this section, to evaluate my performance, I present some super resolution examples using existing sequences such as Miss America, Foreman, Suzie, and Stefan. The sequences in this section contain object motions only in the scene and no camera movement. All tests in this section were processed in the following manner: All 30 frames were involved in the reconstruction of each frame. The similar block size used for computing weight (R) was 13×13 and was not changed for various tests. The low patch extraction operator $R^L_{i,j}$ extracts only one pixel, therefore the $R^H_{q,l}$ extracts a patch of size 3×3 pixels. Also, the search area (the size of neighborhood $\mathcal{N}$) is 31×31 pixels. I set the parameter $\sigma = 2.2$ and the maximum number of iterations equal to 2 for all sequences.



(a) Miss America (b) Foreman (c) Suzie

Figure 2.7: PSNR values of each super resolved frame by Lanczos [3], GNL-Means [2], and the proposed method for (a) the results of Miss America shown in Fig. 2.11, (b) the results of Foreman shown in Fig. 2.6, and (c) the results of Suzie shown in Fig. 2.8. The average PSNR values for all frames for the Miss America example are 34.12[dB] (Lanczos), 35.09[dB] (GNL-Means [2]), 35.73[DB] (SCoBeP-SR) and 35.94[dB] (SCoBeP-NLM) and the average PSNR values for the Foreman example are 28.51[dB] (Lanczos), 29.01[dB] (GNL-Means [2]), 29.71[DB] (SCoBeP-SR) and 29.80[dB] (SCoBeP-NLM), and also the average PSNR values for the Suzie example are 29.73[dB] (Lanczos), 29.79[dB] (GNL-Means [2]), 30.56[DB] (SCoBeP-SR) and 30.77[dB] (SCoBeP-NLM), respectively.

(f) (g) (h) (i) (j)

Figure 2.8: Results for the 3^{th} and 23^{th} frame from the "Suzie" sequence. From Left column to Right column: LR frame; GNL-Means [2]; Lanczos interpolation [3]; result of the proposed SCoBeP-NLM; result of the proposed SCoBeP-SR. Also, the PSNR values for all the frames are shown in Fig. 2.7(c).

Table 2.2: PSNR for 8^{th} and 13^{th} frames of Miss America Sequence

Miss America Sequence	**Nearest Neighborhood**	**Lanczos [3]**	**GNL-Means [2]**	**3-D ISKR [4]**	**SCoBeP-NLM**	**SCoBeP-SR**
8^{th} frame	32.97	34.76	34.49	35.53	36.28	36.20
13^{th} frame	32.74	34.48	35.33	35.15	36.33	36.02

Figure 2.9: Video super resolution for Suzie sequence: (frame 28 and 18 with the resolution ratio 3, PSNR in brackets). From Left column to Right column: Ground truth; LR frame; GNL-Means [2] [PSNR: 29.87 - 29.86]; Lanczos interpolation [3] [PSNR: 29.41 - 29.27]; SCoBeP-NLM [PSNR: 30.95 - 30.75]; SCoBeP-SR [PSNR: 30.55 - 30.71].

Figure 2.10: Video super resolution for Foreman sequence: From Left column to Right column: Ground truth; GNL-Means [2]; 3-D ISKR [4]; NLKR [5]; result of the proposed SCoBeP-NLM.

Figure 2.11: Video super resolution for Miss America sequence; frame 8 (top) and frame 13 (bottom): From left to right: Ground truth; Lanczos interpolation [3]; GNL-Means [2]; 3-D ISKR [4]; super resolution Using TV prior [6]; SCoBeP-NLM; SCoBeP-SR.

To generate the LR frames, first, I degrade the test sequences by blurring the videos with a 3×3 uniform point spread function (PSF) and downsampling them by a resolution ratio of $3:1$ in both horizontal and vertical directions. Then the white Gaussian noise with standard deviation of $\sigma_{noise} = 2$ is added to each frame. Two of the selected LR sequences, Foreman and Suzie for frame numbers $8, 13, 23$, and frame numbers $3, 23$ are shown in Figs. 2.6(a), and 2.8(a), respectively. Then, I upscale the degraded videos using the Lanczos interpolation [3], the GNL-Means method [2], and my proposed methods. Figs. 2.6(b)–(e), and 2.8(b)–(e), respectively, show the results.

The graphs[2] in Fig. 2.7 show the frame by frame PSNR values of Miss America, Foreman and Suzie. My proposed methods beats the GNL-Means method in all frames by a significant margin for all sequences. The average PSNR values for my proposed methods and the compared methods are shown in the caption of Fig. 2.7.

The PSNR results of 8^{th} and 13^{th} frames of the Miss America sequence are summarized in Table 2.2, showing that the proposed methods again constantly outperform the current state-of-the-art methods. Note that the results from 3-D ISKR method is cited directly from [4]. In Fig. 2.9, I show the PSNR result and a clear visual comparison on the Suzie sequence. As shown in Fig. 2.9, although the GNL-Means method [2] acts well at regular-structured areas, it suffers from block artifacts[3] due to poor block matching. In contrast, my proposed methods performs remarkably well for both regular and detail structures and is free of these artifacts. In Fig. 2.10, I further show the results of Foreman sequence compared with the GNL-Means method [2], 3-D ISKR method [4] and NLKR [5]. The super resolution results on Miss America sequence in frames 8 and 13 and Stefan sequence are also given in Figs. 2.11 and 2.12, respectively for visual comparison. The proposed

[2]The PSNR results of 3D-ISKR [4] are not listed as they are not available in their original paper.

[3]Please note that I have adopted the terminology "block artifact" from [5]. The terminology is different from the artifacts typically found in low bitrate compressed image by old JPEG.

methods outperform the other methods by notable improvement.

Moreover, I examined how my methods perform under various noise levels. I added white Gaussian noise with standard deviation σ_n (varying from 0 to 2) to the LR sequences, where the sequence with $\sigma_n = 0$ was degraded by the downsampling process only. Table 2.3 shows that both SCoBeP-NLM and SCoBeP-SR are able to produce fine details when the noise level is increasing. For a clear comparison on varying noise level, I show the results of a noise added Foreman sequence in Fig. 2.13, where I compared my algorithms with the state-of-the-art 3-D ISKR [4].

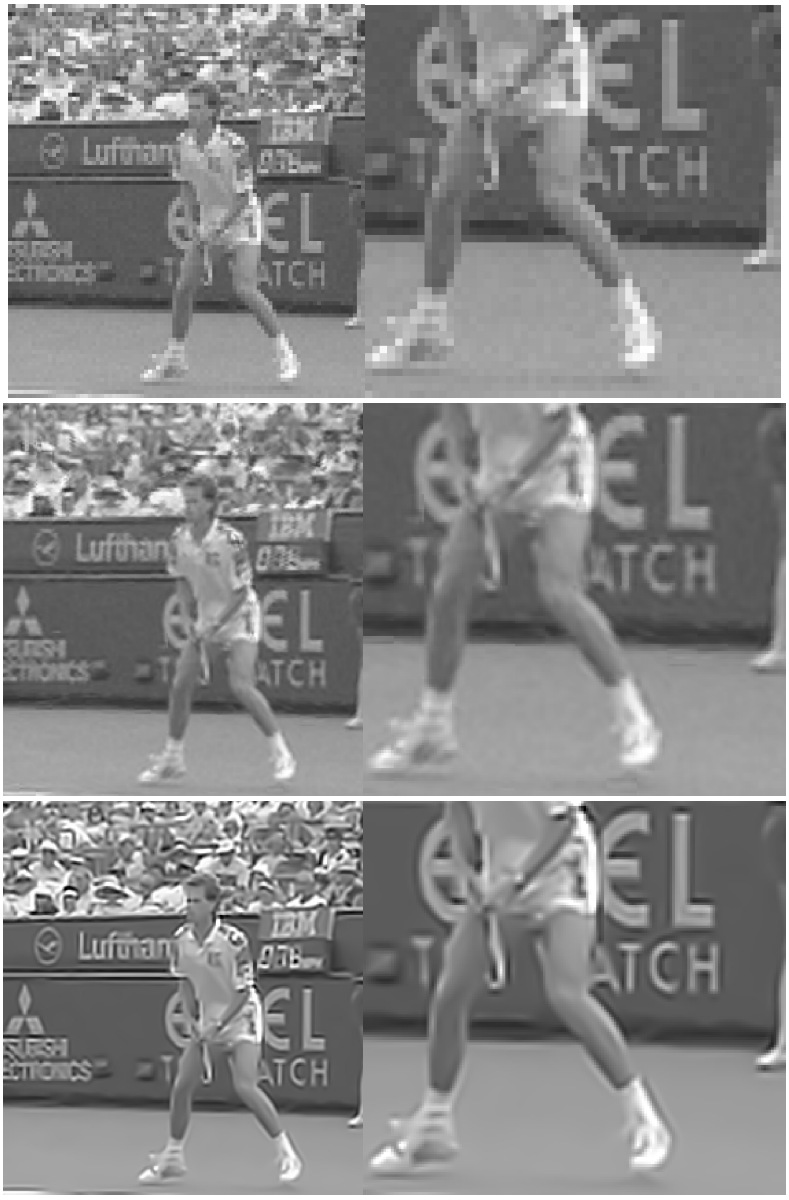


Figure 2.12: Video psuper resolution for Stefan sequence: From top to bottom column: LR frame; 3-D ISKR [4]; result of the proposed SCoBeP-NLM.

Figure 2.13: Video super resolution for Foreman sequence with noise: I added synthetic additive white Gaussian noise (AWGN) to the input LR sequence, with the noise level $\sigma_n = 1.20$ (left) and $\sigma_n = 2.00$ (right). From top to bottom column: Noisy LR; 3-D ISKR [4]; SCoBeP-NLM; SCoBeP-SR.

Table 2.3: Noise Addition: PSNR for 1^{st} frame of Foreman Sequence

	Lanczos [3]	**3-D ISKR [4]**	**SCoBeP-NLM**	**SCoBeP-SR**
$\sigma_n = 0.00$	28.51	28.94	29.88	29.76
$\sigma_n = 1.20$n	28.44	28.93	29.86	29.74
$\sigma_n = 1.60$	28.36	28.87	29.83	29.73
$\sigma_n = 2.00$	28.25	28.86	29.75	29.68

2.4.2 Evaluation on Real video Sequences

In this section, I turn to some real sequences, where I apply my proposed methods directly to the captured sequences without altering the frames. Note that there are no published methods that have tested on real sequences. As no standard sequence is available, I have captured a sequence for testing and with camera motion intentionally introduced. We choose the multi-image super resolution method proposed by Farsui *et al.* [7], 3-D ISKR method [4], super resolution Using TV prior method [6] and a single image up-sampling using the Lanczos algorithm [3] for comparison because their source codes are available publicly. Since no ground truth is available for a real sequence, I cannot evaluate the resulting HR frames with objective measure such as PSNR. However, the perceptual quality illustrate the robustness of my proposed methods on real videos.

Fig. 2.14 shows the superresolution results for a real Navajo Sculpture video sequence (70×80 pixels, 30 frames). One can see some "blocking" artifacts in the original sequence due to its low resolution as shown in Fig. 2.14(a). In Fig. 2.14(m), I illustrate the ability of my proposed methods in removing these artifacts and resulting in a clear output. I also show the superresolution results by the Lanczos interpolation [3], Farsui *et al.* method [7], 3-D ISKR method [4], and the TV prior method [6] with three time magnification per each dimension (i.e., an output resolution of 210×240 pixels) in Fig. 2.14(c)–(i), respectively. As shown in Fig. 2.14, the Farsui *et al.* method [7] and the TV prior method [6] introduce severe block artifacts (near the *mouth* and the *eyes* in Figs. 2.14(e) and 2.14(i) respectively), and the 3-D ISKR method [4] does not preserve the line texture well and generates the ghost image as shown in Fig. 2.14(g). In contrast, my proposed methods do not suffer from these artifacts.

The computational complexity of SCoBeP-SR can be determined by considering the following two steps: 1) computing the locations and prior probabilities of the candidate pixels, 2) calculating weights via the NLM and SCoBeP or only SCoBeP. The complexity associated with the computing the location and weights of the candidate pixels takes 70% of the overall complexity in SCoBeP-NLM. Since I replace NLM weight with SCoBeP weight in Algorithm 1 and I found the probabilities for the candidate matches in the previous step it can significantly reduce computation complexity and storage requirement. Just to put things into perspective, note that the current implementation requires approximately 700 s per frame for the Algorithm 1 using SCoBeP-NLM weights, and 230 s for Algorithm 1 using SCoBeP-SR weights in the most demanding case like "Navajo" sequence with high-resolution frame size of 250×220, with the current pure Matlab implementation on a Pentium 3 GHz (11-GB RAM) machine. In comparison, ISKR takes approximately 5784 s per frame.

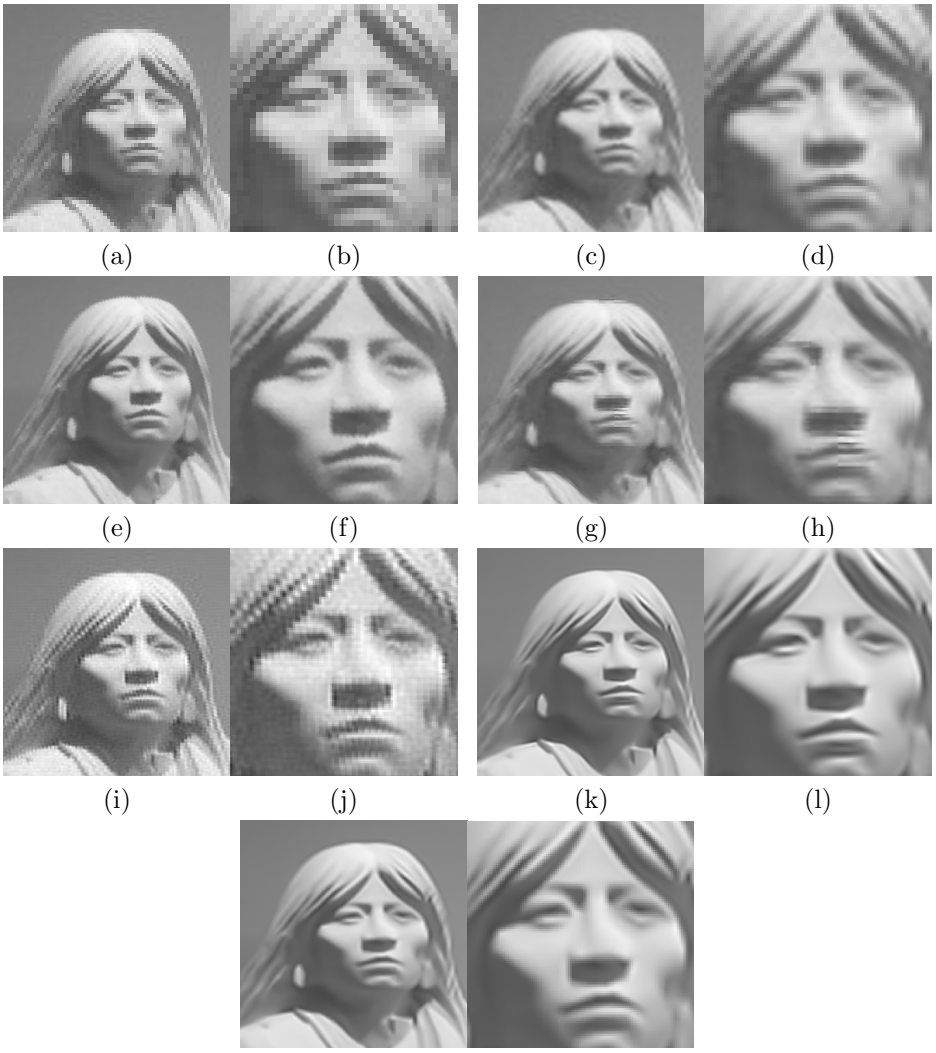


Figure 2.14: Multi-frame super resolution for real frames: "Navajo" sequence. (a,b) LR frame; (c,d) Lanczos interpolation; (e,f) Farsui *et al.* [7] method; (g,h) 3-D ISKR [4]; (i,j) super resolution Using TV prior [6]; (k,l) SCoBeP-NLM; (m,n) SCoBeP-SR.

CHAPTER 3

DENOISING METHOD USING BLOCK MATCHING FILTERING AND LOW-RANK MATRIX COMPLETION

3.1 Introduction

The recovery of an unknown low-rank or approximately low-rank matrix from very limited information is a recent fast growing interest. Consider a large matrix with only a small portion of known entry, an interesting problem is to fill the missing entry assuming the matrix has low-rank. The problem, which is referred to as matrix completion, or more precisely low-rank matrix completion, has gained increasing interests in research communities in recent years. So far, this problem has been studied in many applications such as collaborative filtering [54], system identification [55], computer vision [56], machine learning [57–59], global positioning [60] and remote sensing [61]. An example is the famous Netflix challenge where a huge matrix is used to represent the rating of a movie given by a user. Of course, a typical user will only rate very few movie titles. Therefore, an algorithm will be needed to complete the matrix to predict the ratings of all movies among all users.

It has been shown theoretically that under certain assumptions the matrix can be recovered with very high accuracy [62–64]. Their approaches convert the rank minimization problem into a nuclear norm minimization problem instead and thus can be solved using semidefinite program (SDP). However, the complexity grows rather rapidly with the size of the matrix n ($\sim n^3$). Candes and Recht [62] showed

that one can perfectly recover most low-rank matrices from what appears to be an in complete set of entries, and they proved in some condition, most $n \times n$ matrices of rank r can be perfectly recovered by solving a simple convex optimization program. Also, the authors claimed that their method is accurate even when the few observed entries are corrupted by a small amount of noise. In another work, the problem of recovering low-rank and sparse matrices using a greedy algorithm was discussed for large matrix sizes [65]. Several efficient algorithms have been proposed including Singular Value Thresholding (SVT) [66], Atomic Decomposition for Minimum Rank Approximation (ADMiRA) [67], Fixed Point Continuation with Approximate (FPCA) [68], Accelerated Proximal Gradient (APG) [12], Subspace Evolution and Transfer (SET) [69], Singular Value Projection (SVP) [70], OptSpace [64], and LMaFit [11], where OptSpace and SET are based on Grassmann manifold optimization, SVT and SVP uses iterative hard thresholding (IHT) to facilitate matrix shrinkage, FPCA utilizes Bregman iterative algorithm and Monte Carlo approximate SVD, and LMaFit adopts successive over-relaxation (SOR).

In this chapter, I use a decomposition method [24] to allow very efficient divide-and-conquer approach when known entries are relatively very few. A simple "trimming" method ed in that work will recover the decomposed "cluster" matrix. However, the decomposition method can also be combined with any other existing matrix completion techniques to yield further gain. One advantage of the decomposition approach [24] is that unlike most existing approaches it does not utilize SVD but only relies on basic vector operations. Therefore, the approach is immediately applicable to matrices of any field (including finite field matrices). This opens up opportunities for new applications.

Video sequences are often corrupted by noise during acquisition or transmission. Some noise sources located in camera hardware became active during image acquisition under some lighting conditions. Other noise sources are over transmission

channels. Most video denoising algorithms proposed in the literature assume additive white Gaussian noise, which can be categorized into pixel domain and transform domain methods. However, I consider Impulsive/Poisson/Gaussian noise in my work and will show how robust my video denoising method is.

Many video denoising methods have been proposed in the last few decades, e.g., [9, 10, 71, 72]. One of the first methods to address the denoising problem was the bilateral filter, which was proposed by Tomasi and Manduchi [72]. However, this method fails to perform well in when the noise is strong. Selesnick and Li [10] proposed 2D and 3D dual-tree oriented wavelet transforms which give a motion-based multi-scale decomposition for video. They used the proposed transforms for video denoising, where the 2D transform is applied to each frame individually.

Recently, the idea of patch based sparse coding has been applied to video denoising [9,17,18,73]. Marial *et al.* in [17] suggested to extend the sparse coding approach by proposing that similar patches share the same dictionary elements in their sparse decomposition on denoising. Another recent example based on an enhanced sparse representation in transform domain is block-matching 3-D filter (BM3D) [18]. In BM3D, similar 2D image blocks are grouped into a 3D data array based on the l_2 norm distance function. Then, the 3D data array is filtered by wavelet shrinkage or Wiener filter in 3D transform domain. The denoised image is produced from all grouped blocks after applying the inverse 3D transform. The concept of BM3D is generalized to video denoising in VBM3D [9]. In VBM3D, the noisy video is processed in a block-wise manner in both spatial and temporal domains. Then, a predictive search block-matching is combined with collaborative hard thresholding or collaborative Wiener filtering.

Unfortunately, prior works have been limited to one specific type of noise, there also exist other types of noise will degrade the performance of the denoising methods. In contrast, my method does not suffer from this limitation and can remove serious

mixed noise from video sequences.

In this work, I show that the proposed method can operate directly on the raw noisy images that suffer from non-homogeneous noise. The proposed method is similar to that described in [74]. However, I incorporate the matrix completion method [24] into the denoising algorithm and rather than applying a suboptimal block matching algorithm as in [75], I use a near-optimal block matching method [76] with higher complexity. I can afford latter as the matrix completion method runs significantly faster than other matrix completion methods. The goal of my denoising method is to keep only the reliable pixels and get rid of all other un-reliable pixels I find as noise. For each patch in the reference frame, I find the similar patches in the other frames using a block matching algorithm. The found matches will be vectorized and then stacked into a matrix. The reliable pixel values in the matrix are between the mean $\pm$ standard deviation of all elements in the same row. The main step will be done by applying the matrix completion approach [24] on the incomplete matrix. The output of matrix completion is a noise free full matrix. Then, the average value of each row in the full matrix can recover the denoised patch. Repeating the same procedure for all blocks of reference frame can build a denoised frame.

3.2 Denoising Method

The problem of video denoising can mathematically be shown as

$$y(x) = z(x) + n(x), \tag{3.1}$$

where $z(x)$ is the original video signal and $y(x)$ is the observed video after being corrupted by Gaussian/Poisson/Impulsive noise $n(x)$. $x = (i, j, k) \in X$ are coordinates in the spatio-temporal $3D$ domain $X \subset \mathbb{Z}^3$, where the first two components

Algorithm 2 Video Denoising using matrix completion- estimate version of denoised image $\mathcal{X}$

Inputs : noisy video y, pixel overlap v
Initialize :

- Set $\mathcal{V}$ and $\mathcal{W}$ to be zero images of the same size as the video frame size.

Produce the pre-processing step for removing impulsive noise before patch matching :

- Apply Adaptive Median Filter:
  $Y^{am} = AMF(y)$

Find the denoised patches: For each coordinate $x \in \Omega$ with v pixel overlap in each direction do:

(a) $S_x = BM\Big(Y_x^{am}\Big)$

(b) $\hat{\mathcal{Z}_{S_x}} = ReliableElements\Big(\mathbf{Y}_{S_x}\Big)$

(c) $\check{\mathcal{Z}}_x = DMC\Big(\hat{\mathcal{Z}}_{S_x}\Big)$

(d) $\hat{z}_x = AVG_{row}(\check{\mathcal{Z}}_x)$

(e) $\mathcal{V} = \mathcal{V} + \hat{z}_x$

(f) $\mathcal{W} = \mathcal{W} + \hat{w}_x$

Normalize : $\hat{\mathcal{Z}} = \mathcal{V}/\mathcal{W}$
Output : a denoised image $\hat{\mathcal{Z}}$

(i, j) are the spatial coordinates and the third one k is the time (frame) index. The main procedure for my proposed denoising method is summarized in Algorithm 2.

Implementation Details:

- $Y^{am} = AMF(y)$ performs adaptive median filtering using y. Because, the video is corrupted by image noise, applying a patch matching algorithm directly on noisy video generates unreliable result. Specifically, the block matching algorithm will suffer from impulsive noise, and its performance will be seriously degraded by strong impulsive noise. Hence, using a preprocessing step

to remove impulsive noise before the block matching step will improve the resulting performance. In my work, I simply use the adaptive median filter proposed by Hwang and Haddad in [77].

- $\Omega \subset X$ is a set that includes the coordinates of the reference blocks. In general, each pixel in the reference image is covered by several patches. I aggregate overlapped patches by a weighted average at each pixel.

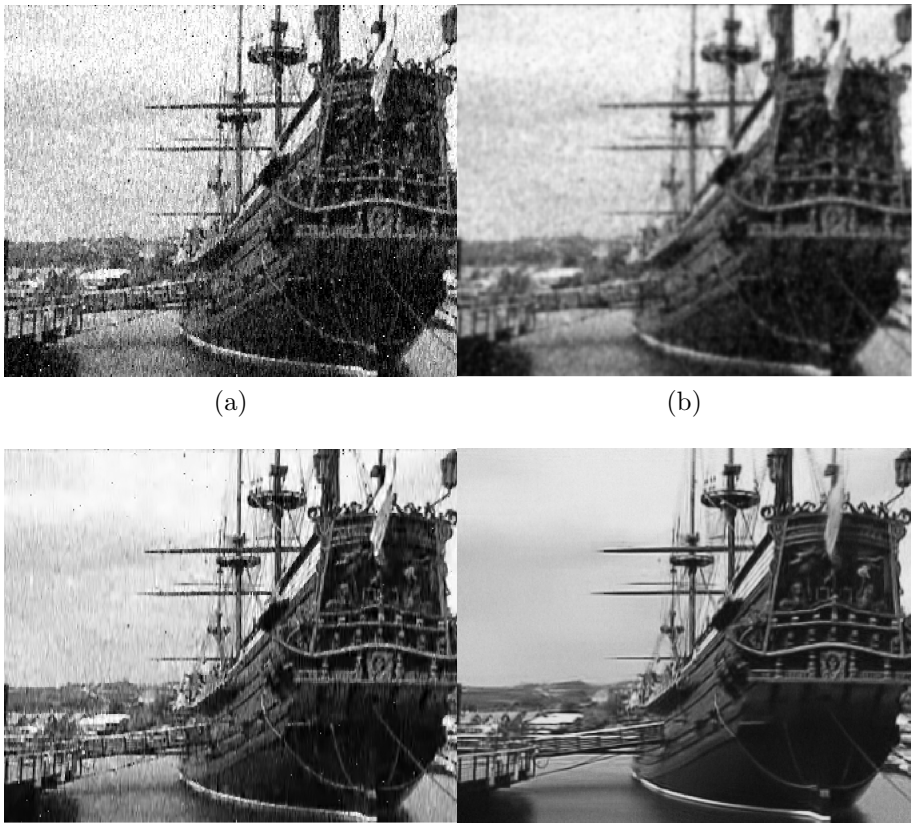
- Y_x^{am} denotes a block of size $q \times q$ in Y^{am}, where its center is at x.

- $S_x = BM\Big(Y_x^{am}\Big)$ presents a block matching algorithm using Y_x^{am} as a reference block, where the result is the set S_x containing the coordinates of the matched blocks. Although there are several methods to find the similar matches [1,9,78, 79], in my work, I use the Adaptive Rood Pattern Search (ARPS) algorithm [76] because of its computational efficiency.

- $\mathbf{Y}_{S_x}$ denotes a matrix formed by stacking the vectorized blocks $Y_{x \in S_x}$ together, where Y_x is a block of size $q \times q$ centered at x in y.

- $\hat{\mathcal{Z}}_{S_x} = ReliableElements\Big(\mathbf{Y}_{S_x}\Big)$ discards those matrix elements of $\mathbf{Y}_{S_x}$ that are far away from $mean \pm standard\ deviation$ of its corresponding row, designates them as unreliable elements, and then replaces them by zero. Note that those unreliable elements could be the pixels corrupted by Gaussian/Poisson/Impulsive noise or from mismatched patches obtained from previous step (block matching). Also, keeping the reliable elements, lets us recover the full matrix needed for the next step.

- $\check{\mathcal{Z}}_x = DMC\Big(\hat{\mathcal{Z}}_{S_x}\Big)$ performs a decomposing approach for low-rank matrix completion algorithm (see Section [24]) using $\hat{\mathcal{Z}}_{S_x}$ and $\check{\mathcal{Z}}_x$ that will be a full matrix with noise free elements. Recently, many matrix completion methods



(a) (b)

(c) (d)

Figure 3.1: Video denoising for *Galleon* sequence: (PSNR in brackets). From left to right: noisy image; tvregv2 [8] [PSNR: 17.8208]; VBM3D algorithm [9] [PSNR: 17.9226]; result of the proposed denoising algorithm [PSNR: 21.2437].

have been studied [11, 55, 64]. In my work, I use a decomposing approach for low-rank matrix completion algorithm, because of its computational efficiency.

- $\hat{z}_x = AVG_{row}(\check{\mathcal{Z}}_x)$ finds the average value of each row in matrix $\check{\mathcal{Z}}_x$ and converts the obtained vector to a block. Also, $\hat{z}_x$ will be an estimated block of size $q \times q$ centered at x in $\hat{\mathcal{V}}$.

- $\hat{w}_x$ is a patch with the same size as $\hat{z}_x$. Note that, all pixel values in $\hat{w}_x$ are equal to 1.

Table 3.1: Output PSNR of my proposed denoising method for the two video sequences; note that, I kept the Gaussian and Poisson noise constant in all tests

Video name (frame size) / **Impulsive noise density**	**Suzie**	**Coastguard**
0.10	26.3679	22.6124
0.15	26.2852	22.6060
0.20	26.3810	22.5819
0.25	26.3135	22.6281
0.30	26.3545	22.6105
0.35	26.4119	22.6163
0.40	26.2867	22.6346
0.45	26.3257	22.6149
0.50	26.3257	22.5907

Table 3.2: PSNR and time comparison for using various matrix completion

		Tempete	**Galleon**	**Coastguard**
Proposed method	$PSNR[dB]$	23.37	22.73	23.57
	$Time(seconds)$	240	218	110
Denoising method using OptSpace [64]	$PSNR[dB]$	22.79	22.60	23.63
	$Time(seconds)$	1355	1683	538
Denoising method using LMAFIT1 [11]	$PSNR[dB]$	22.78	22.52	21.81
	$Time(seconds)$	220	251	119
Denoising method using FPCA [68]	$PSNR[dB]$	21.38	20.39	20.97
	$Time(seconds)$	5828	7069	1584

Table 3.3: Average PSNR for the two video sequences

Sequence	**Wiener2**	**VBM3D [9]**	**3DWTF [10]**	**tvregv2 [8]**	**Proposed denoising Method**
Miss America	26.6796	30.9090	24.5168	28.1036	32.1931
vtc1nw	25.5855	28.2356	22.1496	27.7033	31.6442

3.3 Experimental Results on Denoising

3.3.1 Natural Image Experimental Results

In this section, I present some video denoising examples to evaluate my performance, using existing sequences such as Miss America, Galleon, and Suzie. All tests in this section were processed in the following manner: All 30 frames were involved in the reconstruction of each image. The block size used for block matching (q) was 20×20 and was not changed for various tests. I obtained a locally consistent solution by allowing patches to overlap, where the overlapped regions (v) were 5 pixels in each direction. Also, for each reference patch, I extracted 5 most similar patches used in each frame using block matching algorithm. For simplicity, I employ the basic version of my algorithm without taking advantage of sub u-diagonalization.

In Fig. 3.9, I show the PSNR result and a clear visual comparison of the Galleon sequence. The original video is seriously corrupted by a significant mixed noise level with Poisson noise, Gaussian white noise of mean zero and variance 0.02, and Impulsive noise of the noise density 0.03. As shown in Figures. 3.9 and 3.8, VBM3D method [9] and tvregv2 [8] generate severe artifacts at edge areas, while my proposed denoising method performs remarkably well for the detail structures and is free of these artifacts.

In Table 3.4, I present the PSNR results of the proposed denoising algorithm for a few sequences, where Impulsive noise is changing. This table shows how my algorithm is robust in denoising the corrupted sequences of serious impulsive noise.

In Graph 3.7, I compare my denoising method with the VBM3D method [9], which is among the state-of-the-art in video denoising. In this comparison, I apply my denoising method on Coastguard and Suzie sequences, for which I changed the Gaussian noise but kept the Poisson and Impulsive noise constant for all methods. Note that, for a fair comparison and also because the VBM3D method [9] works

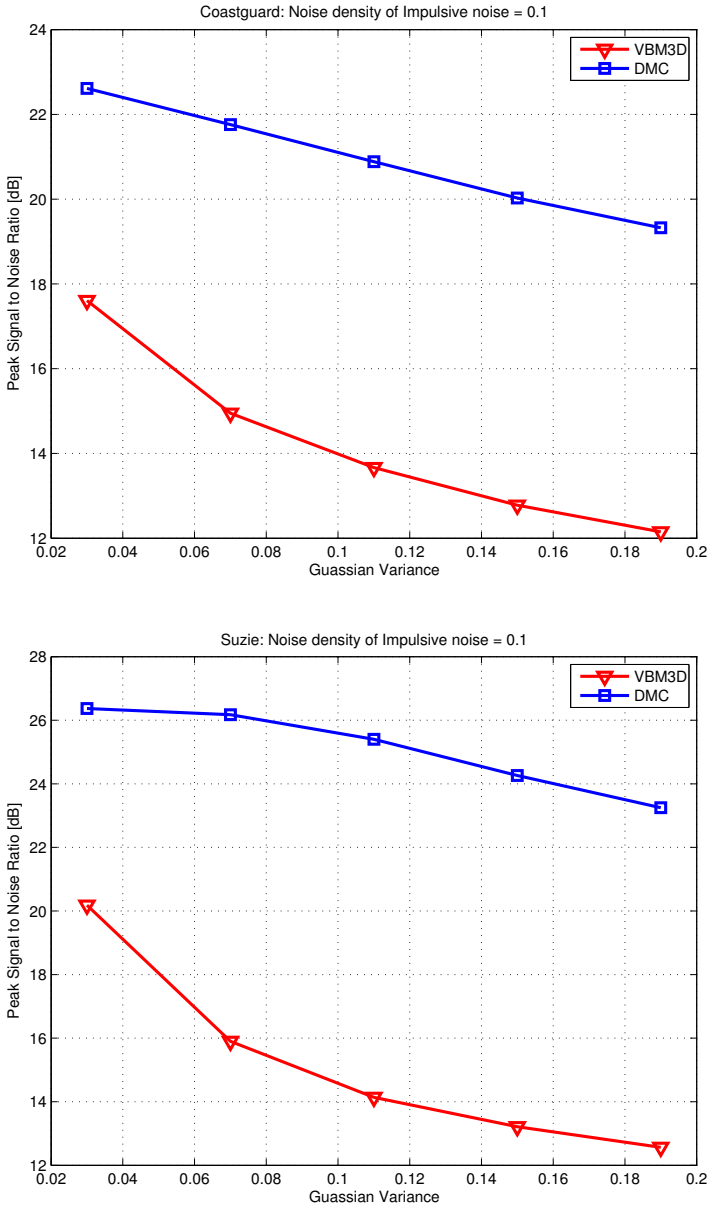
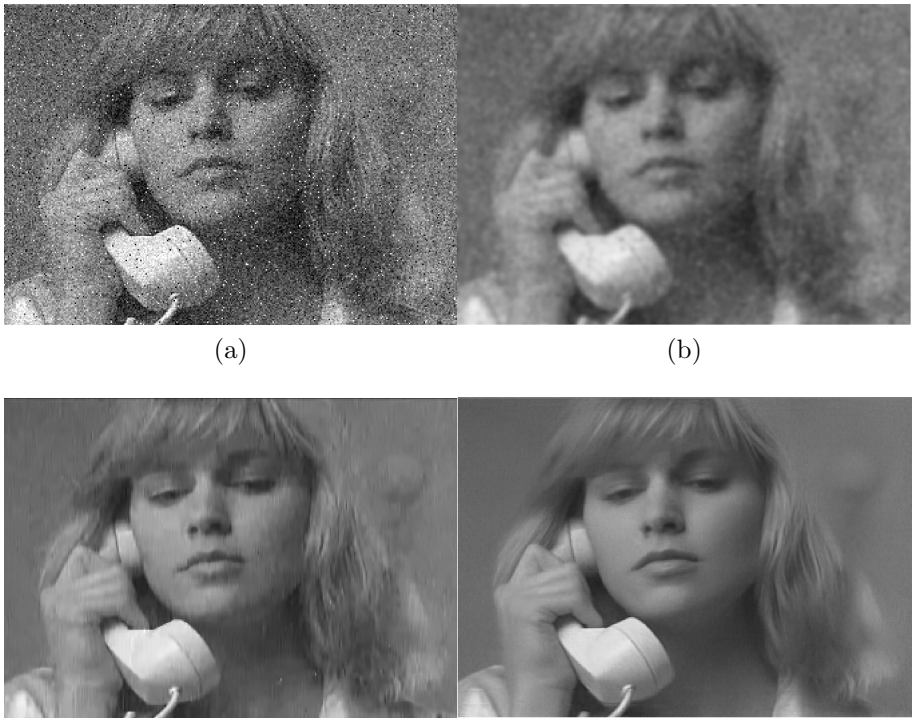


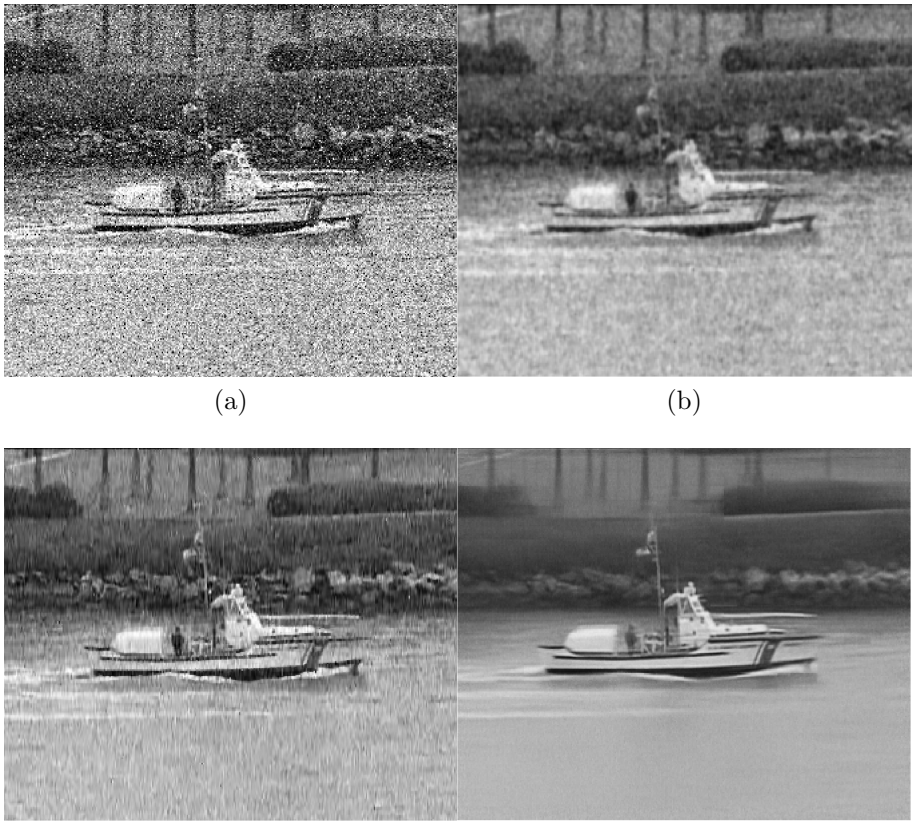
Figure 3.2: PSNR values of VBM3D algorithm [9] and proposed denoising method for *Coastguard* and *Suzie* sequences. Note that, I kept the Impulsive noise consistent in all tests.



(a) (b)

(c) (d)

Figure 3.3: Video denoising for *Suzie* sequence: (PSNR in brackets). (a) noisy image; (b) tvregv2 [8] [PSNR:25.1205]; (c) VBM3D algorithm [9] [PSNR:26.5245]; (d) result of the proposed denoising algorithm [PSNR:29.3254].



(a) (b)

(c) (d)

Figure 3.4: Video denoising for *Coastguard* sequence: (PSNR in brackets). (a) noisy image; (b) tvregv2 [8] [PSNR:20.9469]; (c) VBM3D algorithm [9] [PSNR:21.0090]; (d) result of the proposed denoising algorithm [PSNR:23.5725].

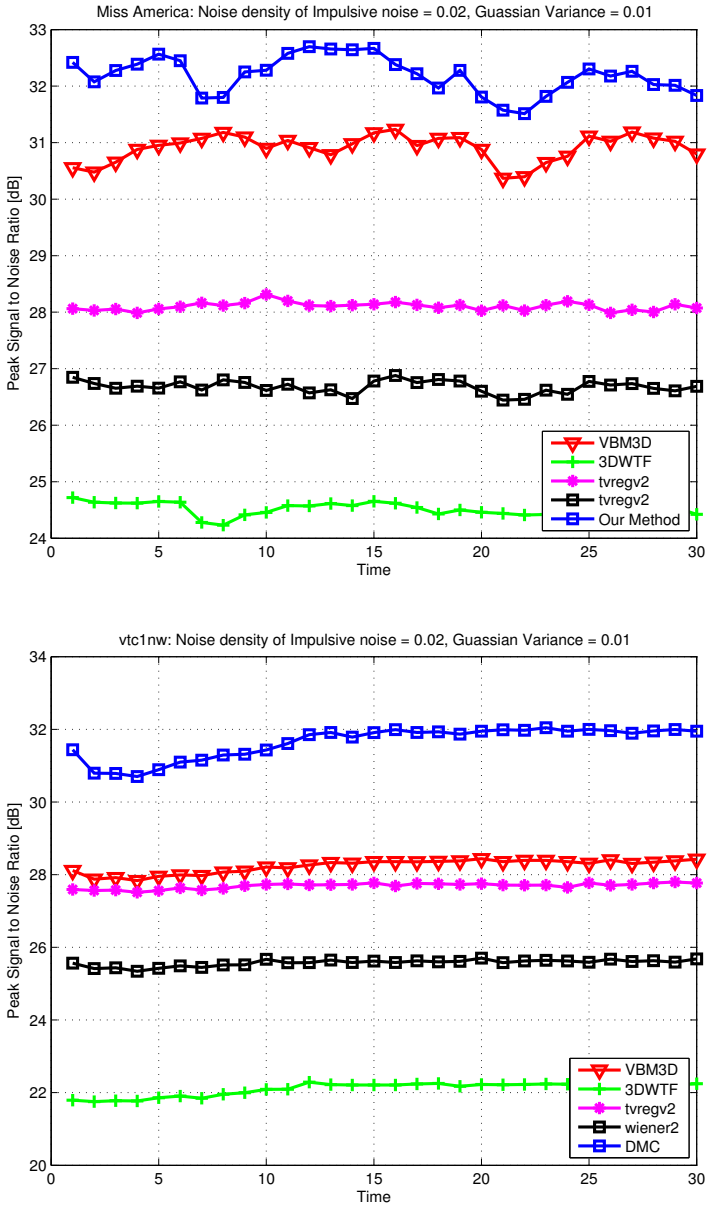


Figure 3.5: PSNR values of each denoised frame by VBM3D algorithm [9], 3DWTF [10], tvregv2 [8], wiener2 and the proposed denoising method for (a) the *Miss America* and (b) the *vtc1nw* sequence.

on removing just the Gaussian noise from the corrupted video, I ran the adaptive median filter method [77] on the test data with a pre-process of removing impulsive noise. In contrast, in my work, I did not use any existing impulsive noise method to detect pixels corrupted by Impulsive noise. The graph in Fig. 3.5 shows the frame by frame PSNR values of Miss America and vtc1nw. Table 3.3 shows the average PSNR values for my proposed method and the compared methods. My proposed method surpasses the VBM3D method [9] in all frames by a significant margin for all sequences with more than 2dB. In contrast, while [74] also outperform the VBM3D method but with a significantly smaller margin, I conjecture that the gain is due to the near-optimal block matching method [76] used in my approach.

I also replaced my proposed decomposition matrix completion with OptSpace [64], LMAFIT1 [11] and FPCA [68] to compare the result and time consumption (see Table 3.6). It can be seen in Table 3.6 that my method has comparable performance in terms of PSNR, for which it executes much faster than those methods.

3.3.2 Inpainting Experimental Results

To graphically illustrate the effectiveness of my proposed method, I applied it to image inpainting. In grayscale image inpainting, the value of some of the pixels on the image are missing, and the task here is to fill these missing values. Note that the missing pixel positions in the image inpainting are not randomly distributed. If the image is of low-rank, or of numerical low-rank, the matrix completion solvers can be applied on the image inpainting problem to obtain low-rank approximations.

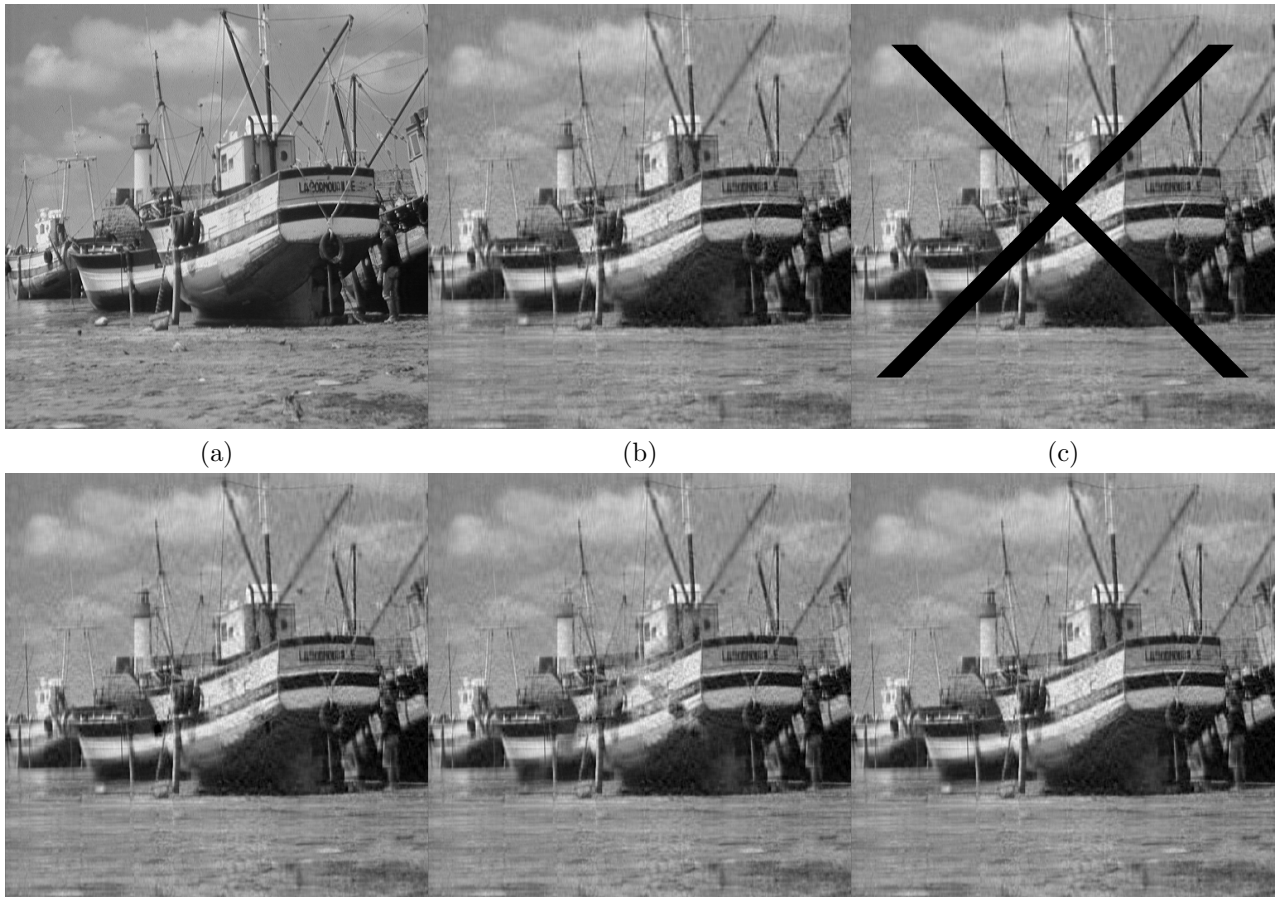


Figure 3.6: Image inpainting Problem for *Boat* image: (a) Original image; (b) rank 40 image; (c) deterministically 9.30% masked rank 40 image; (d) LMaFit1 [11]; (e) APGL [12]; (f) result of the proposed method.

The 512×512 original grayscale image is shown in Fig. 3.6 (a). Fig. 3.6 (b) was obtained by truncating the SVD of the images to get the images of rank 40. Fig. 3.6 (c) is the masked image obtained from Fig. 3.6 (b), where 9.30% of the pixels were masked in a non-random fashion. The recovered images of Fig. 3.6 (c) from LMaFit, APGL, and my proposed method are depicted in Figs. 3.6 (d), (e), and (f).

3.3.3 Medical Experimental Results

To evaluate the performance of the proposed method, I consider in two separated subsections with two different sets of experiments. In Section 3.3.3, I will first generate synthetic noisy slices and then apply denoising methods to the degraded slices. I will then compare the results to the ground truth (the original slices) and results generated from other state-of-the-art techniques [13–15]. Moreover, in Section 3.3.3, I will illustrate additional examples that will assess my denoising method for real CT slices. Comparison will be made against two state-of-the-art techniques: the adaptive multiscale image denoising algorithm [15] and wavelet domain image denoising algorithm [14].

I also replaced my proposed decomposition matrix completion with OptSpace [64] to compare the result and time consumption (Table 3.6). It can be seen that my method compares well or even betters in terms of PSNR, and it performs notably faster than OptSpace [64].

Evaluation on Synthetic Slices

To evaluate the performance of my approach, I conducted tests on the data sets LIDC-IDRI [80] where the size of each slice of the CT slices are 512×512 pixels. All tests in this section were processed in the following manner: All 6 slices were involved in the denoised slice. The similar block size used for block matching was

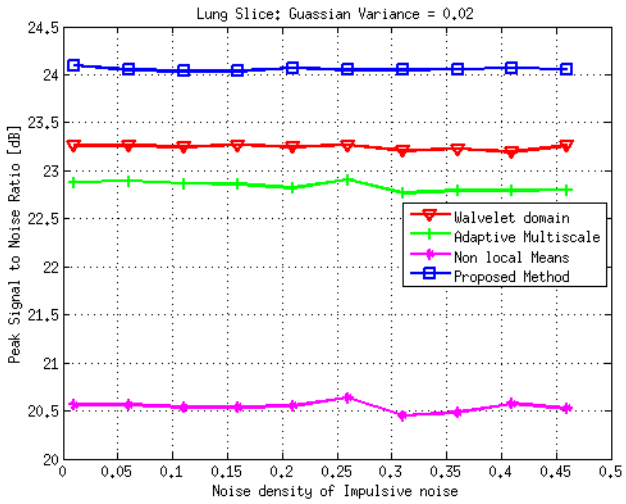


Figure 3.7: PSNR values of non-local means algorithm [13]; wavelet domain image denoising algorithm [14]; adaptive multiscale image denoising algorithm [15]; result of the proposed denoising algorithm for lung CT slices; Note that, I kept the Gaussian noise constant in all tests.

8×8 and was not changed for various tests. I obtained a locally consistent solution by allowing patches to overlap, where the overlapped regions (v) were 5 pixels in each direction. Further, for each reference patch, I extract 5 most similar patches used in each slice using block matching algorithm.

In Fig. 3.8, I show the PSNR result and a clear visual comparison on the CT slices. The original CT slice is corrupted by a mixture of Poisson noise, Gaussian white noise, impulsive noise with significant noise level (variance of Gaussiance noise = 0.02 and noise density of impulsive noise = 0.01). As shown in the figure, non-local means algorithm [13], wavelet domain image denoising algorithm [14], and adaptive multiscale image denoising algorithm [15] generate severe artifacts at edge areas, while my proposed denoising method performs remarkably well for the detail structures and is free of these artifacts.

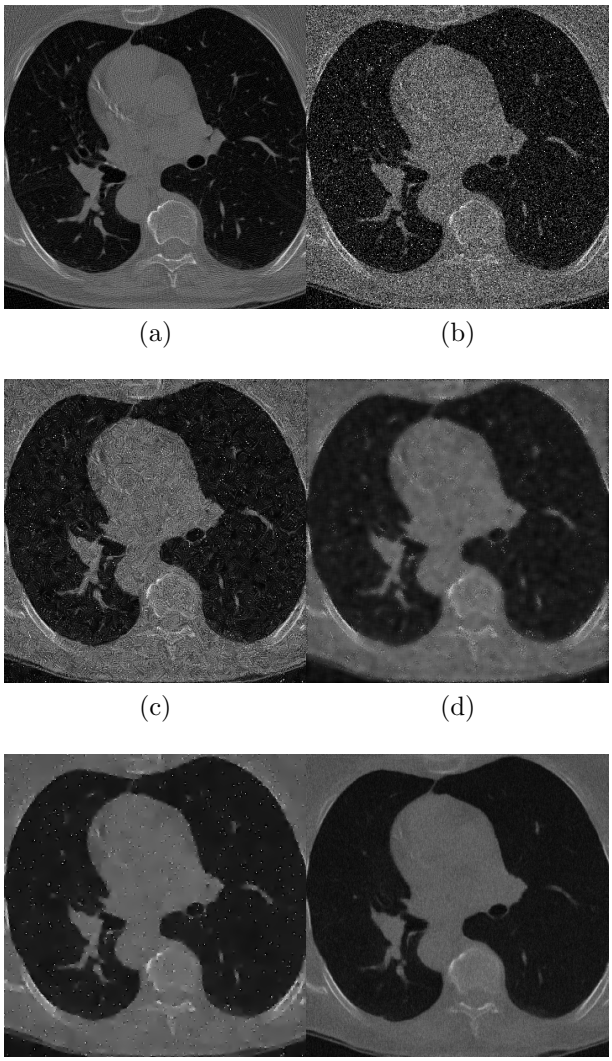
To quantify my denoising performance, I used the Peak Signal to Noise Ratio

Table 3.4: PSNR of my proposed denoising method for the lung slices; note that, I kept the variaces of Gaussian and Poisson noises constant in all tests.

Noise density of Impulsive noise	**Lung Slices** (362×362)
0.06	24.10
0.06	24.05
0.11	24.04
0.16	24.04
0.21	24.07
0.26	24.05
0.31	24.05
0.36	24.05
0.41	24.07
0.46	24.06

(PSNR) measure between the ground truth and denoised CT slice:

$$PSNR = 10 \ \ log_{10}(255^2/Mean \ \ Square \ \ Error)[dB]. \tag{3.2}$$



(a) (b)

(c) (d)

(e) (f)

Figure 3.8: Synthetic experiment 1: (PSNR in brackets). (a) ground truth; (b) noisy slice; (c) non-local means algorithm [13] [PSNR:20.56]; (d) wavelet domain image denoising algorithm [14] [PSNR:23.26]; (e) adaptive multiscale image denoising algorithm [15] [PSNR:22.78]; (f) result of the proposed denoising algorithm [PSNR:24.07]. Note that, I kept the Gaussian/Poisson/Impulsive noise constant in all tests.

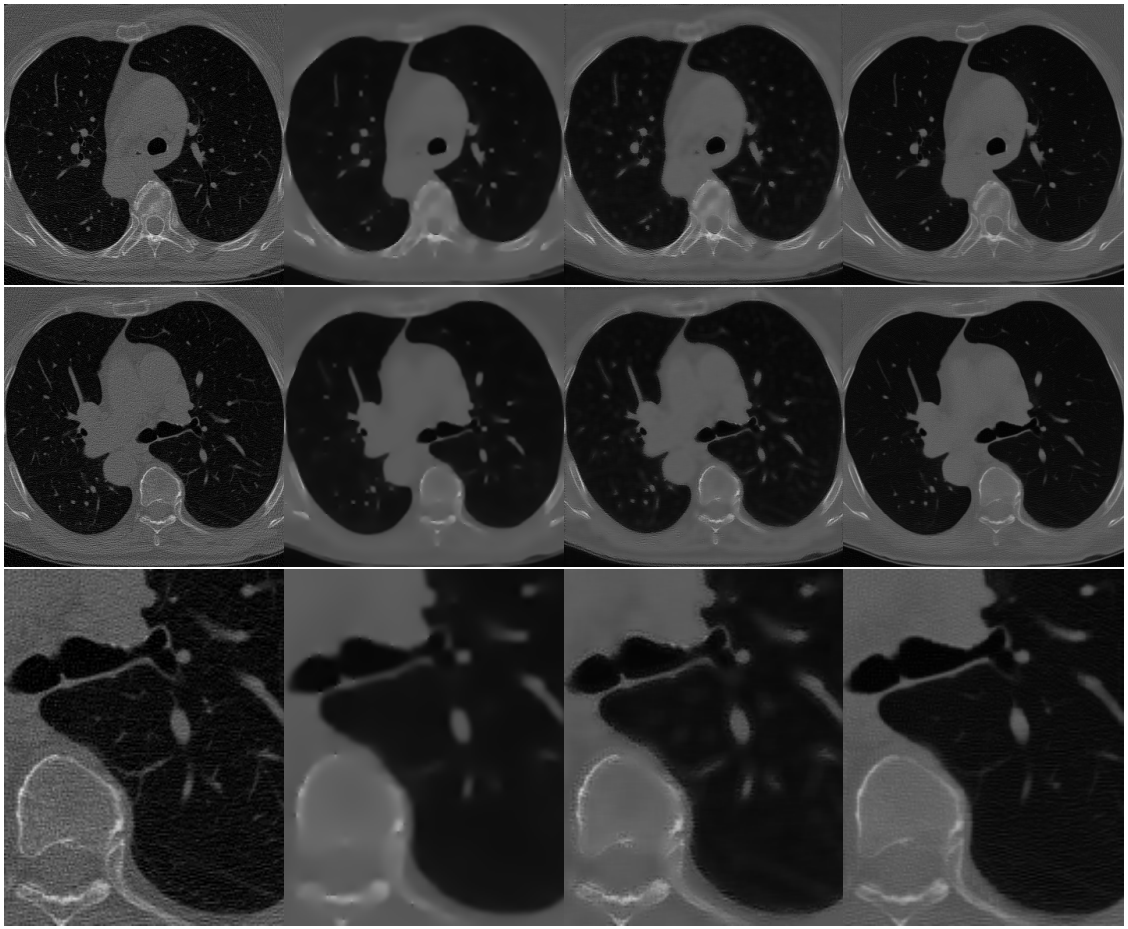


Figure 3.9: Non-synthetic (real) experiment. From left to right: real CT slice; adaptive multiscale image denoising algorithm [15]; wavelet domain image denoising algorithm [14]; result of the proposed denoising algorithm.

Table 3.5: PSNR and time comparison for using various matrix completion for lung slices

	Proposed method	**Denoising method using OptSpace [64]**
PSNR	21.83	21.65
Time (*seconds*)	118	1398

In Table 3.4, I present the PSNR results of the proposed denoising algorithm for a lung CT slices; where impulsive noise is changing. This table shows how my algorithm is robust in denoising of the corrupted slices by serious impulsive noise. In graph 3.7, I compare my denoising method with adaptive multiscale image denoising algorithm [15], wavelet domain image denoising algorithm [14], and non-local means algorithm [13], which are among the state-of-the-art in medical image denoising.

In this comparison, I apply my denoising method on lung slices which I changed the impulsive noise and kept the Poisson and Gaussian noise constant for all methods. My proposed method surpasses the other methods with a significant margin for all additive Impulsive Noise.

Evaluation On Non-Synthetic Slices

In this section, I turn to some real slices, where I apply my proposed method without any changes or generating noisy slices. Note that since there are no published methods that perform denoising on such general slices, I choose the adaptive multiscale image denoising algorithm [15] and wavelet domain image denoising algorithm [14] for comparison, because their source code is available. As for the non-synthetic case, while I do not have the ground truth and thus cannot evaluate the the methods quantitatively using PSNR, the visual comparison illustrates the robustness of my proposed method when it is applied directly to real slices. Note that I used the same slices number, block size, overlapped region, and extracted patches number

for the non-synthetic experiment just as the synthethic case.

Evaluation on 3D Images

I evaluate my framework on Lung data sets LIDC-IDRI [80] where the size of each slice of the CT slices are 512×512 pixels. In general, the evaluation and thus, 3D medical denoising methods is a difficult task. Usually, ground truth data for real noisy data - especially in medical applications - is not available. Therefore, I perform several experiments illustrating the potentials of my approach. In this section, I will illustrate some examples that will assess my 3D medical denoising method for 3D CT data. Comparison will be made against the state-of-the-art technique: PRI-NLM3D [16] by Manjón *et al.* because the source codes are available publicly.

All tests in this section were processed in the following manner: All 10 slices were involved in the denoised slice. The similar 3D block size used for block matching was $7 \times 7 \times 5$ and was not changed for various tests. I obtained a locally consistent solution by allowing 3D patches to overlap, where the overlapped regions v were 5 voxels in each direction. Further, for each reference 3D patch, I extract 4 most similar 3D patches used in whole 3D data using the proposed 3D block matching algorithm.

I first generate synthetic noisy 3D CT data and then apply denoising method to the degraded 3D CT data. The results are compared to the ground truth (the original images).

Fig. 3.10 shows the result of the my 3D medical denoising with a 3D perspective. In this figure, I decided to show only a part of the CT images because the inside details of the lung are more important than the tissue around it. Figs. 3.10(a), 3.10(b) are the real and the noisy CT image, respectively. Fig. 3.10(c) shows the result of 3D medical denoising where I used the voxels of the denoised 3D data.

As PRI-NLM3D [16] can not remove the mixed noise, for fair of comparison I

apply it on the 3D noisy data after removing Impulsive noise.

In Fig. 3.11, I show the PSNR result and a clear visual comparison on the CT Lung slices. The original CT slice is corrupted by a mixture of Poisson noise, Gaussian white noise, impulsive noise with significant noise level (variance of Gaussiance noise = 0.02 and noise density of impulsive noise = 0.01). As shown in the figure, PRI-NLM3D [16] removes the critical information at edge areas, while my proposed denoising method performs remarkably well for the detail structures and is free of these artifacts. To

To quantify my denoising performance, I used the Peak Signal to Noise Ratio (PSNR) measure between the ground truth and denoised CT slice.

Evaluation on Ultrasound

To evaluate the performance of the proposed methods, I now present some experimental results obtained by applying the proposed methods on some ultrasound images. In this section, I will illustrate some examples that will assess my denoising methods for real ultrasound images. Comparison will be made against the state-of-the-art technique: the wavelet domain image denoising algorithm [14].

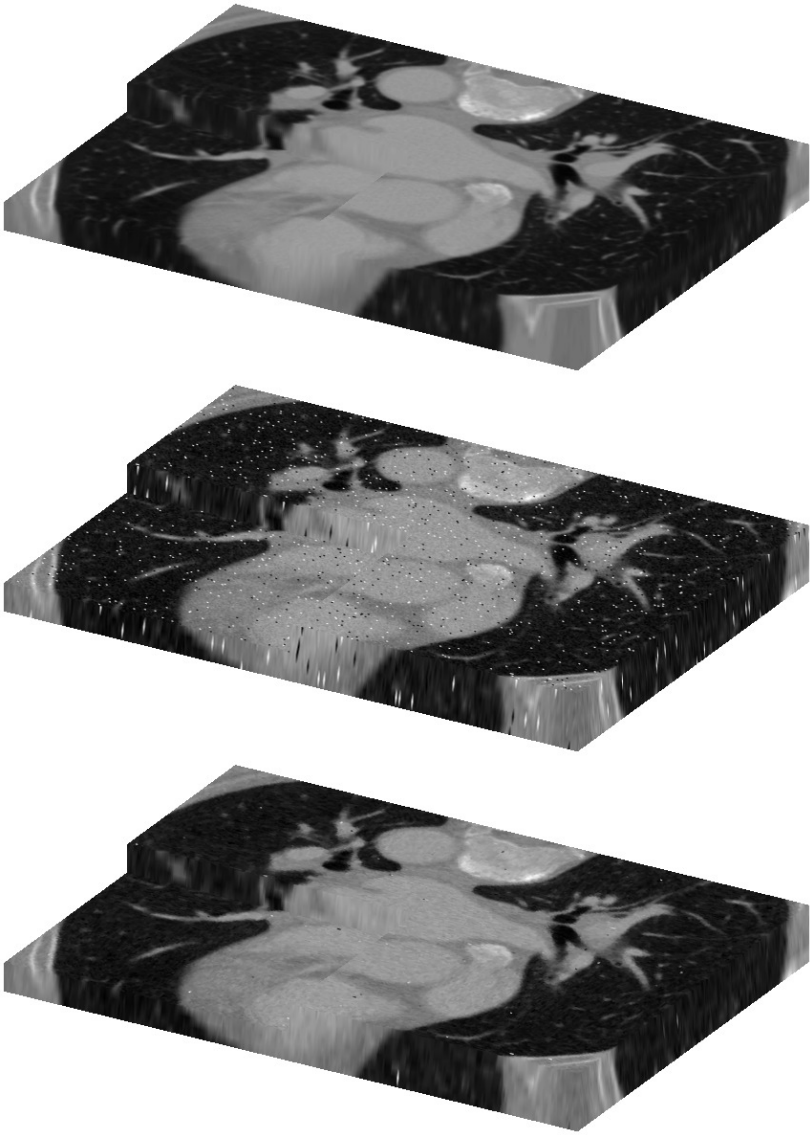


Figure 3.10: Experiment 1 (3D display). (a) real 3D CT data; (b) noisy 3D CT data; (c) the 3D result of my proposed denoising method.

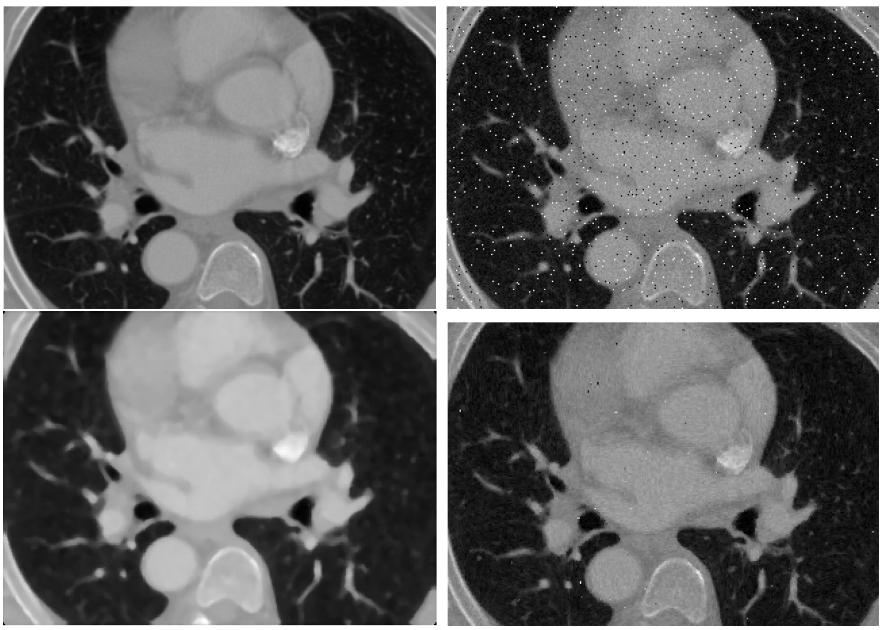


Figure 3.11: Experiment 2: (PSNR in brackets). (a) real CT data; (b) noisy CT data; (c) PRI-NLM3D [16] [25.28 dB]; (d) the proposed denoising method [29.16 dB].

Table 3.6: Time comparison for using various matrix completion

	Algorithm using Dense SCoBeP and matrix completion	**Algorithm** using Overlapped SCoBeP matrix completion	**Denoising method** using OptSpace [64]
Time (*seconds*)	410	120	1398

I also replaced my proposed decomposition matrix completion with OptSpace [64] to compare the result and time consumption (Table 3.6). It can be seen that my methods perform notably faster than OptSpace [64].

All tests in this section were processed in the following manner: All 30 images were involved in the denoised image. The similar block size used for block matching was 63×63 and was not changed for various tests. I obtained a locally consistent solution by allowing patches to overlap in denoising algorithm using Overlapped SCoBeP matrix completion, where the overlapped regions (v) were 5 pixels in each direction. Further, for each reference patch, I extract 3 most similar patches used in each image using SCoBeP.

In this work, I apply my proposed methods without any changes or generating noisy images. Note that since there are no published methods that perform denoising on such general images, I choose the wavelet domain image denoising algorithm [14] for comparison, because their source code is available. As for the non-synthetic case, while I do not have the ground truth and thus cannot evaluate the the methods quantitatively using PSNR, the visual comparison illustrates the robustness of my proposed methods when it is applied directly to real images.

As shown in the Figures 3.12 and 3.13, wavelet domain image denoising algorithm [14] generates severe artifacts at edge areas, while my proposed denoising methods perform remarkably well for the detail structures and are free of these artifacts.

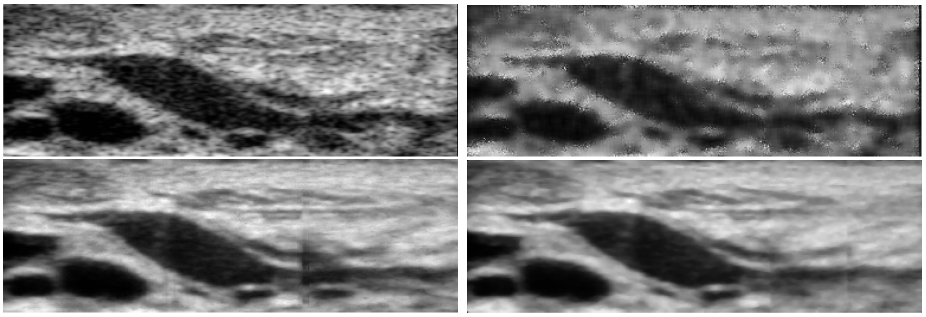


Figure 3.12: Non-synthetic (real) experiment. (a) real ultrasound image; (b) wavelet domain image denoising algorithm [14]; (c) the proposed denoising Algorithm using Dense SCoBeP and matrix completion; (d) the proposed denoising Algorithm using Overlapped SCoBeP matrix completion.

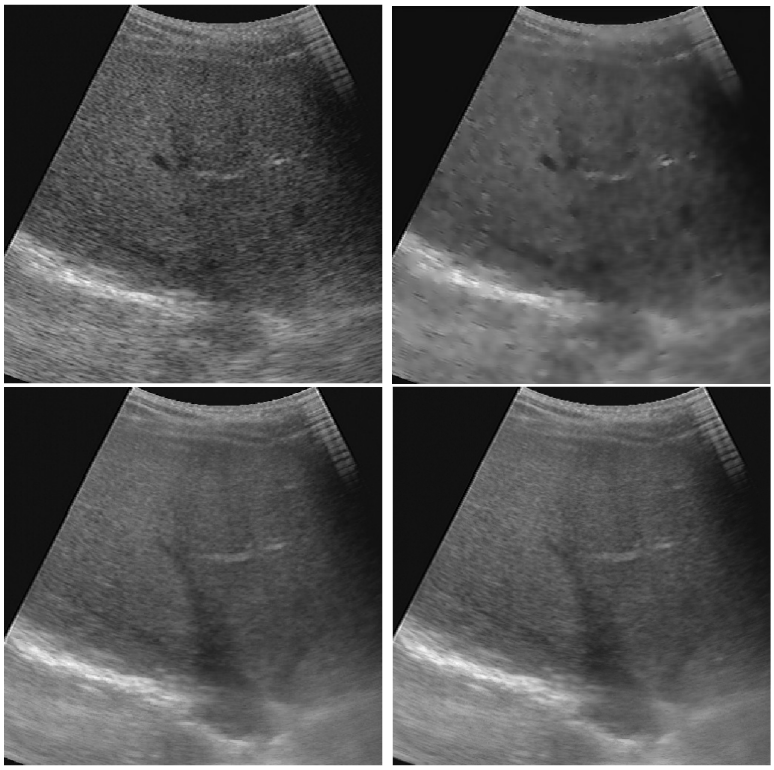


Figure 3.13: Non-synthetic (real) experiment. (a) real ultrasound image; (b) wavelet domain image denoising algorithm [14]; (c) the proposed denoising Algorithm using Dense SCoBeP and matrix completion; (d) the proposed denoising Algorithm using Overlapped SCoBeP matrix completion.

CHAPTER 4

CONCLUSION

This work explored several important aspect of image and video enhancment:

1. In conclusion, I have proposed two novel and efficient super resolution methods based on SCoBeP [1] and Nonlocal-Means (NLM) techniques, which finds corresponding patches using sparse coding and demonstrates competitive results in both the synthetic and the real sequences. My techniques perform super resolution by first running sparse coding over an overcomplete dictionary constructed from the LR frames to gather possible match candidates. Belief propagation is then applied to eliminate bad candidates and to select optimum matches. Finally, in the SCoBeP-NLM, the NLM approach exploits similarity in patches around candidate pixels to average out the noise among similar patches. While the algorithm performs favorably comparing with other recent approaches as illustrated in the experimental results, the algorithm is quite complex and I realized that the source of most computation is originated from the NLM component. As SCoBeP has naturally identified pixels that are most likely to be relevant to a target pixel and also output the corresponding "weight" of the relevant pixels. This suggested us that NLM is probably not essential in my SCoBeP based SR algorithm. Thus, I have also implemented a SCoBeP based SR algorithm, SCoBeP-SR, where "mixing" weights and candidates are extracted from the SCoBeP step only.

I conducted experiments on both the synthetic and the real video sequences, where my approaches work well for both types of sequences demonstrating the effectiveness and robustness of my approaches. Furthermore, unlike many existing super resolution approaches targeting to LR frames that have been pre-registered manually [34] or have assumed a stationary camera [5, 27], the proposed method can handle a sequence captured with a moving camera and do not require preprocessing of the sequence.

As SCoBeP provides decent results in images with both significantly and slightly varying viewpoints [1, 81], hence, it will be useful to a wide range of applications such as de-interlacing, surveillance application and medical image super resolution. As for future work, I plan to extend my approaches to these areas.

2. In addition, I have proposed a novel and efficient denoisinng method using block matching filtering and matrix compleition. A key idea of [24] is to divide and conquer. The input matrix is partitioned into clusters, and then each cluster is filled separately. A dependency scanning step estimates the lowest possible rank of each cluster by identifying independent rows. The unknown elements of these independent rows can then be filled arbitrarily without increasing the rank of the cluster. The remaining unknown elements of the clusters are filled by the dependency relationship obtained earlier. Finally, the "off-diagonal" elements are filled to ensure the entire matrix has the lowest possible rank.

   I proposed a block-based video denoising method using the decomposition approach [24], in which I keep only reliable pixels and eliminate all unreliable pixels. My denoising method can remove the serious mixed noise from video sequence, while most of the existing methods have been limited to one specific

type of noise. Quantitative and qualitative experiments with video sequences corrupted by mixed noise have shown that the proposed algorithm outperforms the state-of-the-art methods for the denoising tasks.

Bibliography

[1] N. Barzigar, A. Roozgard, S. Cheng, and P. Verma, "Scobep: Dense image registration using sparse coding and belief propagation," *Journal of Visual Communication and Image Representation*, 2012.

[2] M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the non-local-means to super-resolution reconstruction," in *IEEE Trans. Image Process*, 2009, p. 36.

[3] G. Wolberg, I. of Electrical, and E. E. C. Society, *Digital image warping*. IEEE computer society press California, 1990, vol. 3.

[4] H. Takeda, P. Milanfar, M. Protter, and M. Elad, "Super-resolution without explicit subpixel motion estimation," *IEEE Trans. Image Process*, vol. 18, no. 9, pp. 1958–1975, 2009.

[5] H. Zhang, J. Yang, Y. Zhang, and T. Huang, "Non-local kernel regression for image and video restoration," *Computer Vision–ECCV 2010*, pp. 566–579, 2010.

[6] S. Babacan, R. Molina, and A. Katsaggelos, "Variational bayesian super resolution," *Image Processing, IEEE Transactions on*, vol. 20, no. 4, pp. 984–999, 2011.

[7] S. Farsiu, M. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Trans. Image Process*, vol. 13, no. 10, pp. 1327–1344, 2004.

[8] P. Getreuer. (2009) http://www.getreuer.info/home/tvreg.

[9] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3d transform-domain collaborative filtering," in *Proc. 15th European Signal Processing Conference*, vol. 1, no. 2. Citeseer, 2007, p. 7.

[10] I. Selesnick and K. Li, "Video denoising using 2d and 3d dual-tree complex wavelet transforms," *Wavelets: Applications in Signal and Image Processing X*, vol. 5207, pp. 607–618, 2003.

[11] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm."

[12] K. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems," *preprint*, 2009.

[13] A. Buades, B. Coll, and J. Morel, "A non-local algorithm for image denoising," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 60–65.

[14] A. Pizurica, A. Wink, E. Vansteenkiste, W. Philips, and B. Roerdink, "A review of wavelet denoising in mri and ultrasound brain imaging," *Current medical imaging reviews*, vol. 2, no. 2, pp. 247–260, 2006.

[15] P. Bao and L. Zhang, "Noise reduction for magnetic resonance images via adaptive multiscale products thresholding," *Medical Imaging, IEEE Transactions on*, vol. 22, no. 9, pp. 1089–1099, 2003.

[16] J. Manjón, P. Coupé, A. Buades, D. Louis Collins, and M. Robles, "New methods for mri denoising based on sparseness and self-similarity," *Medical Image Analysis*, vol. 16, no. 1, pp. 18–27, 2012.

[17] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2272–2279.

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

[19] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process*, vol. 19, no. 11, pp. 2861–2873, 2010.

[20] J. Wang, S. Zhu, and Y. Gong, "Resolution enhancement based on learning the sparse association of image patches," *Pattern Recognition Letters*, vol. 31, no. 1, pp. 1–10, 2010.

[21] J. Maintz and M. Viergever, "A survey of medical image registration," *Medical image analysis*, vol. 2, no. 1, pp. 1–36, 1998.

[22] A. Roozgard, S. Cheng, and H. Liu, "Malignant nodule detection on lung ct scan images with kernel rx-algorithm," in *IEEE-EMBS International Conference on Biomedical and Health Informatics*, Hong Kong, Shenzhen, 2012.

[23] S. Ali, S. Vathsal, *et al.*, "A ga based window selection methodology to enhance window based multi wavelet transformation and thresholding aided ct image denoising technique," *Arxiv preprint arXiv:1003.1826*, 2010.

[24] R. Ma, N. Barzigar, A. Roozgard, and S. Cheng, "Decomposition approach for low-rank matrix completion and its applications," 2014.

[25] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[26] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *International Journal of Imaging Systems and Technology*, vol. 14, no. 2, pp. 47–57, 2004.

[27] M. Protter and M. Elad, "Super resolution with probabilistic motion estimation," *IEEE Trans. Image Process*, vol. 18, no. 8, pp. 1899–1904, 2009.

[28] R. Pan and S. J. Reeves, "Efficient huber-markov edge-preserving image restoration," *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3728–3735, 2006.

[29] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 460–489, 2005.

[30] H. Takeda, S. Farsiu, and P. Milanfar, "Deblurring using regularized locally adaptive kernel regression," *Image Processing, IEEE Transactions on*, vol. 17, no. 4, pp. 550–563, 2008.

[31] A. Buades, B. Coll, J. Morel, *et al.*, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2006.

[32] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *iccv*. Published by the IEEE Computer Society, 1999, p. 1033.

[33] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical models and image processing*, vol. 53, no. 3, pp. 231–239, 1991.

[34] J. Chan, J. Ma, P. Kempeneers, F. Canters, J. Vandenborre, and D. Paelinckx, "An evaluation of ecotope classification using superresolution images derived from chris/proba data," *IGARSS, July*, pp. 6–11, 2008.

[35] S. Bonchev and K. Alexiev, "Improving super-resolution image reconstruction by in-plane camera rotation," in *Information Fusion (FUSION), 13th Conference on*. IEEE, 2010, pp. 1–7.

[36] D. Mitzel, T. Pock, T. Schoenemann, and D. Cremers, "Video super resolution using duality based tv-l 1 optical flow," *Pattern Recognition*, pp. 432–441, 2009.

[37] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, "Image and video super-resolution via spatially adaptive block-matching filtering," in *Proceedings of International Workshop on Local and Non-Local Approximation in Image Processing (LNLA)*. Citeseer, 2008.

[38] R. Tsai and T. Huang, "Multiframe image restoration and registration," *Advances in computer vision and Image Processing*, vol. 1, no. 2, pp. 317–339, 1984.

[39] S. Kim, N. Bose, and H. Valenzuela, "Recursive reconstruction of high resolution image from noisy undersampled multiframes," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, no. 6, pp. 1013–1027, 1990.

[40] C. Liu and D. Sun, "A bayesian approach to adaptive video super resolution," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 209–216.

[41] G. Chantas, N. Galatsanos, and N. Woods, "Super-resolution based on fast registration and maximum a posteriori reconstruction," *IEEE Trans. Image Process*, vol. 16, no. 7, pp. 1821–1830, 2007.

[42] X. Li, K. Lam, G. Qiu, L. Shen, and S. Wang, "Example-based image super-resolution with class-specific predictors," *Journal of Visual Communication and Image Representation*, vol. 20, no. 5, pp. 312–322, 2009.

[43] C.-Y. Yang, J.-B. Huang, and M.-H. Yang, "Exploiting self-similarities for single frame super-resolution," in *Computer Vision–ACCV 2010*. Springer, 2011, pp. 497–510.

[44] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[45] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001.

[46] C. Liu and W. T. Freeman, "A high-quality video denoising algorithm based on reliable motion estimation," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 706–719.

[47] Q. B. Do, A. Beghdadi, and M. Luong, "Combination of closest space and closest structure to ameliorate non-local means method," in *Computational Intelligence for Multimedia, Signal and Vision Processing (CIMSIVP), 2011 IEEE Symposium on*. IEEE, 2011, pp. 134–141.

[48] S. Villena, M. Vega, R. Molina, and A. Katsaggelos, "Bayesian super-resolution image reconstruction using an l1 prior," in *Image and Signal Processing and Analysis, 2009. ISPA 2009. Proceedings of 6th International Symposium on*. IEEE, 2009, pp. 152–157.

[49] A. Yang, A. Ganesh, Z. Zhou, S. Sastry, and Y. Ma, "Fast l1-minimization algorithms and an application in robust face recognition: a review," in *Proceedings of the International Conference on Image Processing*, 2010.

[50] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *Information Theory, IEEE Transactions on*, vol. 55, no. 5, pp. 2230–2249, 2009.

[51] R. Maleh, A. Gilbert, and M. Strauss, "Sparse gradient image reconstruction done faster," in *IEEE International Conference on Image Processing, ICIP*, vol. 2, 2007, pp. 77–80.

[52] W. Burger and M. Burge, *Principles of digital image processing: core algorithms*. Springer-Verlag New York Inc, 2009.

[53] R. Keys, "Cubic convolution interpolation for digital image processing," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, no. 6, pp. 1153–1160, 1981.

[54] N. Srebro, "Learning with matrix factorizations," Ph.D. dissertation, Citeseer, 2004.

[55] E. Candes and Y. Plan, "Matrix completion with noise," *Arxiv preprint arXiv:0903.3131*, 2009.

[56] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[57] T. Graepel, "Kernel matrix completion by semidefinite programming," *Artificial Neural NetworksICANN 2002*, pp. 141–142, 2002.

[58] J. Abernethy, F. Bach, T. Evgeniou, and J. Vert, "Low-rank matrix factorization with attributes," *arXiv preprint cs/0611124*, 2006.

[59] Y. Amit, M. Fink, N. Srebro, and S. Ullman, "Uncovering shared structures in multiclass classification," in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, vol. 24, 2007, p. 17.

[60] A. Singer, "A remark on global positioning from local distances," *Proceedings of the National Academy of Sciences*, vol. 105, no. 28, p. 9507, 2008.

[61] R. Schmidt, "Multiple emitter location and signal parameter estimation," *Antennas and Propagation, IEEE Transactions on*, vol. 34, no. 3, pp. 276–280, 1986.

[62] E. Candes and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[63] E. Candes and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *arXiv*, vol. 903, 2009.

[64] R. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *Information Theory, IEEE Transactions on*, vol. 56, no. 6, pp. 2980–2998, 2010.

[65] A. Waters, A. Sankaranarayanan, and R. Baraniuk, "Sparcs: Recovering low-rank and sparse matrices from compressive measurements," Technical report, Rice University, Houston, TX, Tech. Rep., 2011.

[66] J. Cai, E. Candes, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *preprint*, 2008.

[67] K. Lee and Y. Bresler, "Admira: Atomic decomposition for minimum rank approximation," *arXiv*, vol. 905, 2009.

[68] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization," *Mathematical Programming*, pp. 1–33, 2009.

[69] W. Dai and O. Milenkovic, "Set: an algorithm for consistent matrix completion," *Arxiv preprint arXiv:0909.2705*, 2009.

[70] R. Meka, P. Jain, and I. Dhillon, "Guaranteed rank minimization via singular value projection," 2009.

[71] S. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on Image Processing*, pp. 1532 – 1546, September 2000.

[72] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on.* IEEE, 1998, pp. 839–846.

[73] M. Orchard and G. Sullivan, "Overlapped block motion compensation: An estimation-theoretic approach," *Image Processing, IEEE Transactions on*, vol. 3, no. 5, pp. 693–699, 1994.

[74] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust video denoising using low rank matrix completion," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on.* IEEE, 2010, pp. 1791–1798.

[75] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 3, no. 2, pp. 148–157, 1993.

[76] Y. Nie and K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation," *Image Processing, IEEE Transactions on*, vol. 11, no. 12, pp. 1442–1449, 2002.

[77] H. Hwang and R. Haddad, "Adaptive median filters: new algorithms and results," *Image Processing, IEEE Transactions on*, vol. 4, no. 4, pp. 499–502, 1995.

[78] S. Zimmer, S. Didas, and J. Weickert, "A rotationally invariant block matching strategy improving image denoising with non-local means," in *Proc. 2008 International Workshop on Local and Non-Local Approximation in Image Processing*, 2008, pp. 135–142.

[79] N. Barzigar, A. Roozgard, P. Verma, and S. Cheng, "A video super resolution framework using scobep," *IEEE Transactions on Circuits and Systems for Video Technology*, 2013.

[80] N. C. Institute, "The cancer imaging archive," https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI, September 2011. [Online]. Available: https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI

[81] A. Roozgard, N. Barzigar, S. Cheng, and P. Verma, "Medical Image registration using sparse coding and belief propagation," in *The 34th Annual IEEE International Conference of the Engineering in Medicine and Biology Society, San Diego.*, 2012.

DEDICATION

to

My parents and Amin

For

Encouraging me to follow my dreams