

RECONSTRUCTION MODELS TO IMPROVE IMAGE QUALITY

A Dissertation

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Emil Bilgazyev

December 2013

RECONSTRUCTION MODELS TO IMPROVE IMAGE QUALITY

Emil Bilgazyev

APPROVED:

Ernst L. Leiss, Chairman
Professor, Computer Science Department
University of Houston

Marc Garbey
Professor, Computer Science Department
University of Houston

Olin Johnson
Professor, Computer Science Department
University of Houston

Nikolaos Tsekos
Associate Professor, Computer Science Department
University of Houston

Dipan Shah
Assistant Professor
Weill Cornell Medical College

Dean, College of Natural Sciences and Mathematics

ACKNOWLEDGMENTS

First I would like to express profound gratitude to my adviser Professor Ernst Leiss for all guidance, advice, and support that he has given me during my Ph.D. studies at the Computer Science Department at the University of Houston. I have very much enjoyed working with him, and I have learned a lot from him; it will benefit me for my whole life.

I also would like to thank Dr. Nikolaos Tsekos for his suggestions and support to accomplish the MRI project. I was deeply impressed and inspired by his eagerness to find new approaches and solutions for a single problem, and by learning all approaches, I learned more and more. In addition, I would also thank Dr. Dipan Shah from Methodist DeBakey Heart and Vascular Center, Houston, TX for this idea of the reconstruction of MRI images, and for providing the dataset of MRI images.

Finally, I want to thank my mother, Giunlara Asanova, my grandparents Burulbu Shaganova and Orozbay Asanov for their encouragement for me to come to U.S.A. and pursue a Ph.D. I would like to give a special thanks to my wife, Munara Tolubaeva, for being very supportive and patient during the course of the Ph.D. studies. I am very happy to have her as my wife. I also want to thank my sisters Nadira and Nagima, my father Bilgazy, and brother Ahmet Kus for their support.

RECONSTRUCTION MODELS TO IMPROVE IMAGE QUALITY

An Abstract of a Dissertation
Presented to
the Faculty of the Department of Computer Science
University of Houston

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

By
Emil Bilgazyev
December 2013

Abstract

High-resolution images provide more information for better post-processing such as detection, recognition, segmentation, identification, and visualization. The need for high-resolution images occurs in health care, where the physician needs a high-quality image of the patient in order to make better decisions or to perform surgery. Breath-holding MRI scanners have high-acquisition speed and collect a large number of low-quality frames; similarly, some surveillance cameras have low acquisition speed and collect low quality images, due to the storage space restrictions or limited network bandwidth to transfer the data. It is hard to perform satisfactory image processing on low-quality images. Image reconstruction models, e.g., multi-frame fusion, and single image super-resolution, have been successfully used in image processing and computer vision to improve the quality of the image. Many algorithms have been proposed to fuse multiple low-quality images in order to get a single high-resolution image, or to train the model on the training images and to use that model to improve the quality of the single input image. The goal of this dissertation is to study previous approaches related to image quality, find their limitations, and introduce new approaches to solve them. Since it is difficult to design a single algorithm that will work for all types of images, such as MRI images and images obtained by surveillance cameras, we divide the problem into sub-problems. We introduce new algorithms to address the following objectives: *(i)* to fuse multiple low-resolution frames acquired by an MRI scanner, *(ii)* to improve the quality of a single image by adding information from training images, and *(iii)* to perform better recognition on facial images.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives	3
1.2.1	Objective I: Multi-frame Reconstruction from MRI images	3
1.2.2	Objective II: Speed-up Super-resolution Reconstruction	4
1.2.3	Objective III: Face Recognition	6
1.3	Organization of the Dissertation	7
2	Objective I: Cardiac MRI Reconstruction	8
2.1	Methodology	10
2.2	Implementation Details and Results	13
2.2.1	Segmentation	14
2.2.2	Signal Extraction and Denoising	19
2.2.3	Reconstruction	26
2.2.4	Experimental Results	35
3	Objective II: Speed-up Super-resolution	42
3.1	Introduction	42
3.2	Methodology	46
3.3	Implementation Details and Results	50

3.3.1	Space Reduction	50
3.3.2	Wavelet Decomposition of Patches	52
3.3.3	Color Images	53
3.3.4	Parallelization of the Algorithm	54
3.3.5	Experiment Details	56
3.3.6	Experimental Results	57
4	Objective III: Face Recognition	62
4.1	Introduction	62
4.2	Methodology	65
4.3	Implementation Details and Experimental Results	69
4.3.1	Experimental Databases	70
4.3.2	Preprocessing	75
4.3.3	Experimental Results	75
5	Conclusion	79
5.1	Future Work	82
6	Publications during Ph.D. Study	83
	Bibliography	85

List of Figures

2.1	MRI project pipeline	10
2.2	Depiction of MR image	13
2.3	Segmentation of MR images	15
2.4	Segmentation of MR images	16
2.5	Segmentation of MR images	17
2.6	Gradient of the image	18
2.7	Difference between consecutive frames	19
2.8	Noisy signal	20
2.9	Wavelet transform	22
2.10	Wavelet transform	23
2.11	Scaling and wavelet functions	24
2.12	The output of wavelet transform	25
2.13	Torso region for tracking	26
2.14	Tracking the gradient change	27
2.15	Depiction of tracking points for different frames	28
2.16	Extracted breath rate and heart rate signals	29
2.17	Extracted heart beat and breath rate signal from MR images	30
2.18	Breath rate signal	31
2.19	Selection of a frames	31
2.20	Depiction of local maxima and neighboring frames	32

2.21	Segment selection	33
2.22	Depiction of synthetic dataset	34
2.23	Depiction of region of interest	36
2.24	Computation time	37
2.25	Region of interest computation	38
2.26	Depiction of region of interest	39
2.27	MR image reconstruction	40
2.28	GUI implementation	41
3.1	Super-resolution framework	43
3.2	From high resolution to compact dictionary	46
3.3	Pipeline for SR	48
3.4	Gradients of patches	51
3.5	Wavelet decomposition	52
3.6	From RGB to YCbCr	54
3.7	Depiction of Berkeley Segmentation Dataset 500	56
3.8	Output of SR algorithms	59
3.9	Depiction of zoomed SR outputs	60
4.1	Face Recognition pipeline	64
4.2	Face99 dataset	71
4.3	YFace dataset	72
4.4	Depiction of Active Shape Model based landmark detection	73
4.5	Sparse representation of input image	74
4.6	Weight parameters	76
4.7	Face recognition results	78

List of Tables

3.1	Notations used in this chapter	47
3.2	Experimental results	60
4.1	Notations used in this chapter	66

Chapter 1

Introduction

1.1 Motivation

In image processing, there is a high demand for the input image to be processed to be high-resolution, due to the fact that when the input image has high-resolution, more information can be retrieved from the image for post processing such as identification, verification, recognition, segmentation, detection, etc. For example, nowadays, when we go to the Department of Public Safety to get a driving license, the clerk asks the applicant to remove glasses, scarf, etc., and stand still with no expression to take the picture of the applicant. Then, the specific type of camera takes the high-resolution picture of the applicant where it is stored in the database. The purpose of taking a high-resolution image of the applicant is not just to print it on the ID, it also has other purposes, such as: detecting license and ID frauds from the facial images (duplicates under different name), recognition during criminal investigations, etc. [30]. Similar cases can be found at airports. Another example for

the need for high-resolution are traffic violators. When the input image has low-resolution, we might identify the make, model and the color of the car, which is not enough to find the violators, but if we have a high-resolution image of the car, we might get license plate information, where each car has a unique license plate. Different examples for the need for high-resolution image can be found in health care, where the doctor needs a better image quality of the patient in order to make better decision, or perform surgery [45]. Similar cases can be found in dentistry, where the dentist needs to see the depth of a cavity from the X-ray image before he/she starts the treatment. Many examples can be found for the need for high-resolution image for processing. However, there are cases where we cannot acquire high-resolution images, such as in surveillance cameras. There are many reasons for the acquired image from the surveillance camera to be low-resolution [50]. One of them is due to the network bandwidth to transfer frames from the camera to the storage device. Another factor is the storage, where high-resolution images will require larger storage space. Assuming that some criminal activity was captured on one of the low-resolution surveillance cameras, this is sometimes the reason why we see on the news that if we know or can identify the person on the low-quality picture, we should call the police. When we enlarge the low-resolution image, then the enlarged image will result in a blurred image, and it will pose more difficulty for a person being to identify [14]. In such cases, preprocessing of the image is required in order to improve the quality of the image.

Super-resolution reconstruction is an approach that enhances the resolution of an image [2]. Resolution can have many meanings such as: number of independent pixels, total number of pixels, covered distance (aerial imaging), number of photons (microscopy), etc., and there is no generic super-resolution algorithm that could solve all constraints. Thus,

most of the super-resolution algorithms are domain specific [76].

1.2 Objectives

In this section we will describe the objectives, and briefly explain the proposed solutions for the objectives.

Super-resolution (SR) is divided into two categories: multi-frame super-resolution and single image super-resolution. In multi-frame-based SR, multiple frames are fused to obtain a single super-resolution image. Single-frame-based super-resolution uses training images to obtain higher-resolution images.

1.2.1 Objective I: Multi-frame Reconstruction from MRI images

Cardiac gating or breath-hold MRI acquisition is challenging. In particular, data collected in a short amount of time might be insufficient for the diagnosis of patients with impaired breath-holding capabilities and/or arrhythmia. A major challenge in cardiac MRI is the motion of the heart itself, the pulsate blood flow, and the respiratory motion. Furthermore, the motion of the diaphragm in the chest moving up and down gets translated to the heart when a patient breathes. Therefore, artifacts arise due to the changes in signal intensity or phase as a function of time, resulting in blurry images.

Assuming that we can collect multiple frames of MRI images of an internal organ over time, we need an algorithm to fuse these multiple frames to obtain a better quality image. This will be our first objective. For the first objective, we propose a novel reconstruction

strategy for real-time cardiac MRI without requiring the use of an electro-cardiogram or of breath holding [8]. In this research we focus on automation and evaluation of the performance of our proposed method in real-time MRI data to ensure a good basis for the signal extraction. Hence, it assists in the reconstruction. The proposed method enables one to extract cardiac beating waveforms directly from real-time cardiac MRI series collected from freely breathing patients and without cardiac gating. Our method only requires minimal user involvement as an initialization step. Thereafter, the method follows the registered area in every frame and updates itself. We evaluate different basis functions, varying levels of decomposition, and multiple methods for coefficient fusion, so as to maximize the benefit of the better quality image for the task of higher MRI image acquisition. We have used the ordinary Discrete Wavelet Transform (DWT) to denoise the signal, and K-Spaces of the image in order to fuse them. To evaluate them, we used a synthetic dataset. Experiments are designed and results reported across both synthetic datasets and real MRI data acquired from five different patients in order to determine the limitations of multi-frame diastole/systole reconstructions.

1.2.2 Objective II: Speed-up Super-resolution Reconstruction

Some of the surveillance cameras have low acquisition speed (frames per second), due to the storage space to store or the bandwidth of the network to transfer the data. Thus, usually we obtain only a few frames of an object or a subject. We need an algorithm to estimate a higher-resolution image from a single-frame image. This will be our second objective. Obtaining a super-resolution image from a single low-resolution image is a difficult task

and will require additional information during reconstruction. Most of the single-frame-based super-resolution algorithms utilize training images to train an algorithm, or build a dataset that can be used during reconstruction.

We propose a novel approach to obtain super-resolution images from low-resolution images with the goal of improving the quality of the input image [9]. Our approach is motivated by the observation that high-frequency information in images plays a significant role in the formation of the high-resolution image. However, it is not sufficient by itself to get higher-resolution. Therefore, we propose to recover an SR image of a given low-resolution image by adding high-frequency components from the training datasets.

When the parameters of the camera settings (point-spread-function, (PSF), motion, and pose) of the input LR image are known *a priori*, then specific databases can be used to create an SR image. By this method, a better SR output is accomplished, since there are fewer constraints to satisfy. If there is no information about camera settings, motion, and pose, then the training database should be very large to include all images for various constraints. Thus, it will be difficult to estimate an SR image, as well as computationally expensive, when there are too many unknowns. However, instead of reconstructing the full image, one can reconstruct the SR patches and stitch them together to obtain a full SR image.

Our proposed algorithm reconstructs a super-resolution image from a given low-resolution image, by adding high-frequency information that is extracted from natural high-resolution images in the training dataset. The selection of the high-frequency information from the training dataset is accomplished in two steps: a nearest-neighbor search algorithm is used to select the closest images from the training dataset, which can be implemented in the

GPU, and a sparse-representation algorithm is used to estimate a weight parameter to combine the high-frequency information of selected images. The weights are determined through manifold learning via sparse representation techniques. This simple but very powerful super-resolution algorithm can produce state-of-the-art results.

1.2.3 Objective III: Face Recognition

Face recognition is a challenging research topic, especially when the training (gallery) and recognition (probe) images are acquired using different cameras under varying conditions. Even a small noise or occlusion in the images can compromise the accuracy of recognition. Lately, sparse encoding-based classification algorithms have given promising results for such uncontrollable scenarios. We propose a novel methodology by modeling the sparse encoding with weighted patches to increase the robustness of face recognition even further [10]. In the training phase, we define a mask (i.e., weight matrix) using a sparse representation selecting the facial regions, and in the recognition phase, we perform comparison on selected facial regions. The algorithm is evaluated both quantitatively and qualitatively using two comprehensive facial image databases, i.e., Face99 [55] and YFace [19], with the results clearly superior to common state-of-the-art methodologies in different scenarios.

1.3 Organization of the Dissertation

The rest of the dissertation is organized as follows: the proposed algorithm for MR image reconstruction to obtain better quality images is presented in Chapter 2. The new extension to the super-resolution algorithm is presented in Chapter 3. The proposed face-recognition method is presented in Chapter 4. Finally, Chapter 5 concludes the dissertation with our findings.

Chapter 2

Objective I: Cardiac MRI Reconstruction

In this chapter we describe a novel reconstruction strategy for real-time cardiac MRI without requiring the use of an electro-cardiogram or of breath holding. The proposed method enables one to extract cardiac beating waveforms directly from real-time cardiac MRI series collected from freely breathing patients and without cardiac gating. Our method only requires minimal user involvement as initialization step. Thereafter, the method follows the registered area in every frame and updates itself.

Cardiac gating or breath-hold MRI acquisition is challenging. In particular, data collected in a short amount of time might be insufficient for the diagnosis of patients with impaired breath-holding capabilities and/or arrhythmia [77]. A major challenge in cardiac MRI is the motion of the heart itself, the pulsate blood flow, and the respiratory motion [46]. Furthermore, the motion of the diaphragm in the chest moving up and down

gets translated to the heart when a patient breathes. Therefore, artifacts arise due to the changes in signal intensity or phase as a function of time, resulting in blurry images [66]. In our research we focused on automation and evaluation of the performance of our proposed method in real-time MRI data to ensure a good basis for the signal extraction [78]. Hence, it assists in the reconstruction.

MRI is a safe, noninvasive test that creates detailed images of organs and tissues [38]. Cardiac MRI creates both still and moving images of the heart and major blood vessels [18, 36, 48]. To overcome the challenges due to motion, two approaches are being commonly used [28, 41].

The first approach uses the breath-hold acquisition. In this acquisition technique, the MRI technician collects data only when patient holds his breath [18]. Hence, one can only collect MRI data for a short amount of time. However, the collected data might not be sufficient for a physician to perform proper diagnosis. The second approach refers to a technique called “gating” [69]. The system identifies the quiet period between breaths and then outputs a trigger each time the heart beats. The moment the patient starts to take a breath, the system stops triggering and waits for the next quiet period to improve the quality of the scans [27]. In order to get viable signals, patients need to be outfitted with probes. For instance, the cardiac motion can be synchronized using the ECG. However, using ECG inside the MRI machine can be challenging since the magnetic field induces differential voltages in the leads of the probes [18, 36].

Several works [48, 25, 57] have been done recently to alleviate the motion artifacts and deterioration of image quality. The method to extract the surface area reported in [79] has

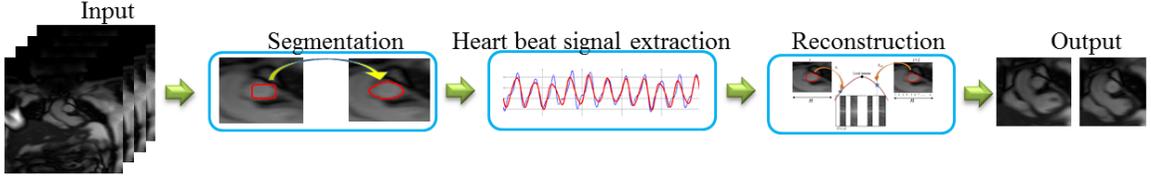


Figure 2.1: Depiction of a proposed pipeline.

a higher computational load and works well with cine MRI, but has limitations in real-time MRI due to the noise effects, and does not provide a statistical evaluation. In this work, we propose an approach that is more efficient than the combination of level set and wavelet decomposition [65]. Our proposed beat reconstruction pipeline takes multiple stacks of cardiac MRI sequences acquired from a freely breathing patient as an input, then performs segmentation for each image, and extracts the heart beat signal; then the reconstruction is performed to get systole and diastole images to guide the reconstruction (Figure 2.1).

2.1 Methodology

Let $x_i \in \mathcal{R}^{m \times n}$ be an i^{th} MR image of an image sequence $\mathcal{X} = \{x_0 \dots x_N\}$, $s \in \mathcal{R}^{N \times 1}$ be a heart beat signal corresponding to the image sequence \mathcal{X} , and y be an image that represents the exact systole or diastole position which we need to reconstruct from \mathcal{X} . To estimate y from the given image set \mathcal{X} and a heart beat signal s , the Bayesian equation can be utilized [12]:

$$P(y|s, \mathcal{X}) = \frac{P(s, \mathcal{X}|y)P(y)P(\mathcal{X})}{P(\mathcal{X}|s)P(s)}, \quad (2.1)$$

where $P(s, \mathcal{X}|y)$ is the likelihood of getting the signal and the sequence from a given image y , $P(s)$ is the prior knowledge of the heart beat signal, $P(y)$ is the prior knowledge

of the exact diastole/systole image, and $P(\mathcal{X}|s)$ is the likelihood of getting the sequence from the heart beat signal. Let us take the negative logarithm ($-\ln(\cdot)$) on both sides of Eq. 2.1, and denote $\ln P(y|s, \mathcal{X})$ by $\mathcal{L}(y, s, \mathcal{X})$, then Eq. 2.1 can be written as:

$$\mathcal{L}(y, s, \mathcal{X}) \approx \ln P(s, \mathcal{X}|y) + \ln P(y) + \ln P(\mathcal{X}) - \ln P(\mathcal{X}|s) - \ln P(s) , \quad (2.2)$$

where $\mathcal{L}(\cdot)$ is the function that we need to minimize. The likelihood $P(\mathcal{X}|s)$ can be estimated as:

$$P(\mathcal{X}|s) \approx \prod_{i=0}^N P(\mathcal{X}_i)P(s_i|\mathcal{X}_i) , \quad (2.3)$$

where the prior knowledge $P(\mathcal{X}_i)$ and the likelihood $P(s_i|\mathcal{X}_i)$ are written as:

$$P(\mathcal{X}_i) = e^{-\|E_{ext}(f_{\mathcal{X}_i}) - E_{int}(f_{\mathcal{X}_i})\|_2^2 / \sigma_2^2} , P(s_i|\mathcal{X}_i) = e^{-\|s_i - f(\mathcal{X}_i)\|_2^2 / \sigma_1^2} . \quad (2.4)$$

In Eq. 2.4, $E_{int}(\cdot)$ and $E_{ext}(\cdot)$ are the internal and external energy [52] of the function (\cdot) , $f_{\mathcal{X}_i}$ is the contour that encircles the region of interest in the MR image \mathcal{X}_i , and $\sigma_{1,2}$ are the distribution values ($\|\sigma_{1,2}\|_2^2 > 0$). Since in Eq. 2.4, we are estimating the heart beat signal from the image sequence \mathcal{X} , we have a priori knowledge of the signal, and we know that it should be periodic [5]. The output signal can be filtered out with a band pass filter [55], but a major problem is that we can't design such a filter without knowing the cutoff frequencies. Here is an alternative solution:

$$P(s) = \prod_{i=0}^N P(s_i) = \prod_{i=0}^N e^{-\|\mathcal{W}(s_i)\|_2^2 / \sigma_3^2} , \quad (2.5)$$

where $\mathcal{W}(\cdot)$ is the wavelet denoising function [12,13], and σ_3 is the distribution value. Using wavelet denoising we remove the high variation signal and preserve the low variation signal. To formulate the likelihood $P(s, \mathcal{X}|y)$, we need to define the function that

can extract the image y from the given image sequence \mathcal{X} and heart rate signal s . Assuming that during the MR image acquisition, first raw $k - space$ data is collected for each image \mathcal{X}_i , and then forward-FFT with post-processing is performed to acquire the *dicom* images [19, 44]. Given the raw data (k-space columns) of an image sequence \mathcal{X} and heart beat signal s , we can select the $k - space$ columns that fall into the peak of the heart rate signal s , and combine them ($\mathcal{G}(\mathcal{X}_i, s_i)$ in Eq. 2.6). Let $r(s_i)$ be a function that returns a mask vector ($[0 \dots 1 \dots 0]$), which is used to select the columns ($k - space$ columns) of 2D Fourier Transformed \mathcal{X}_i^* (Eq. 2.7):

$$\mathcal{G}(\mathcal{X}_i, s_i) = \bigcup_{i=0}^N \mathcal{X}_i^* r(s_i) \quad (2.6)$$

$$\mathcal{X}_i^*(u, v) = \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} \mathcal{X}_{k,l}^i e^{-j2\pi(uk/K+vl/L)} \quad (2.7)$$

The likelihood $P(s, \mathcal{X}|y)$ can be written as:

$$P(s, \mathcal{X}|y) = \prod_{i=0}^N P(s_i, \mathcal{X}_i|y) = e^{\|\mathcal{G}(\mathcal{X}_i, s_i) - y\|_2^2 / \sigma_4^2} \quad (2.8)$$

By combining Eq. 2.2, 2.4, 2.5, 2.6 and 2.8, the minimization function will be:

$$y = \arg \min_{y, f(\mathcal{X}_i), s} \left\| \bigcup_{i=0}^N \mathcal{X}_i^* r(s_i) - y \right\|_2^2 + \|s_i - f(\mathcal{X}_i)\|_2^2 + \left\| E_{ext}(f\mathcal{X}_i) - E_{int}(f\mathcal{X}_i) \right\|_2^2 + \left\| \sum_{i=0}^N w(s_i) \right\|_2^2 \quad (2.9)$$

It is hard to minimize the function in Eq. 2.9 [58]. However, we can minimize functions by separating them into multiple small minimization functions based on the arguments, $\partial\mathcal{L}/\partial y$:

$$y = \arg \min_y \left\| \bigcup_{i=0}^N \mathcal{X}_i^* r(s_i) - y \right\|_2^2, \quad (2.10)$$

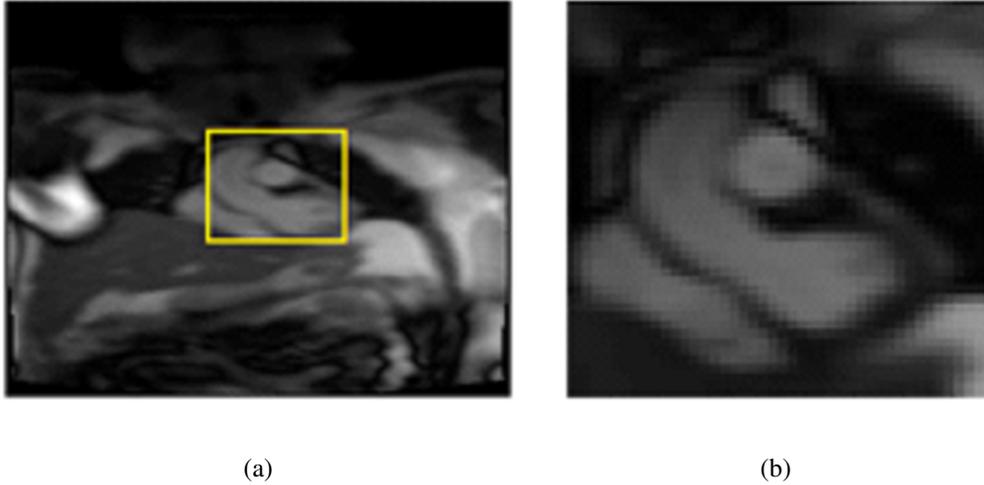


Figure 2.2: Depiction of MRI image. a) Original MRI image, and b) the yellow region is zoomed for better visualization.

and s can be estimated from:

$$s = \arg \min_{f_{\mathcal{X}_i}, s} \left\| E_{ext}(f_{\mathcal{X}_i}) - E_{int}(f_{\mathcal{X}_i}) \right\|_2^2 + \|s_i - f_{\mathcal{X}_i}\|_2^2 + \left\| \sum_{i=0}^N \mathcal{W}(s_i) \right\|_2^2. \quad (2.11)$$

2.2 Implementation Details and Results

As discussed above, first we need to extract the heart beat signal from the MR image sequence (Figure 2.2). We can break Eq. 2.11 into three parts, segmentation, signal extraction and denoising, and reconstruction. In each subsections we will describe the details of the implementation.

2.2.1 Segmentation

Segmentation of an object can be divided into two categories: edge-based segmentation [35] and region-based segmentation [23]. In general, edge-based models rely on the quality of the edges on the bounds of the object, where these edges are used to attract the contour. On the other hand, the region-based segmentation has a better performance than edge-based segmentation, since it relies on the intensity homogeneity; in other words, region-based segmentation assumes the image intensities within the region are homogeneous [35].

In order to segment the regions of interest, we used the Chan-Vesse algorithm proposed by Chan *et al.* [13]. This algorithm is a type of active contours to detect or segment objects in a given image, based on techniques of curve evolution, the MumfordShah problem [17]. The most important aspect of this model is that with it we can segment the regions whose boundaries are not necessarily defined by gradient. In MRI images to be able to stop at the boundaries is the most challenging part.

The global fitting term of Chan-Vesse algorithm is:

$$f(c^0, c^1, C) = \lambda_0 \int_{in} (\mathcal{X}_i - c_{x,y}^0)^2 dx dy + \lambda_1 \int_{out} (\mathcal{X}_i - c_{x,y}^1)^2 dx dy + \lambda_2 |C| \quad (2.12)$$

where C is the contour of segmentation, \mathcal{X}_i is an MRI image, c^0 and c^1 are two constants that best approximate the image intensities inside ($in(C)$) and outside ($out(C)$) of contour C , and $|C|$ is the length of the contour C . The terms λ_0 , λ_1 , and λ_2 are nonnegative constant values. They are used as regularization parameters. Eq. 2.12 can be solved using the energy minimization algorithm [13], where the function is iteratively minimized until it reaches a stop criterion or the maximum number of iterations. The contour C_i at different

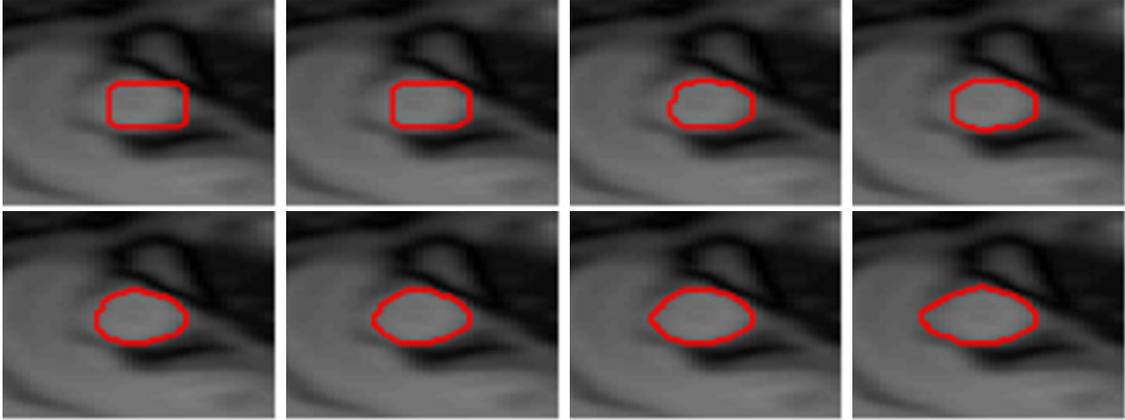


Figure 2.3: Depiction of segmentation at iteration: 1, 2, 3, 4, 7, 10, 15, 149. In the first iterations the contour grows fast, which is the approximation of the segmentation region, and in the latest iterations the contour grows slow, which is fine tuning the segmented region.

iterations for the frame \mathcal{X}_i is depicted in Figure 2.3.

The first term $\int_{in} (\mathcal{X}_i - c_{x,y}^0)^2 dx dy$ in the Eq. 2.12 is called global fitting energy. This is the term which makes the model not sensitive to the seed point (Figure 2.4). Figure 2.4(a) depicts the 16th frame of the MRI image sequence. The resolution of the image is 254×184 pixels, and we selected the region of interest (box marked in white in Figure 2.4(a)); Figure 2.4(b) depicts the selected region. In this figure, the red round markers and green star marker are the seed points for the segmentation experiment, to test if it will depend the initial seed position in the image.

The segmentation outputs of the red round markers in Figure 2.4 are depicted in Figure 2.5(a)-e, and the segmentation output of the star marker in Figure 2.4(b) is depicted in Figure 2.5(f). As one can see, the segmentation output of the different seed points within the boundary of the object is the same.

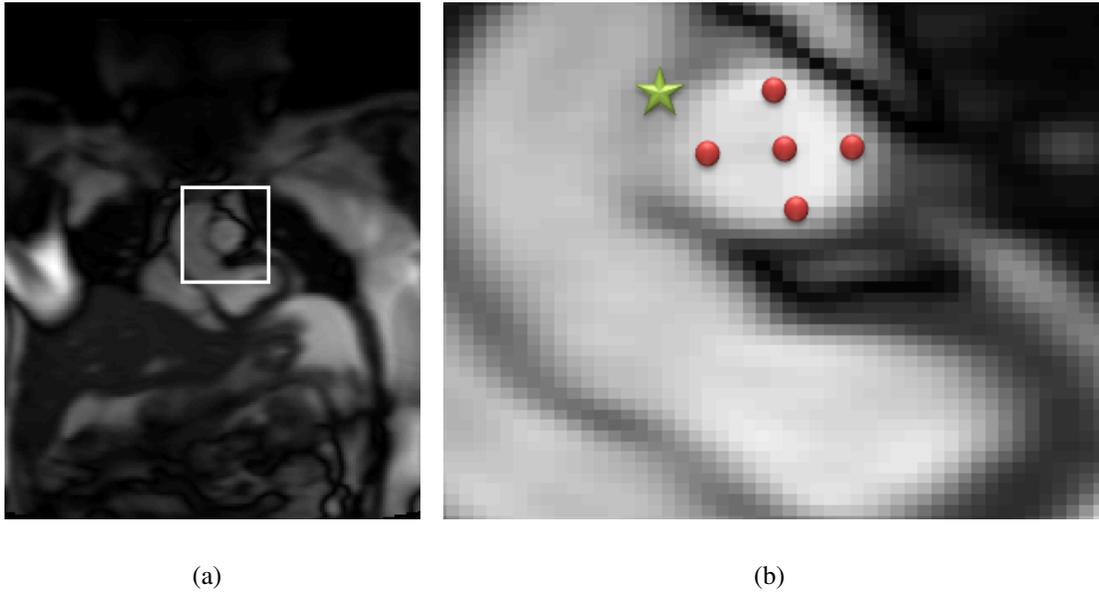


Figure 2.4: Depiction of a) MRI image, and b) zoomed in region. Round and star markers on the image are the seed points for the segmentation.

However, when we select the boundary of the object as a seed point (star marker in Figure 2.4(b)) then we will have a different segmentation output (Figure 2.5(f)). The reason that we did not segment a complete region of two objects is that we set the maximum number of iterations to 350. The question why we have segmented two regions instead of one, we can answer as follows: since we have selected the peak region of the gradient as depicted in Figure 2.6(a), the Chan-Vese algorithm contracts the curve in 360° , and since the neighboring gradients are lower than the gradient in the selected location, the region of segmentation will grow to the left and right (assuming 1D) of the selected seed point.

We have covered the segmentation of one frame; however we need to segment all of the frames in the MRI image sequence in order to be able to reconstruct diastole/systole images. Selecting seed point for each image in the sequence would be time consuming [54].

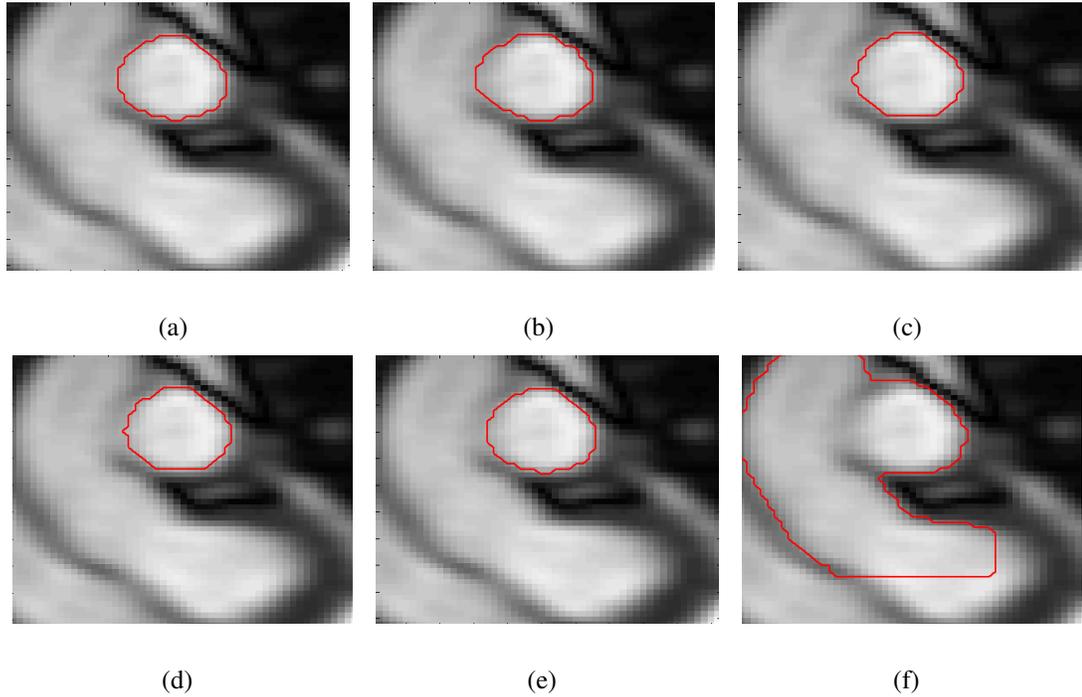


Figure 2.5: Depiction of segmentation with a seed point a-e) for the round marks in Fig 2.4(b), and f) is the segmentation with a seed point for the star marker.

One of the solutions might be to use the initial seed point for all images in the sequence. This will work only in cases where the subject does not move. If the subjects move, the segmentation algorithm will segment different regions. A better solution would be to use the contour of the previous image as an initial contour to the current image. This has two advantages over using one seed point for all images. First of all it will not require human interaction for segmentation and even if the subject moves during acquisition, due to the high rate of acquisition speed of the MRI, the displacement will be very low. We have showed that the different seed point location within the boundary of the region will give the same results (Figure 2.4 and Figure 2.5). However, if in the current frame the region of interest is shrunk compared to the previous frame, then the contour might grow during

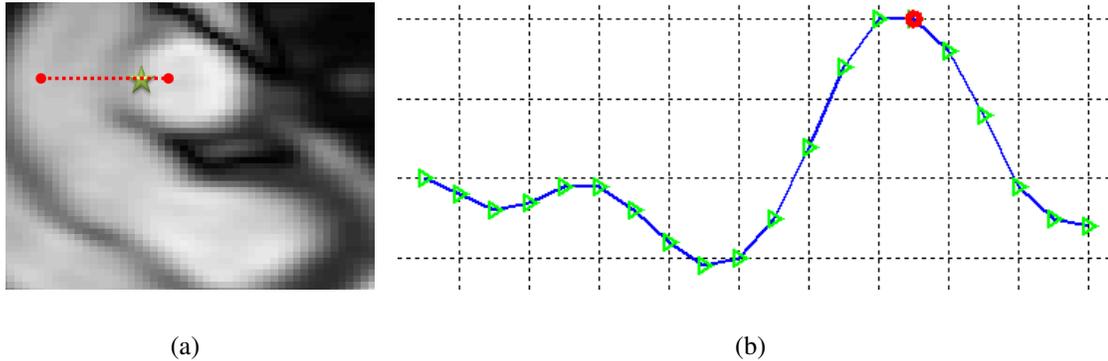


Figure 2.6: a) Region of interest, b) a 1D signal is extracted along the red line in (a), and its gradients are plotted. The red round marker in (b) represents the seed point selected in the (a) (star marker).

segmentation instead of shrinking, as shows in Figure 2.5(f), because the initial contour boundary might hit the peak gradient in the current frame, while it was within the boundary of the region in the previous frame. One of the solutions to this can be to find the center point $q = \frac{1}{b-a} \int_a^b f_{x_i} df_{x_i}$ of the contour and use this as a seed point to the current frame for segmentation. The drawback of this approach is that in each frame we will segment as if we are performing segmentation for the first time, and we don't have any knowledge of the region, thus we will run full iterations for each frame in the sequence which will have significant computation times. A better solution would be to use the information from the previous frames, which should require fewer iterations.

We know that the global difference between consecutive frames is very small (small body movement) due to the high speed frame acquisition and low variation of the body. The highest variation of the body is the aorta in the heart region due to the heart beat, but still the displacement of the center of the aorta will be very small, and the difference between regions of the two consecutive frame of the aorta regions will be also very small (Figure 2.7).

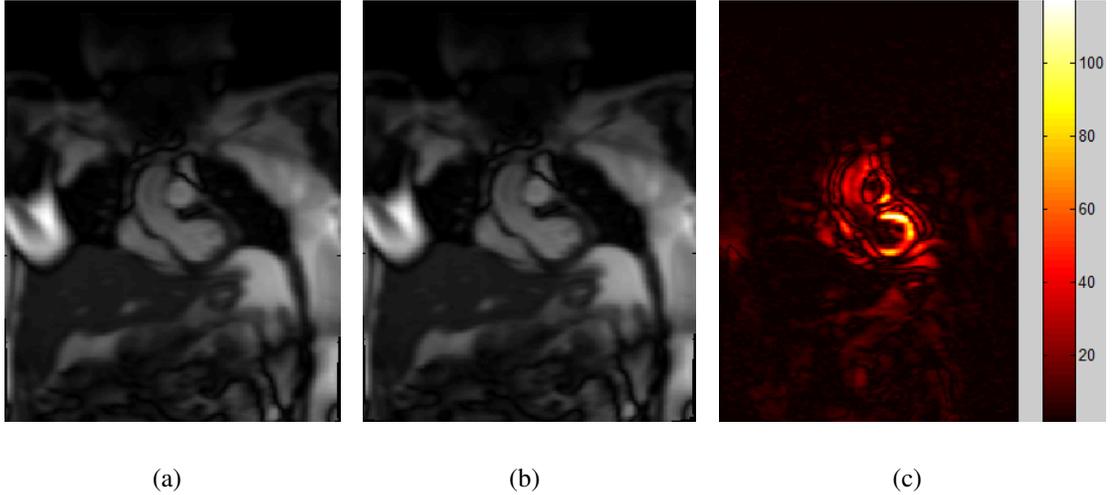


Figure 2.7: Depiction of two consecutive frames in the MRI image sequence and their absolute difference. Depiction a) 120th frame, b) 121th frame and c) the absolute difference between two frames. The color bar on the right of the (c) is color representation of the absolute difference values.

With this knowledge about the images, we can find a solution that would satisfy our objectives. The simplest solution is to shrink the contour of the previous frame and use it as an initial contour for the current frame. Another solution would be to estimate the radius of the contours in each previous frames and use a circle contour with mean radius of the previous contours in the current frame. Both solutions will have low computation cost.

2.2.2 Signal Extraction and Denoising

After finding the contours $f_{X_i} = C$ for each MR image, we need to extract signal s_i ($s \in \mathcal{R}^{1 \times n}$) from each frame contours C_i . This can be done by minimizing $\|s_i - \xi(f_{X_i})\|_2^2$ with respect to s_i , where $\xi(\cdot)$ is the function that returns the signal value from the given

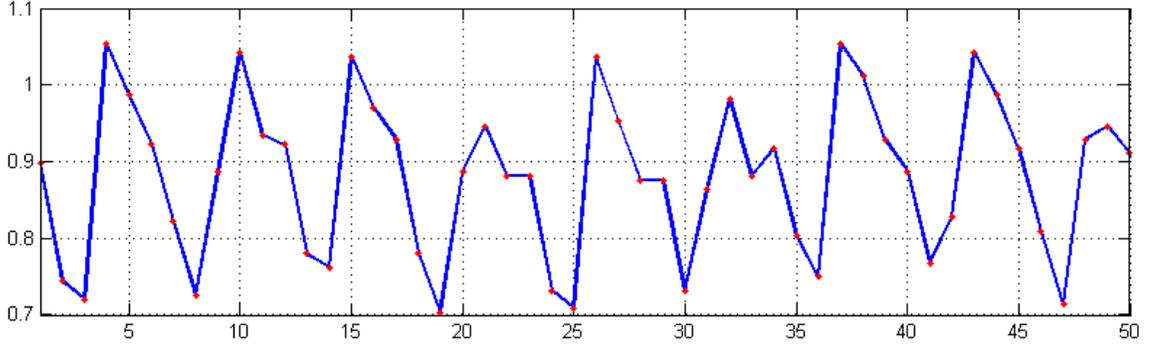


Figure 2.8: Depiction of heart beat signal exacted from MRI images through segmentation. After integrating over the contour, we will have the area inside of the segmented region, with is the total number of pixels. The area for each frame is normalized by dividing the area of the region in the current frame with the area of the initial frame.

f_{x_i} contour. We propose to integrate over the contour f_{x_i} , where the result will be the area inside the contour. It can be written as:

$$s_i = \frac{1}{b-a} \int_a^b f_{x_i} df_{x_i} , \quad (2.13)$$

where f_{x_i} is the contour of the segmented region of i^{th} MR image in the sequence, (a, b) are the start and end points of the contour, and s_i is the signal value at time i . We have normalized the signal by dividing the signal with the inital signal value, which is the ratio between the area of the segmented region of the current frame and the initial frame ($s_i = s_i/s_0$).

After integrating over the contours f_{x_i} , we will have a 1D signal which is the heart beat signal of the subject. The signal extracted from MR images through segmentation might be noisy (Figure 2.8).

To remove the noise from the heart beat signal we need to filter the signal. Since we

have a priori knowledge of the signal ($P(s)$), we can easily denoise it. There are many ways to denoise the signal, such as band pass filtering, wavelet filtering [34], etc. Since we don't know the cut off frequencies of the signal to filter out the noise part, we used a wavelet transform to denoise the signal.

Figure 2.9 depicts the test of three levels of decomposition of a heart beat signal using the Haar wavelet function [34]. Notice that when we use Haar wavelet functions (Figure 2.11(a)) to decompose the signal into three levels of decomposition, the details d_1 , d_2 , d_3 and the analysis a_3 part of the signal are rectangular.

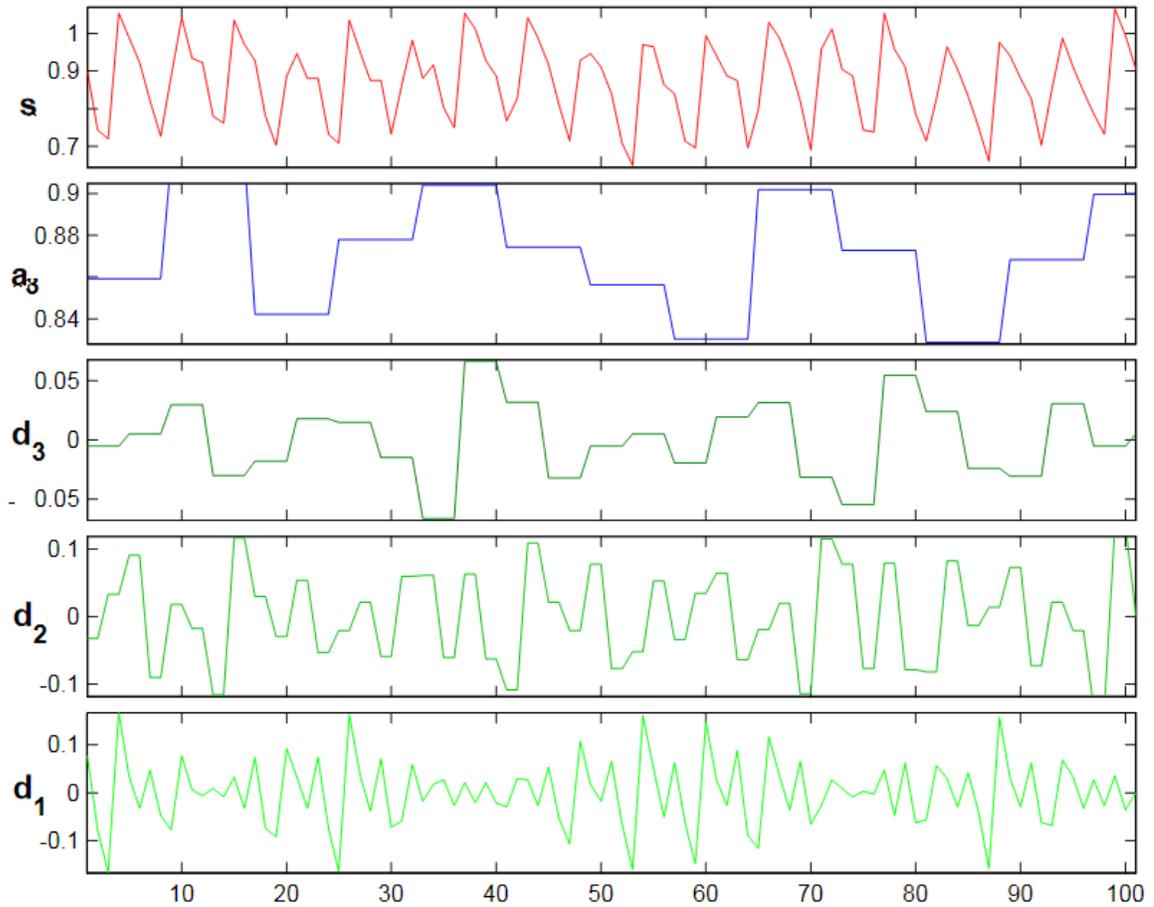


Figure 2.9: Depiction of three levels of decomposition using a Haar wavelet function. The top figure is the original signal (s), the second figure is the analysis (a_3) part of the signal after three levels of decomposition, and d_1 , d_2 and d_3 are the detail parts of the signal at each level.

This is due to the Haar wavelet function operating on the consecutive neighboring samples [49], where the detail and analysis parts of the signal will be rectangular with sharp changes (locality). In other words, Haar wavelet functions do not take into account the global shape of the signal. Notice that the lowest variation in the signal will be depicted in the analysis part a_3 , and the highest variation in the signal will be depicted in the detail

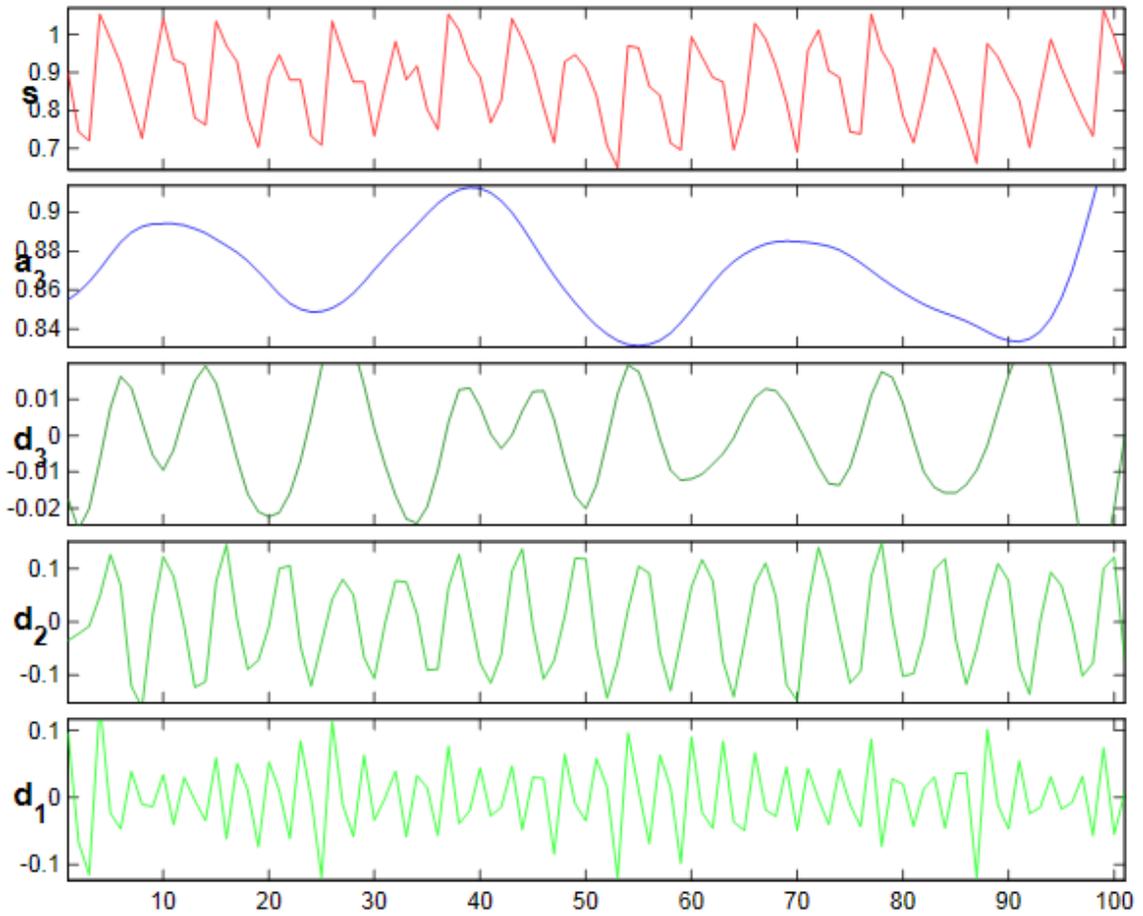


Figure 2.10: Depiction of discrete wavelet transform (DWT) using Daubechie (8) as a wavelet function.

part d_1 . By looking at the original signal s in Figure 2.9, it can be noticed that the original signal (heart beat) values varies between 0.7-1, and the detail part at level two (d_2) is the most similar to the original signal. Thus, we need to perform two levels of decomposition and remove the high variation (d_2) and take the inverse Haar wavelet transform, where the output will be the denoised signal. Figure 2.10 depicts the three levels of decomposition of a heart beat signal using the Daubechie (8) wavelet function [33]. Notice that the analysis and detail parts (except the high variations d_1) of the Daubechie decomposition are smooth

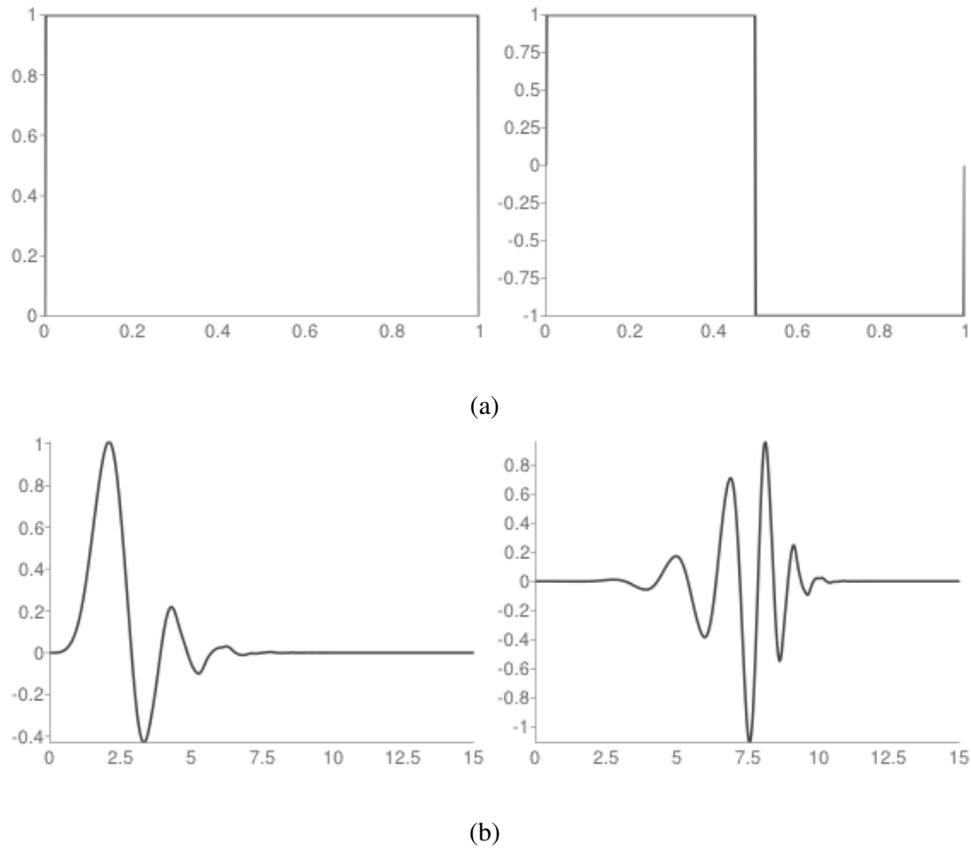
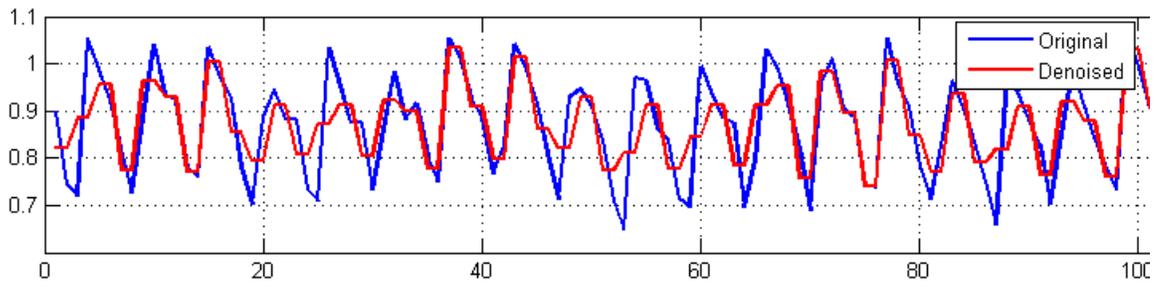
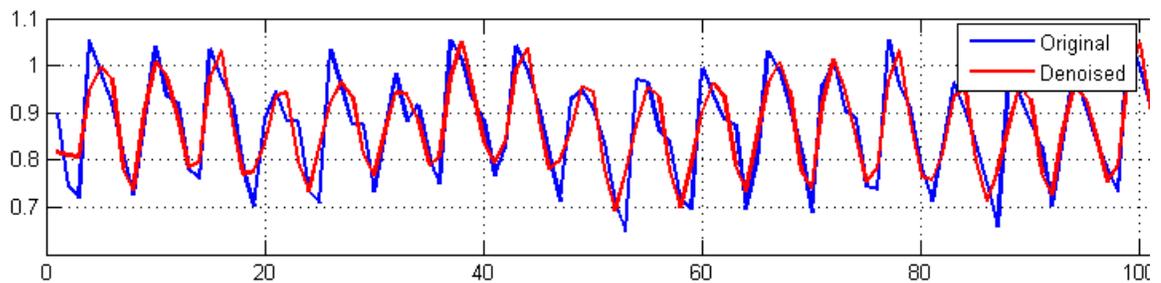


Figure 2.11: Depiction of scaling and wavelet functions for a) Haar and b) Daubechie (8).

compared to the analysis and detail parts of the Haar wavelet decomposition. This is because the scaling and wavelet functions (Figure 2.11(b)) of the Daubechie (8) operate on a higher range of the neighboring samples, where it takes into account the shape of the signal. Performing the same steps as in the Haar wavelet transform to denoise the signal, we can remove the noise. Figure 2.12(a) depicts the original signal and the output of the denoising using Haar wavelets and two level of decomposition. Notice that the output is still noisy after denoising it. Thus, we used Daubechie as a wavelet with a window size of 8 (Figure 2.11(b)) and two levels of decomposition [13,15]. Figure 2.12(b) depicts the



(a)



(b)

Figure 2.12: Depiction of original signal and output of denoising using a) Haar and b) Daubechie (8) wavelet transform. Notice that the denoised output of the signal using Haar wavelet is worse compared to the output of using Daubechie (8). This is because the Haar wavelet transform operates on local (small neighboring range) samples, whereas the Daubechie (8) operates on more global (higher neighboring range) samples which take into account the signal shape.

original signal (blue) and the denoised signal (red) after denoising using Daubechie (8) wavelet decomposition.

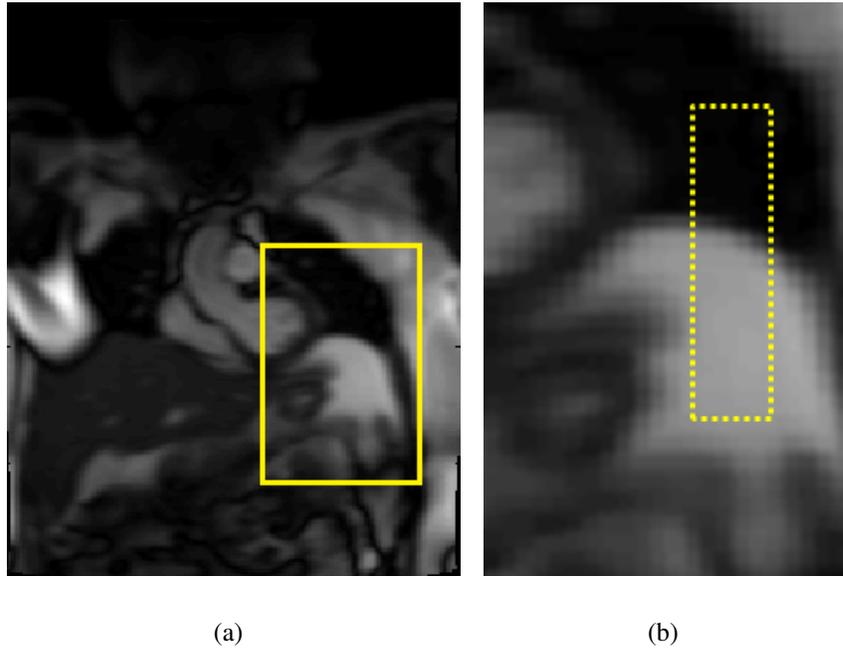
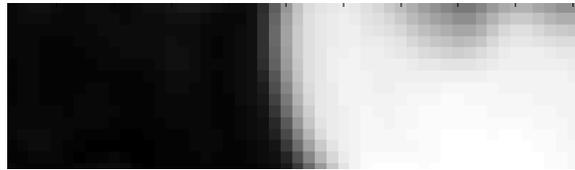


Figure 2.13: Depiction of a) 500th MRI image at frame, and b) zoomed in on the yellow box in (a).

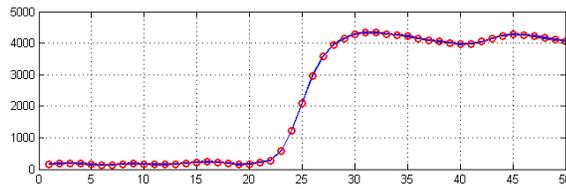
2.2.3 Reconstruction

The purpose of estimating the heart rate signal from the MR image sequence is that we want to reconstruct a single image y from multiple images. This is a task similar to obtaining super-resolution from multiple images [29, 19]. In multi-frame super-resolution, we assume that the object is not deformed, but has been displaced (affine transformation). Thus, by registering the object by bringing it to the same position in the spatial domain, we can apply algorithms that can combine multiple frames into one [29, 19]. In our project, we have a different situation, since the object (aorta) is displaced as well as deformed due to the blood pumping mechanism of the heart. This brings an additional challenge to the existing challenges in multi-frame reconstruction [59]. However, we know that even if the object (aorta) is deformed and displaced, it will come to the same position and the

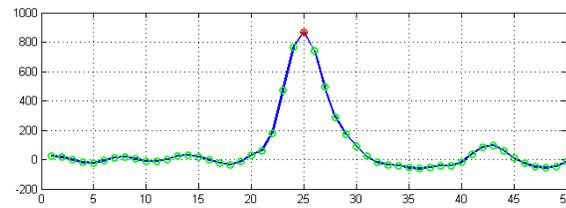
same shape ($f_{\mathcal{X}_i} \approx f_{\mathcal{X}_{i+k}}$) after some time, and we know that it will be repeated (periodic) [65, 79]. The heart beat signal is estimated by calculating the area of the segmented regions. Thus, from the heart beat signal we can only extract the information of the deformation state ($f_{\mathcal{X}_i}$) of the aorta at the time sequence i . In order to perform super-resolution or the fusion of multiple frames, we need the information of displacement of the aorta at time i .



(a)



(b)



(c)

Figure 2.14: Depiction of tracking. a) The image inside the yellow dotted rectangle in Figure 2.13. It is rotated for the display. b) The sum values along the column of the image in (a). c) First derivative of the sum values. The red dot is the highest peak in the summed values, which is the tracking point.

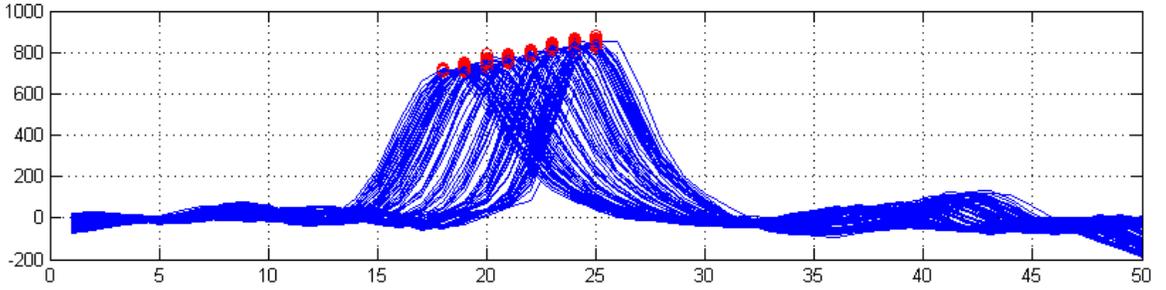


Figure 2.15: Depiction of tracking points for the frames numbered 100 through 199.

One possible solution is to perform a non-rigid transformation on the frame \mathcal{X}_i to estimate the parameters [3], and by using these transformation parameters register the frame \mathcal{X}_i to the reference frame \mathcal{X}_{ref} . Non-rigid or elastic transformation-based registrations are computationally expensive, thus we propose a different approach to avoid dealing with deformational displacements. We track the region inside the body close to the aorta, and by tracking the point we will have the displacement (x, y) of the aorta.

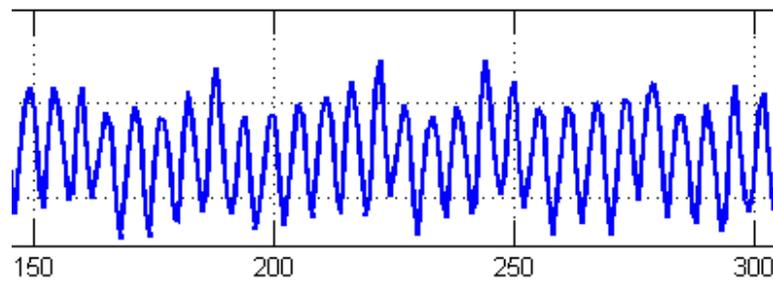
In Figure 2.13, we select the upper boundary of the torso region (yellow dotted rectangle) to track the gradient displacement. After selecting the region to track the point, we calculate the location of the gradient change, which will be our tracking point. First we extract the region (Figure 2.14(a)), let us denote it by \mathcal{Z}_i ($\mathcal{Z}_i \in \mathcal{X}_i$), and let $q \in \mathcal{R}^{1 \times n}$ be the sum of the values along the vertical direction, calculated as:

$$q(x) = \sum_{y=0..n} Z(x, y) \quad \forall x . \quad (2.14)$$

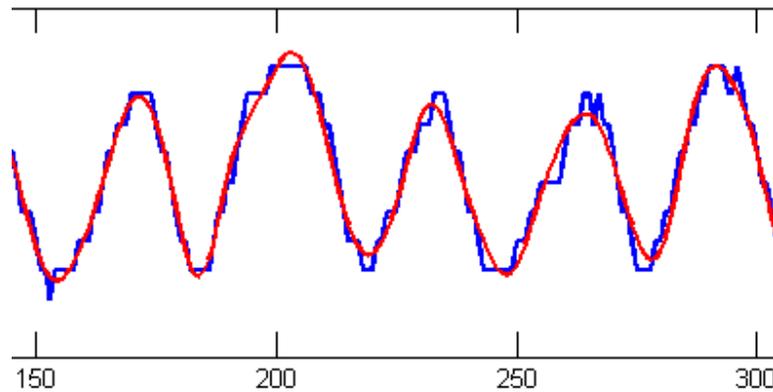
Figure 2.14(b) depicts the sum of the image along the column. The tracking point can be calculated by finding the maximum value of $\partial q / \partial x$ which is depicted in Figure 2.14(c).

Figure 2.15 depicts the peak values of the gradient changes and their positions. Notice

that using the proposed method for tracking, we will have only one peak point, which is the highest gradient change in the selected region. In other words, it is the boundary of the torso region. Tracking this region will give us the breath rate of the subject (low variation signal). Figure 2.16 depicts the denoised heart beat signal and the breath rate signal. In Figure 2.16(b), the blue line is the original signal (breath rate), and the red line is the denoised signal using Daubechie (8) wavelet with one level of decomposition.



(a)



(b)

Figure 2.16: Depiction of extracted a) heart beat and b) breath rate signals. The blue line and the red line in (b) are the original and denoised breath rate signals, respectively.

The extracted heart beat signal s_i at frame i gives us the information of the shape of the

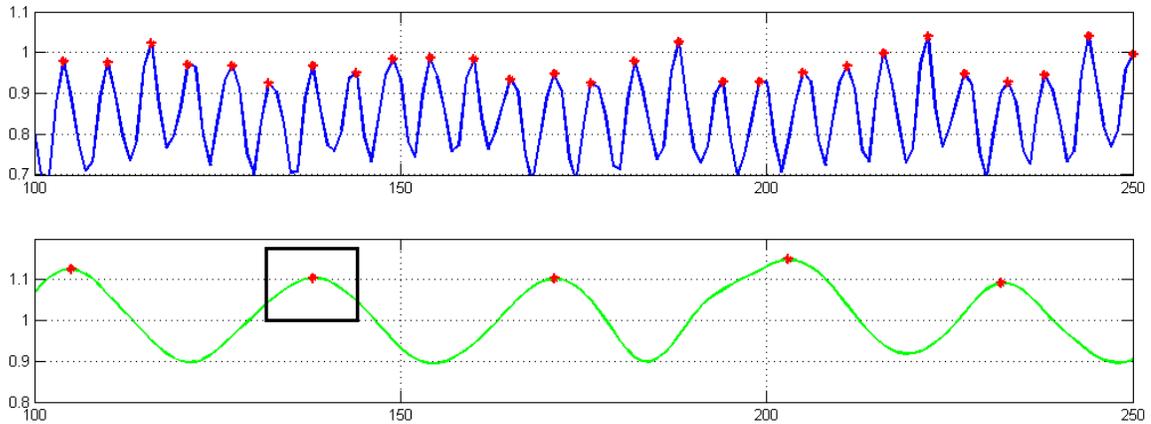


Figure 2.17: Depiction of heart beat and breath rate signal extracted from MR images (\mathcal{X}_i $i = 100 \dots 250$) through segmentation. Top figure is the heart beat signal, and bottom figure is the breath rate signal. Red stars (*) are the local maxima that the algorithm found.

aorta, and the breath rate signal at frame i gives us the information of the spatial position of the aorta. After denoising the heart beat and breath rate signals, we find the peaks in the heart beat signal which will be the diastole positions of the aorta. If we subtract the original signal from its maximum value ($s'_i = \max(s) - s_i \forall i$), the output signal will be the inverse of the original signal, and the peaks of the inverse signal will be the systole positions of the aorta.

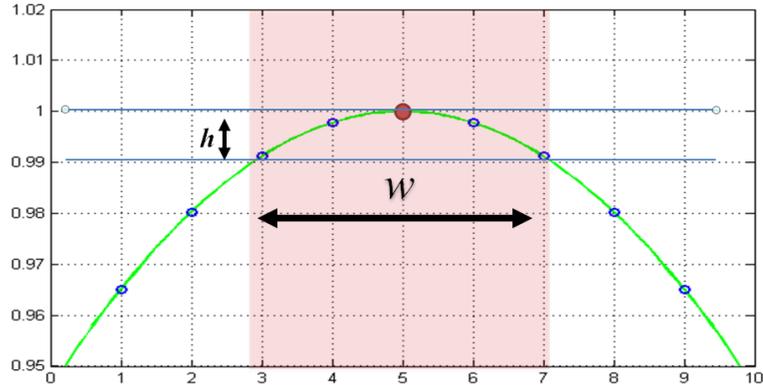


Figure 2.18: Depiction of breath rate signal (zoomed in on the black rectangle in Figure 2.17). The width of the selection box (red transparent box) is defined by w , which is $w/2$ to the left and to the right of the peak value (red dot).

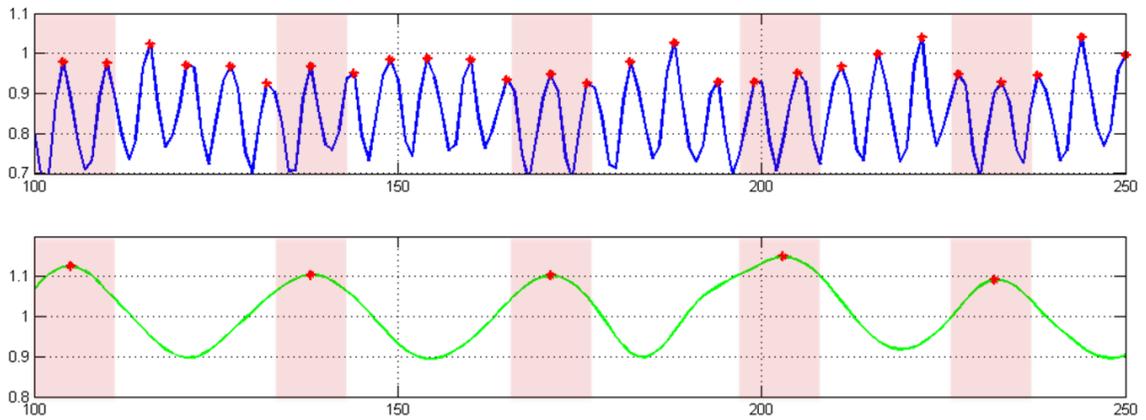


Figure 2.19: Depiction of breath rate signal (zoomed in on the black rectangle in Figure 2.17), heart beat and breath rate signal extracted from MR images (\mathcal{X}_i , $i = 100 \dots 250$) through segmentation. Red transparent rectangles are calculated from the breath rate signal, and are used to select heart beat signals, which will be used for reconstruction.

Thus, all we need to do is to select the location in one period from the heart beat signal and find a similar location in the periodic signal. We need to reconstruct the image for the

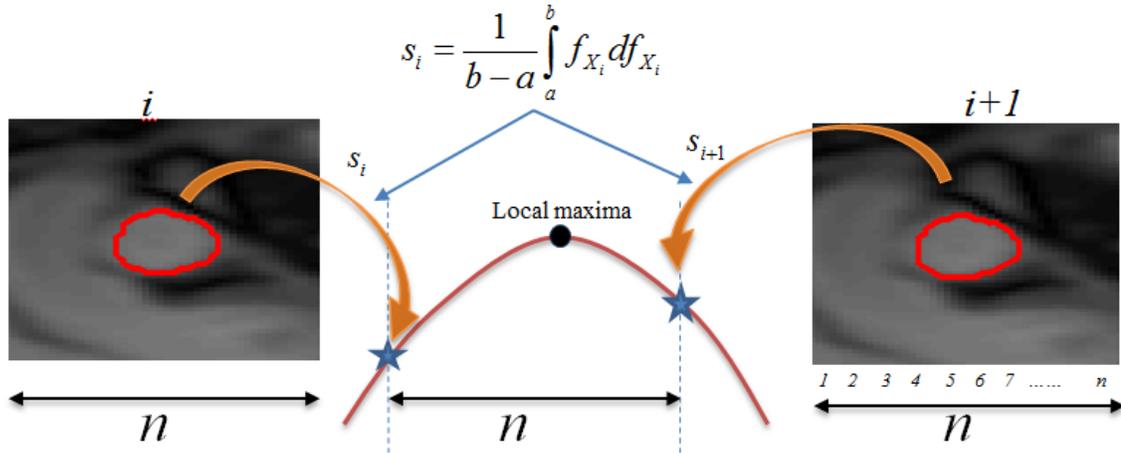


Figure 2.20: Depiction of local maxima and neighboring frames. The red curve is the zoomed in region of the selected region (rectangle with black borders) in Figure 2.17. The dark round point on the curve is the local maxima which is the highest point in the region, the i and $i + 1$ are the frame numbers, s_i and s_{i+1} are the signal values in the i^{th} and $(i + 1)^{th}$ position.

diastole position, which is the maximum integral of the contour from segmentation, and the systole position, which is the minimum integral of the contour ($\max(s_i) \quad i \in S, \quad S$ is the neighboring region) from the segmentation. In other words, we need to find local maxima in the signal first, and then invert the signal and find local maxima in the signal again, which are actually the local minima of the signal. Figure 2.17 depicts the peaks of the heart beat and breath rate signals estimated by our local maxima algorithm [64]. We cannot select the frames for the reconstruction randomly, which would result in a blurred image.

Figure 2.18 depict the zoomed in rectangle box in Figure 2.17. Notice that five points near the peak (red dot) region have small differences in their y values (small h). The vertical axis in the figure represents the displacement of the aorta in the spatial domain.

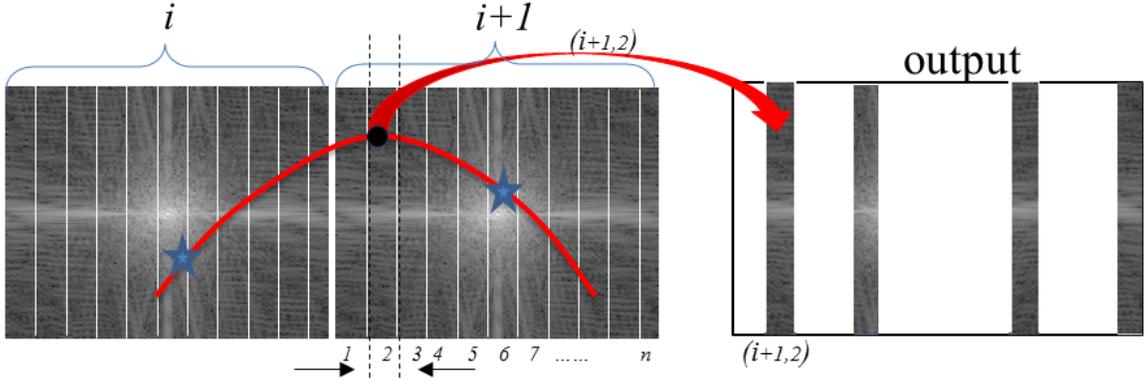


Figure 2.21: Depiction segment selection. 2D FFT is performed on both of the images, and each transformed image is segmented into n segments. Blue stars on the curve represent the center segment of the transformed image, and the dark round point on the curve represents the highest point in the region. The segment with the highest point on the curve is selected.

Thus, we select the width w of the windowing function where we include similar spatial displacements of the aorta. The idea of having a window function is to select the frames in which the aorta is in the same position in the spatial domain. After having the window function with its width w defined, we select the frames from the heart beat signal. We want to select the peak of the signals that fall into the window that we have calculated from the breath rate signal. Figure 2.19 depicts the heart beat and breath rate signal that is depicted in Figure 2.17, and the transparent red rectangle boxes are the windowing functions, where the frames fall into the peak region on the heart beat signal that will be selected for reconstruction.

Notice that in Eq. 2.10, the process of reconstruction is as follows. First we resample the signal by using n times bicubic interpolation to preserve the signal structure. If we have N MR images in the sequence, then we end up with $s \in \mathcal{R}^{N \times 1}$, by resampling the signal s , it will be $s \in \mathcal{R}^{nN \times 1}$. We resample the signal to find the local maxima (Figure 2.17)

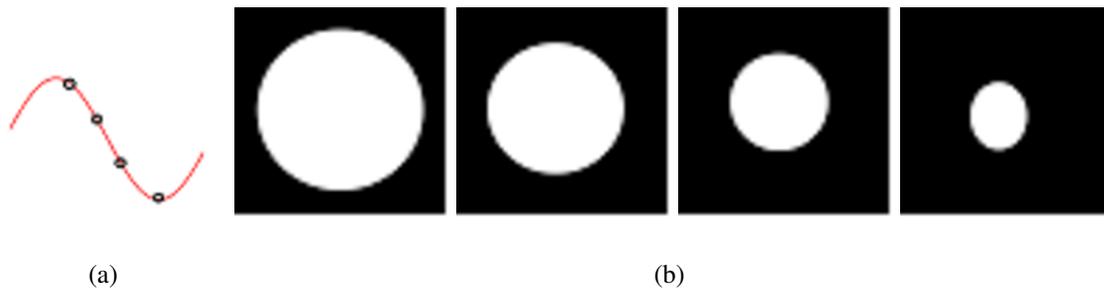


Figure 2.22: Depiction of our synthetic dataset. a) One cycle of a pure sinusoid signal, and marks are used to calculate the radius of a circle to generate image with round circle inside. b) Depiction of circle areas with different radii.

otherwise without resampling the local maximas (peaks) would always fall into one of the frames. After finding the peaks in the heart beat signal, we pick the left (i^{th}) and the right ($i + 1^{th}$) frames (Figure 2.21).

We take a 2D Fourier transform of both images, and segment them into n uniform segments along the column-wise. Since we have n samples between two signal points (between s_i and s_{i+1}) due to resampling, all we need to do is to find the segment that contains the peak of the signal (Figure 2.21), and place it into the output image based on its location in the original image. In Figure 2.21, we have selected the second segment of the $i + 1^{th}$ frame, thus we place it on the second segment location on the output image. This step is repeated for each peak (local maximum) in the signal. After filling in the output image, and by taking the inverse 2D Fourier transform we get the reconstructed image for diastole/systole.

2.2.4 Experimental Results

To test the performance of the proposed method we conducted several experiments to measure its accuracy, speed, and visual quality. For the first test, we tested the accuracy of the proposed algorithm. The only way to test the accuracy of the algorithm is to measure the difference between the ground truth image and the reconstructed image.

To ascertain the performance of our algorithm, we generated a synthetic dataset (Figure 2.22). First, we generated heart rate signal by using a pure sinusoid (range $-/+1$). For each sample of the signal z_i we calculate a radius $R_i(R_i = (2 + z_i) \times c)$ where c is a constant number (30), so the max radius of the circle is 90 pixels while the minimum radius of a circle is 30 pixels.

After generating the synthetic dataset, we run our proposed algorithm to reconstruct the systole and diastole images (images with maximum and minimum circles) on the synthetic dataset. After generating the reconstructed images, we calculate the MSE (mean-square-error) with the original images. For the original systole and diastole images we generate the circles with maximum radius (90 pixels) and minimum radius (30 pixels). We compared our algorithm against Farsiu *et al.*'s FRSR (Fast and Robust Super-Resolution) algorithm [29]. The MSE for our algorithm is 0.89, while the MSE for FRSR is 4.63.

For the second test, we tested the speed of the reconstruction. We mentioned earlier that the speed of the proposed method depends on the computation complexity of other algorithms, such as auto seed point selection for the current frame from previous frame, window size for the frame selection, resolution of the frame, total number of frames, etc.

We tested the effect of the resolution on the segmentation. Since the segmentation

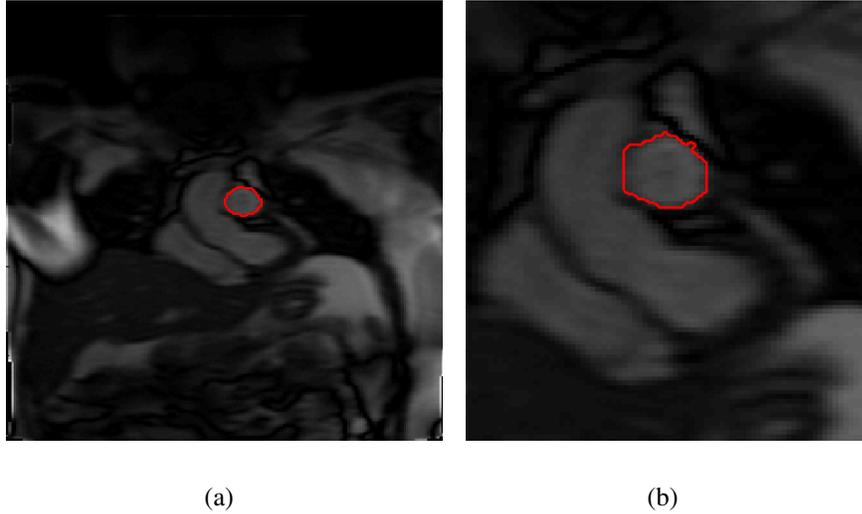


Figure 2.23: Depiction of segmentation using a) original image, b) region of interest that is manually selected between 75 and 175 pixels in vertical, 60 and 140 pixels in horizontal direction of the original image. The maximum number of iterations for segmentation in the test is set to 950.

algorithm needs to compute the two-dimensional gradient on the whole image, to find the exterior and interior forces of the contour, the higher the resolution, the higher the computation cost will be. Figure 2.26 depicts the two different MRI images, with different resolutions, where Figure 2.26(a) is the original image size (25x184 pixels), and for Figure 2.26(b) we selected the region of interest (100x80 pixels) which is between 75 to 175 pixels in the vertical direction, and 60 to 140 pixels in the horizontal direction. Figure 2.24 depicts the computation time for different frames. Notice that the average computation time for segmentation is 0.9 seconds when we use the whole image, and the computation time for segmentation is lower (average 0.6s) when we use only the region of interest. We wrote the segmentation code in MatLab 2011b, and performed the tests on Intel i7 960 3.20Ghz with 9GB or RAM with Windows 64bit as Operating System.

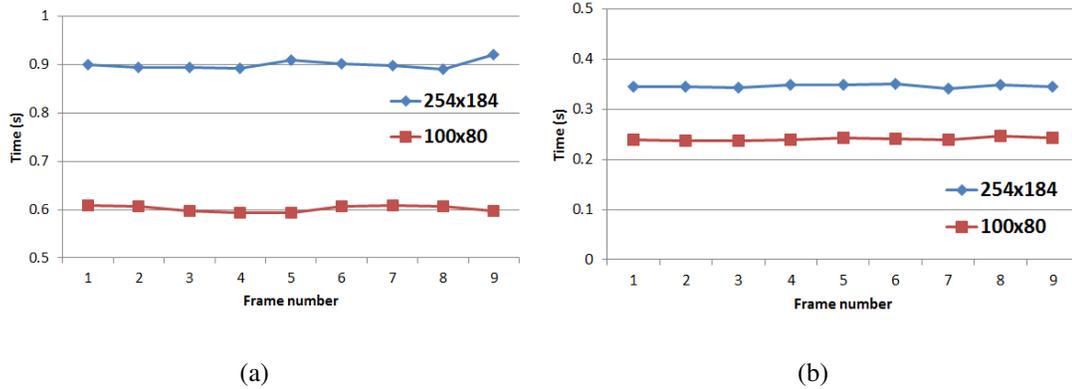


Figure 2.24: Depiction of computation time for different MRI images when we set the maximum number of iterations for segmentation to a) 950 and b) 350.

Assuming we have 1000 frames and we need to reconstruct the diastole/systole images from them, the computation time difference would be 300 seconds between using the original image and the region of interest image. The region of interest can be estimated from the previous frame, where we already extracted the contour of the object. We have mentioned earlier that the displacement of the aorta between two consecutive frames is very small (Figure 2.17), and by using this assumption we can estimate the region of interest.

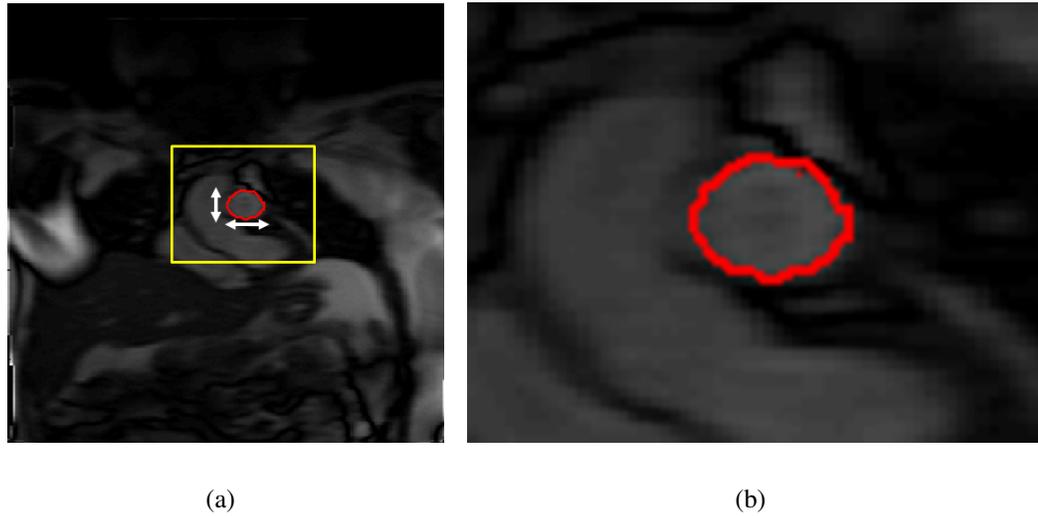


Figure 2.25: Depiction of region of interest estimation, where after segmentation performed on the initial frame we can compute the width and height of the contour. By using the estimated height and width information we can extract the region of interest for the next frame segmentation.

It will be computationally very fast to get the maximum and minimum values of the x and y positions of the contour, and by multiplying the width and height by 2 or 3, we will get the region of interest that segmentation can operate on for the next frame (Figure 2.25). Another factor that affects the computational performance of the segmentation is the maximum number of iterations. As is depicted in Figure 2.3, in the initial iterations the contour contracts more to merge the boundaries of the object, and in later iterations it fine tunes the contour. In Figure 2.24 we have set the maximum number of iterations to 950, and in Figure 2.26 we set the maximum number of iterations to 350. Notice that visually there is no difference between the two contours when we use 950 and 350 as a maximum number of iterations. However, one can see that there is a huge difference in the computation time of the segmentation (Figure 2.24).

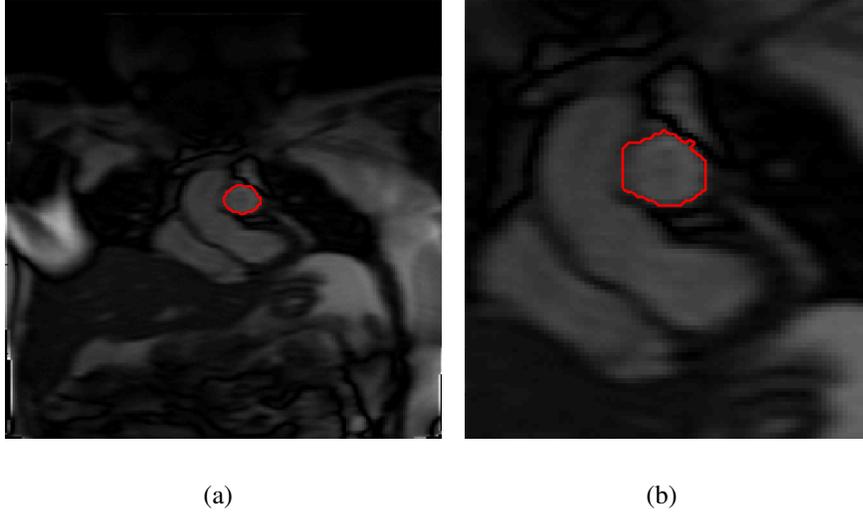


Figure 2.26: Depiction of segmentation using a) original image, b) region of interest that is manually selected between 75 and 175 pixels in vertical, 60 and 140 pixels in horizontal direction of original image. The maximum number iterations for segmentation in the test is set to 350. Notice that there is no difference between the output of Figure 2.26.

The average computation time for segmentation when we use the original image is 0.34 seconds, and when we use the segmentation on an image of size 100x80, the average computation time is 0.24 seconds. If we take the same example of 1000 frames, then the computation time difference between using the original image with 950 iteration and the region of interest image with 350 iterations would be 560 seconds, just for the segmentation. Signal extraction and denoising are computationally fast and don't depend on any parameters. As was mentioned earlier, we resample the heart beat signal (from $\mathcal{R}^{1 \times N}$ to $\mathcal{R}^{1 \times nN}$) to get finer approximation of the K-Spaces; this dramatically increases the computation time. Assuming that we have 1000 frames, the extracted heart beat signal will have 1000 samples, and by resizing it with the factor of n , which is the number of columns

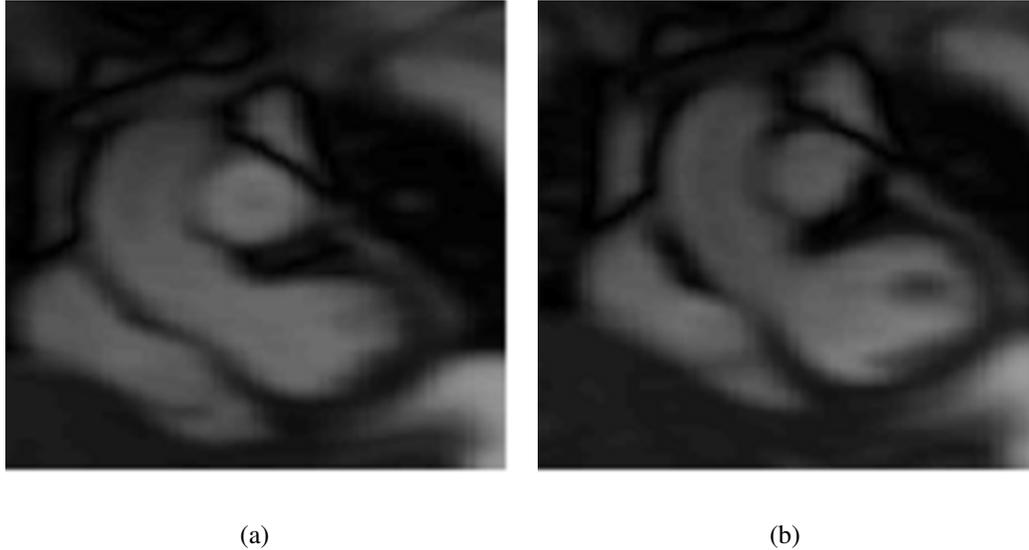


Figure 2.27: Output of reconstruction for an exact a) diastole and b) systole MRI images

(width) of the frame (in our frames it is 184), we find the peak values from 184,000 samples points. The average computation time to reconstruct diastole or systole image is 14.6 seconds, 13.2 seconds of which is to find local maximas (peaks) in the resampled signal. In addition, it was implemented in MatLab. It can be dramatically improved by using a GPU and C++.

We used five real-time MRI sequence clips from five different patients. Each sequence contains between 345 to 492 frames of dimension 256x184. The MRI sequences were acquired with the 1.5T Siemens MRI scanner at Houston Methodist Hospital. Figure 2.27 depicts the visual output of our proposed method. Notice the difference between the diastole and systole image outputs. We have also implemented a user-friendly graphical interface for this project (Figure 2.28).

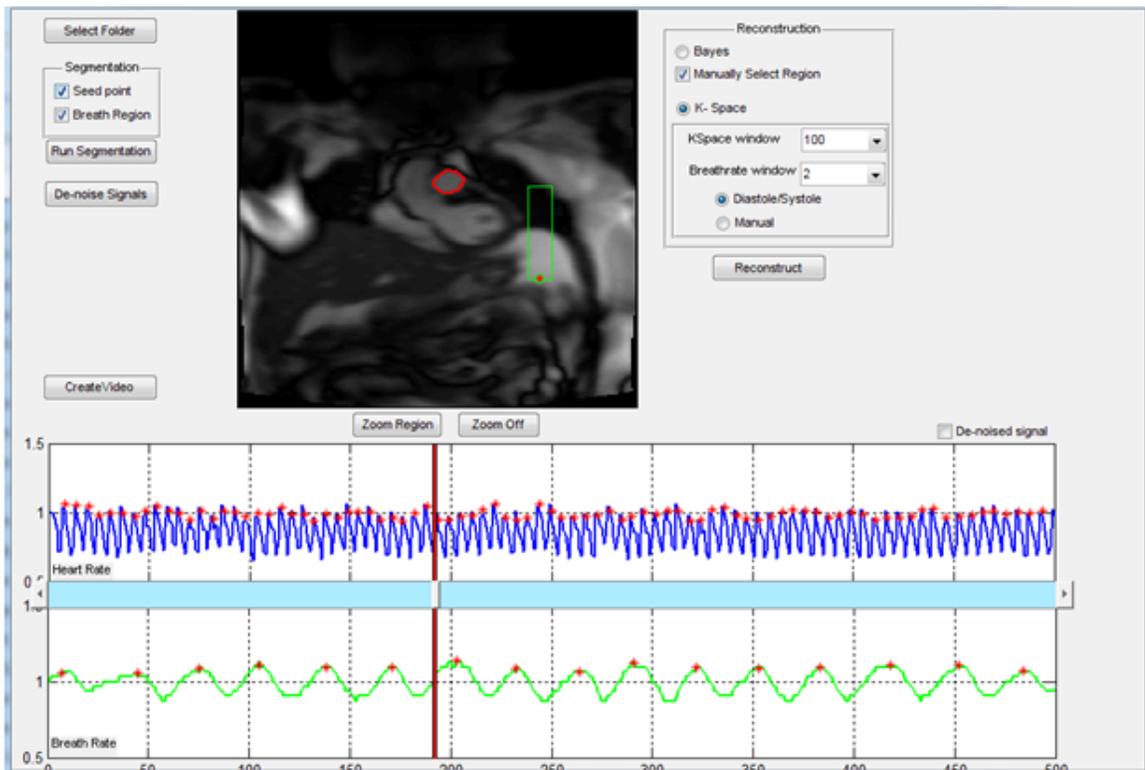


Figure 2.28: Depiction of the GUI that we have developed for this project.

Chapter 3

Objective II: Speed-up Super-resolution

3.1 Introduction

In the field of digital imaging, a dependable algorithm for creating super-resolution (SR) images is highly desirable [51, 6]. Such image reconstruction techniques can help overcome the definition limitations caused by low-cost imaging equipment which makes those methods highly useful in real-life applications such as biomedical and satellite imaging, as well as security surveillance and biometrics recognition [39, 74, 62, 63]. Note that hardware-based solutions require high-cost optics and sensors; this has been the main reason that made software-based techniques a hot research topic [2]. Current methods for creating SR images can be basically divided into two categories, namely: (i) (conventional) multiple-image SR, and (ii) machine learning (example)-based SR techniques [51, 20, 2]. The conventional methods require multiple low-resolution (LR) frames (images) of the same scene at subpixel misalignments and solve the inverse problem of recovering the

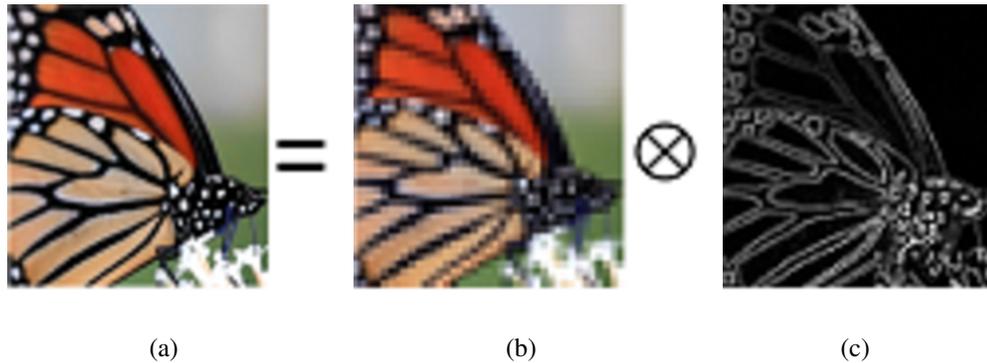


Figure 3.1: A sample SR image (a) obtained by combining a given (b) LR image, and (c) high-frequency information estimated from a natural SR training dataset.

original SR images by combining those LR frames. Such conventional techniques have been reported to increase resolution by a factor of less than 2 [22]. It is highly likely that this low performance rate is caused by the fact that the inverse problem of recreating the original image is ill-posed due to an insufficient number of input frames. Although there have been some efforts to address the issue of finding a robust solution to the inversion problem, the performance of those algorithms is far from giving stable results when only a small number of images is available and a significant increase in resolution is required [61, 2]. Example-based techniques learn the “correspondence” between LR and SR image patches from a database of LR and SR image pairs and apply this knowledge to a new LR image to recover its most likely SR version. As one of the most prominent methods, Freeman *et al.* [21] introduced a machine-learning based algorithm based on a Markov Random Field (MRF) to learn this correspondence, which was later extended by Sun *et al.* [60]. These techniques, however, require a large database of images to train the MRF.

In this chapter we introduce a robust algorithm for obtaining super-resolution images

by combining their low-resolution versions with high-frequency information extracted from training images, as illustrated in Figure 3.1. Bilgazyev *et al.* showed that the high-resolution image can be obtained by adding proper high-frequency information that is estimated from training database to input low-resolution image [7]. They proved that it improves the visual quality of the output image as well as the quantitatively outperforms existing super-resolution for face recognition [40]. However, their method highly depends on the size of the training database [67]. To reduce computation cost when the number of images in the training database is high, Bilgazyev *et al.* extended the work of Yang *et al.*, who built the compact dictionaries from patches to recover the high-resolution image [73]. This algorithm assumes, when we divide the whole training database into patches, that the large number of the patches will be similar to each other. This is $\|x_i - x_j\|_2^2 < \xi_0$ where x_i and x_j are the i^{th} and j^{th} patch in the training database, ξ_0 is a nonnegative value, and $\|\cdot\|_2^2$ is the l_2 -norm [1]. This is true when ξ_0 is high. For example, if we set ξ_0 very high, then this equation to measure the difference between patches will hold true for all patches in the training database. If we set ξ_0 to zero, then this will not hold true for the patches in the training database unless there is zero overlap between two images. In addition to removing the patches through sparse representation, Yang *et al.* performed preprocessing on the patches to reduce the computation cost of the sparse representation. They calculated the variance of gradients of each patch and removed those patches who had very small number of variance. We use this idea in our approach, which will be explained in the implementation section; however, we do not reduce the size of the training database by building compact dictionaries. Figure 3.2(a) is the original image in the training database; assuming that each image is 1024x1024 pixels, and we have 20 images, the

total number of extracted patches of a size of 16×16 pixels through the sliding window will be $\approx 21 \times 10^6$, which will take 5GB of storage space (Figure 3.2(b)). By removing unnecessary patches, which are the patches that don't contain any gradients (Figure 3.2(c)), we reduce the storage space to 550MB, and by building a compact dictionary through sparse representation we reduce the size of the training database to $\approx 0.5\text{MB}$ (256×2048) [68] (Figure 3.2(d)), where 2048 is the parameter that one defines during dictionary building. Figure 3.3 depicts the pipeline of the super-resolution algorithm that many algorithms use [32, 70, 76, 31, 26]. Notice that the idea behind building a compact dictionary is to reduce the size of the training database, because the computation time of the super-resolution reconstruction heavily depends on it.

The size of the training database can be dramatically reduced by this method, but the performance also will be reduced [68]. To overcome this issue, we propose a method where we don't build compact dictionaries; instead we use original images and search inside the images. This will reduce the storage space bandwidth; using parallelization of the code it can be implemented in a GPU to reduce the computation time of reconstruction.

To this end, we used (i) a nearest-neighbor-search algorithm to pick the closest images from the training dataset, and (ii) compressed-sensing to determine a weight parameter to combine the high-frequency information of the selected images. The main idea behind our algorithm is that it searches for similar images from the database and uses the extracted information from them rather than numerically predicting the output [15].

As shown in Figure 3.1, high-frequency data enables the algorithm to determine the sharp edges on the reconstructed image.

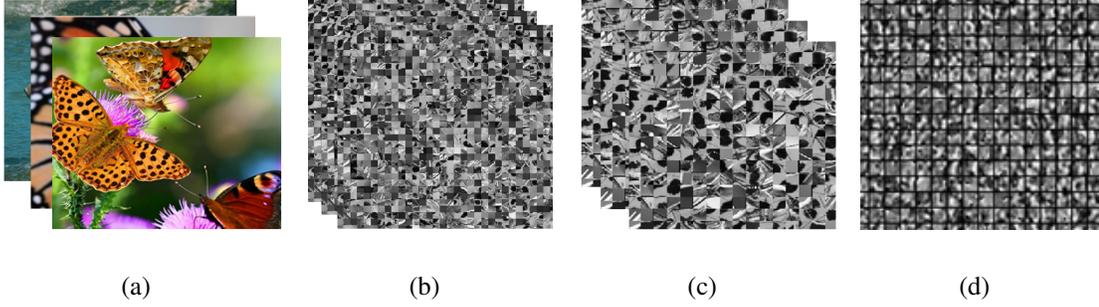


Figure 3.2: Depiction of building compact dictionary from high-resolution images by reducing the number of patches. a) Original high-resolution image in the training database (20MB), b) patches from sliding window on training images (5GB), c) number of patches reduced by removing the patches with no or small gradients (550MB), and e) compact dictionary (0.5MB).

3.2 Methodology

Let $X_i \in R^{m \times n}$ be the i^{th} image of the training dataset $X = \{X_i : i = 0 \dots N\}$, and $x_{i,j} \in R^{k \times l}$ be the j^{th} patch of an image $X_i = \{x_{i,j} : j = 0 \dots M\}$. The wavelet transform of an image patch x will return low- and high-frequency information:

$$W(x) = [\varphi(x), \psi(x)] , \quad (3.1)$$

where W is the forward wavelet transform, $\phi(x)$ is the low-frequency information, and $\psi(x)$ is the high-frequency information of an image patch x . Taking the inverse wavelet transform of high- and low-frequency information of original image (without any processing on them) will result in the original image:

$$x = W^{-1}([\varphi(x), \psi(x)]) , \quad (3.2)$$

where W^{-1} is the inverse wavelet transform. If we use a Haar wavelet transform with its coefficients being 0.5 instead of $\sqrt{2}$ (nonquadratic-mirror-filter), then the low-frequency

Table 3.1: Notations used in this chapter

Symbols	Description
X	Collection of training images
X_i	i^{th} training image ($\mathcal{R}^{m \times n}$)
$x_{i,j}$	j^{th} patch of i^{th} image ($\mathcal{R}^{k \times l}$)
ξ	threshold value
α_i	sparse representation of i^{th} patch
$\ \cdot\ _p$	l_p -norm
\mathcal{D}	Dictionary
$\mathcal{W}, \mathcal{W}^{-1}$	forward and inverse wavelet transforms
ψ, ϕ	high- and low-frequencies of image
$[\dots]$	Concatenation of vectors or matrices

information of an image x will actually be a low-resolution version of an image x , where four neighboring pixels are averaged; in other words, it is similar to down-sampling an image x by a factor of 2 with nearest-neighbor interpolation, and the high-frequency information $\psi(x)$ of an image x will be similar to the horizontal, vertical, and diagonal gradients of x .

Assume that, for a given LR image patch y_i which is the i^{th} patch of an image y , we can find a similar patch $x_j = \{j = 0 \dots NM\}$ from the natural image patches, then by combining y_i with the high-frequency information $\psi(x_j)$ of a high-resolution patch x_j ,

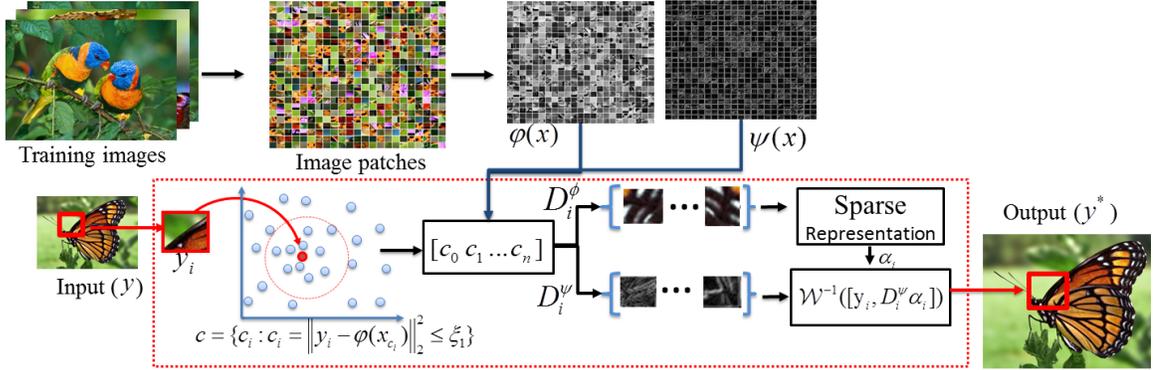


Figure 3.3: Depiction of the pipeline for the proposed SR algorithm.

and taking the inverse wavelet transform, we will get the SR y^* (see Figure 3.1):

$$y_i^* = W^{-1}([y_i, \psi(x_j)]) , \quad \|y_i - \phi(x_j)\|_2^2 \leq \xi_0 , \quad (3.3)$$

where ξ_0 is a small nonnegative value.

It is not guaranteed that we will always find an x_j such that $\|y_i - \phi(x_j)\|_2^2 \leq \xi_0$, thus, we introduce an approach to estimate a few closest LR patches ($\phi(x_j)$) from the training dataset and then estimate a weight for each patch $\phi(x_j)$ which will be used to combine high-frequency information of the training patches $\psi(x_j)$.

To find closest matches to the LR input patch y_i , we use a nearest-neighbor search algorithm:

$$c = \{c_i : c_i = \|y_i - \phi(x_{c_i})\|_2^2 \leq \xi_1, \forall x_{c_i} \in X\} , \quad (3.4)$$

where c is a vector containing the indexes (c_i) of training patches of the closest matches to input patch y_i , and ξ_1 is the radius threshold of a nearest-neighbor search. After selecting the closest matches to y_i , we build two dictionaries from the selected patches x_j ; the first dictionary will be the joint of low-frequency information of training patches $\phi(x_j)$ where

it will be used to estimate a weight parameter, and the second dictionary will be the joint of high-frequency information of training patches $\psi(x_j)$:

$$D_i^\phi = \{\phi(x_j) : j \in c\}, \quad D_i^\psi = \{\psi(x_j) : j \in c\}. \quad (3.5)$$

We use a sparse representation algorithm [60] to estimate the weight parameter. The sparse-representation α_i of an input image patch y_i with respect to the dictionary D_i^ϕ , is used as a weight for the fusion of the high-frequency information of training patches (D_i^ψ):

$$\alpha_i = \arg \min_{\alpha_i} \left\| y_i - D_i^\phi \alpha_i \right\|_2 + \lambda \|\alpha_i\|_1. \quad (3.6)$$

The sparse representation algorithm (Eq. 3.6) tries to estimate y_i by fusing a few atoms (columns) of the dictionary D_i^ϕ , by assigning non-zero weights to these atoms. The result will be the sparse-representation α_i , which has only a few non-zero elements. In other words, the input image patch y_i can be represented by combining a few atoms of D_i^ϕ ($y_i \approx D_i^\phi \alpha_i$) with a weight parameter α_i ; similarly, the high-frequency information of training patches D_i^ψ can also be combined with the same weight parameter α_i , to estimate the unknown high-frequency information of an input image patch y_i : $y_i^* = W^{-1}([y_i, D_i^\psi \alpha_i])$, where y_i^* is the output (SR) image patch, and W^{-1} is the inverse wavelet transform.

Figure 3.3 depicts the pipeline for the proposed algorithm. For the training step, from high-resolution training images we extract patches, then we compute low-frequency (will become LR training image patches) and high-frequency information for each patch in the training dataset. For the reconstruction step, given an input LR image y , we extract a patch y_i , find nearest neighbors c within the given radius ξ_1 (this can be speeded-up using a GPU), then from the selected neighbors c , we construct low-frequency and high-frequency dictionaries D_i^ϕ and D_i^ψ , where the low-frequency dictionary is used to estimate

the sparse representation α_i of the input low-resolution patch y_i with respect to the selected neighbors, and the high-frequency dictionary D_i^ψ will be used to fuse its atoms (columns) with a weight parameter, where the sparse representation α_i will be used as a weight parameter. Finally, by taking the inverse wavelet transform (W^{-1}) of a given LR image patch y_i with fused high-frequency information, we will get the SR patch y^* . Iteratively repeating the reconstruction step (red-dotted block in Figure 3.3) for each patch in the LR image y , we will obtain the SR image y^* .

3.3 Implementation Details and Results

In this section we will explain the implementation details.

3.3.1 Space Reduction

We extract patches for each training image $X_i = \{x_{i,j} : j = 0 \dots M\}$ with $x_{i,j} \in R^{k \times l}$. The number M depends on the window function, which determines how we would like to select the patches. There are two ways to select the patches from the image; one by selecting distinct patches from an image, where two consecutive patches don't overlap, and the other by selecting overlapped patches (sliding windows), where two consecutive patches overlap. Since the l_2 -norm in a nearest-neighbor search is sensitive to the shift, we slide the window by one pixel in the horizontal or vertical direction, where the two consecutive patches will overlap each other by $(k-1) \times l$ or $k \times (l-1)$, where $x_{i,j} \in R^{k \times l}$. To store these patches we will require an enormous amount of storage space, $N \times (m-k) \times (n-l) \times k \times l$,

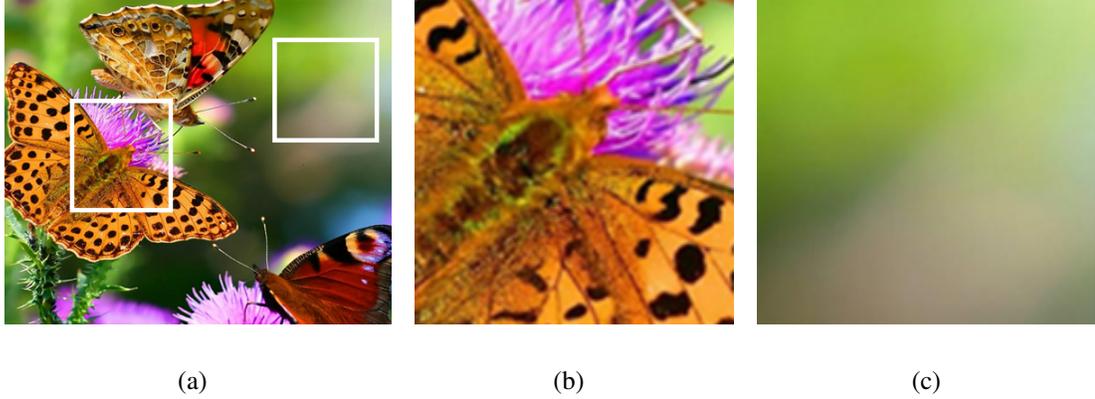


Figure 3.4: Depiction of a) original image, and b-c) two extracted patches from (a) (white rectangular boxes).

The variance of gradients of patch depicted (b) is 250.6, where the variance of gradients of patch depicted in (c) is 0.2.

where N is the number of training images and $X_i \in R^{m \times n}$. For example, if we have 1000 images natural images in the training dataset, and each has a resolution of 1000×1000 pixels, to store patches of size 40×40 , we will require 1.34TB of storage space, which would be inefficient and computationally expensive (Figure 3.2). To reduce the number of patches, we removed patches which contain few gradients:

$$\|\nabla x_{i,j}\|_2^2 \leq \xi_2, \quad (3.7)$$

where ∇ is the sum of gradients along the vertical and horizontal directions ($\nabla x_{i,j} = \frac{\partial x_{i,j}}{\partial x} + \frac{\partial x_{i,j}}{\partial y}$), and ξ_2 is the threshold value to filter out the patches with less gradient variation. Figure 3.4 depicts the original image, and two patches extracted from the original images (white rectangular boxes in Figure 3.4).

Similarly, we calculate the gradients on input LR patches (y_i), and if they are below the threshold ξ_2 , we upsample them using bicubic interpolation, where no SR reconstruction will be performed on that patch. To improve the computation speed, the nearest-neighbor

search can be carried out in the GPU, and since all given LR patches are calculated independently of each other, multi-threaded processing can be used for each SR patch reconstruction.

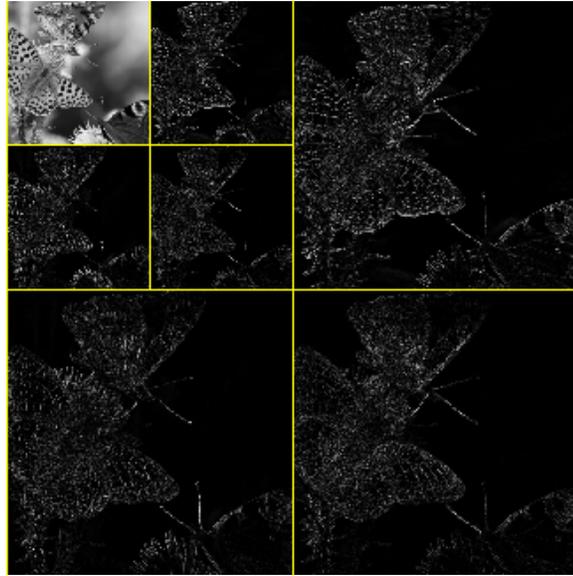


Figure 3.5: Depiction of two level Haar wavelet decomposition of image in Figure 3.4(a).

3.3.2 Wavelet Decomposition of Patches

In the wavelet transform, for the low-pass and high-pass filters we used $[0.5, 0.5]$ and $[-0.5, 0.5]$, and 2D filters for the wavelet transform are created from them. These filters are not quadratic-mirror-filters (nonorthogonal); thus, during the inverse wavelet transform we need to multiply the output by 4. The reason for choosing these values for the filters is that low-frequency information (analysis part) of the forward wavelet transform will be the same as down-sampling the signal by a factor of 2 with nearest neighbor interpolation,

which is used in the nearest-neighbor search. Figure 3.5 depicts the two-level decomposition using Haar wavelets of the image depicted in Figure 3.4(a). For one-level of wavelet decomposition there will be four outputs, one analysis output which is the low-resolution of original image (down-sampled by factor two), since we used a Haar wavelet, and three details part, which contain horizontal, vertical and diagonal gradients. In two-level decomposition there will be seven outputs, one analysis, which is the down-sampled version of original image (down-sampled by factor 4), and six detail parts, two horizontal, two vertical, and two diagonal images.

3.3.3 Color Images

During the experiments, all color images are converted to $YCbCr$, where only the luminance component (Y) is used (Figure 3.6). For the display, the blue- and red-difference chroma components (Cb and Cr) of an input LR image are up-sampled and combined with the SR image to obtain the color image y^* .

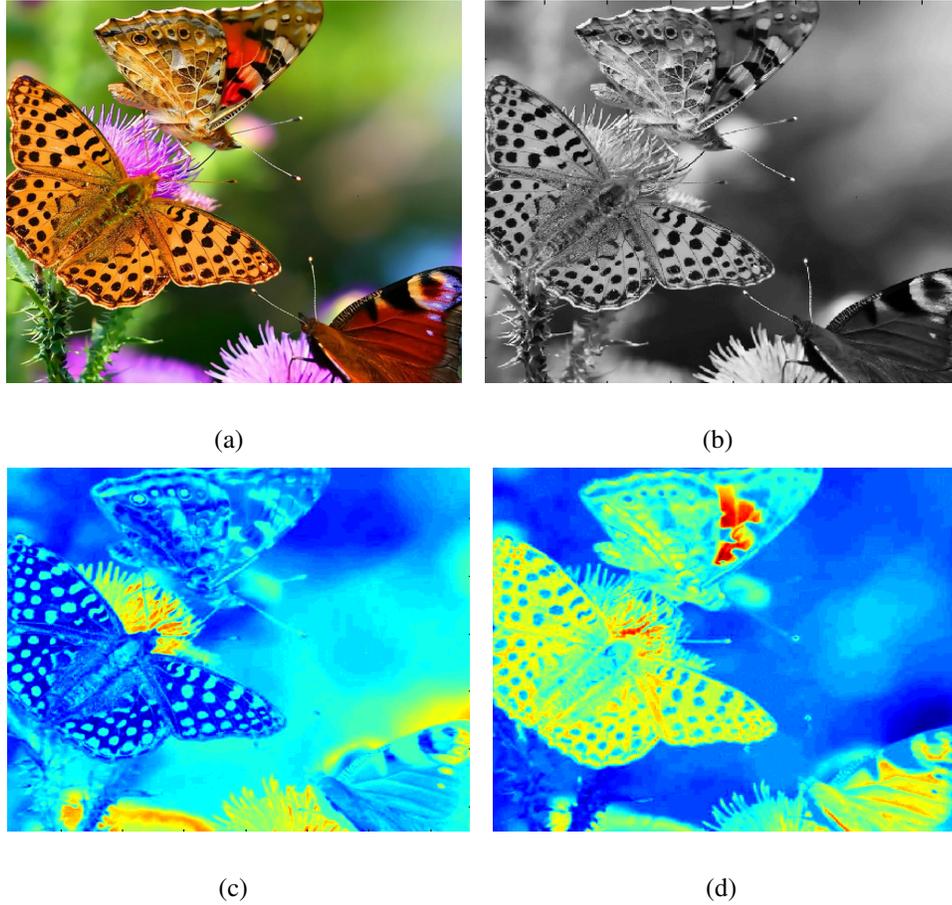


Figure 3.6: Depiction of a) original RGB image converted to YCbCr components. b) Chroma component, c) red-difference component, and d) blue-difference component.

3.3.4 Parallelization of the Algorithm

Note that we can reduce the storage space for the patches effectively to zero, by extracting the patches of the training images during reconstruction. This can be accomplished by changing the neighbor-search algorithm, and can be implemented in the GPU. During the neighbor-searching, each GPU thread will be assigned to extract low-frequency

$\phi(x_{l,j})$ and high-frequency information $\psi(x_{l,j})$ at an assigned position j of a training image X_l ; we compute the distance to the input LR image patch y_i , and if the distance is less than the threshold ξ_2 , then the GPU thread will return the high-frequency information $\psi(x_{l,j})$, where the returned high-frequency information will be used to construct D_i^ψ : $D_i^\psi = \{\psi(c_i) : c_i = \|y_i - \phi(x_{c_i})\|_2^2 \leq \xi_1, \forall x_{c_i} \in X\}$.

As a threshold (radius) value for the nearest-neighbor search algorithm we used 0.5 for natural images, and 0.3 for facial images. Both low-frequency information of training image and input image patches are normalized before calculating Euclidean distance. We selected these values experimentally: at these values we get highest SNR and lowest MSE. The Euclidean distance (in our nearest-neighbor search) is sensitive to noise, but in our approach, our main goal is to reduce the number of training patches which are close to the input patch. Thus, we take a higher threshold value for the nearest-neighbor search, where we select closest matches; then the sparse representation is performed on them. Note that sparse representation estimation (Eq. 3.6) tends to estimate an input patch from training patches, where noise is taken care of [42]. Reducing the storage space will slightly increase the SR reconstruction time, since the wavelet transform will be computed during the reconstruction.



(a)



(b)

Figure 3.7: Depiction of Berkeley Segmentation Dataset 500 images used for (a) training and (b) testing the SR algorithms.

3.3.5 Experiment Details

We performed experiments on a variety of images to test the performance of our approach (HSR) against the following SR algorithms: BCI [60], SSR [73], and MSR [21], using the Berkeley Segmentation Dataset 500 [43]. It contains natural images, where the natural images are divided into two groups; the first group of images (Figure 3.7(a)) are used to train SR algorithms (except BCI), and the second group of images (Figure 3.7(b)) are used to test the performance of the SR algorithms. To measure the performance of the algorithms, we use mean-square-error (MSE) and signal-to-noise ratio (SNR) as metrics.

These algorithms measure the difference between the ground truth and the reconstructed images.

3.3.6 Experimental Results

In Figure 3.8, we show the output of our proposed SR algorithm (HSR), and of BCI, SSR, and MSR. The red rectangle is zoomed-in and displayed in the corner of the figure. In this figure we focus on the effect of SR algorithms on low-level patterns (fur of the bear). Most of the SR algorithms tend to improve the sharpness of the edges along the border of the objects, which looks good to human eyes, and the low-level patterns are ignored. One can see that the output of BCI is smooth (Figure 3.8(a)), and from the zoomed-in region it can be noticed that the edges along the border of the object are smoothed, and similarly, the pattern inside the regions is also smooth. This is because BCI interpolates the neighboring pixel values in the lower-resolution to introduce a new pixel value in the higher-resolution. This is the same as taking the inverse wavelet transform of a given LR image with its high-frequency information being zero, thus the reconstructed image will not contain any sharp edges. The result of MSR has sharp edges; however, it contains block artifacts (Figure 3.8(c)). One can see that the edges around the border of an object are sharp, but the patterns inside the region are smoothed, and block artifact are introduced. On the other hand, the result of SSR doesn't contain sharp edges along the border of the object, but contain sharper patterns compared to BCI and MSR (Figure 3.8(d)). The result of the proposed SR algorithm has sharp edges, sharp patterns, as well as fewer artifacts compared to other methods (Figure 3.8(e)), and visually it looks more similar to the ground truth image (Figure 3.8(f)).

Figure 3.9 shows the performance of the SR algorithms on a different image with fewer patterns. One can see that the output of BCI is still smooth along the borders, and inside the region it is clearer. The output of MSR looks better for the images with fewer patterns, where it tends to reconstruct the edges along the borders. In the output of SSR, one can see that the edges on the borders are smooth, and inside the regions it has ringing artifacts. The SSR algorithm builds dictionaries from high-resolution and LR image patches by reducing the number of atoms (columns) of the dictionaries under a constraint that these dictionaries can represent the image patches in the training dataset with minimal difference. This is similar to compressing or dimension reduction, where we try to preserve the structure of the signal, not the details of the signal, and sometimes we get artifacts during the reconstruction.

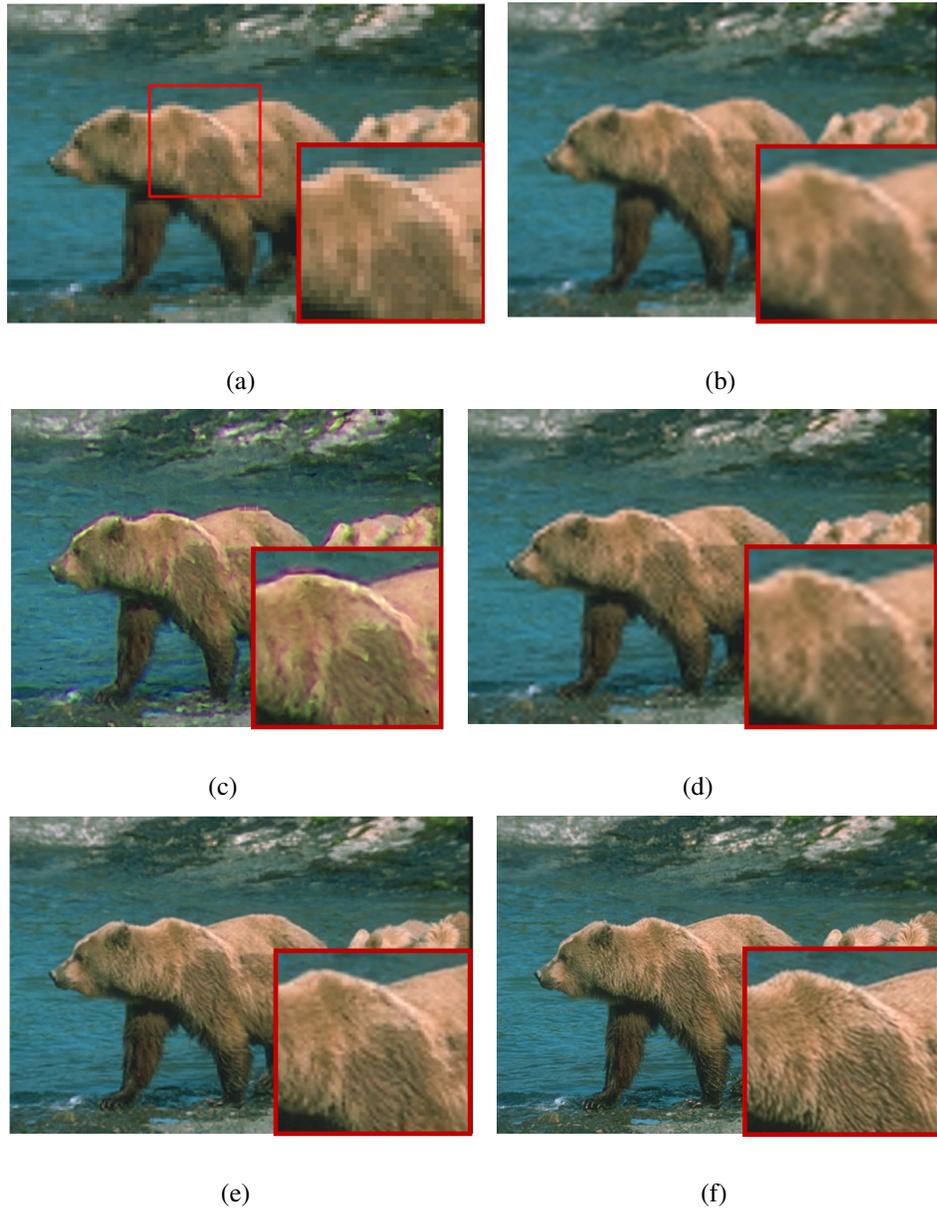


Figure 3.8: Depiction of LR, SR and original high-resolution images. (a) original LR image, (b) output of BCI, (c) output of SSR, (d) output of MSR, (e) output of our proposed algorithm, and (f) original high-resolution image. The solid rectangle boxes in red color represents the region that is magnified and displayed for better visualization. One can see that the output of the proposed algorithm has sharper patterns compared to other SR algorithms.

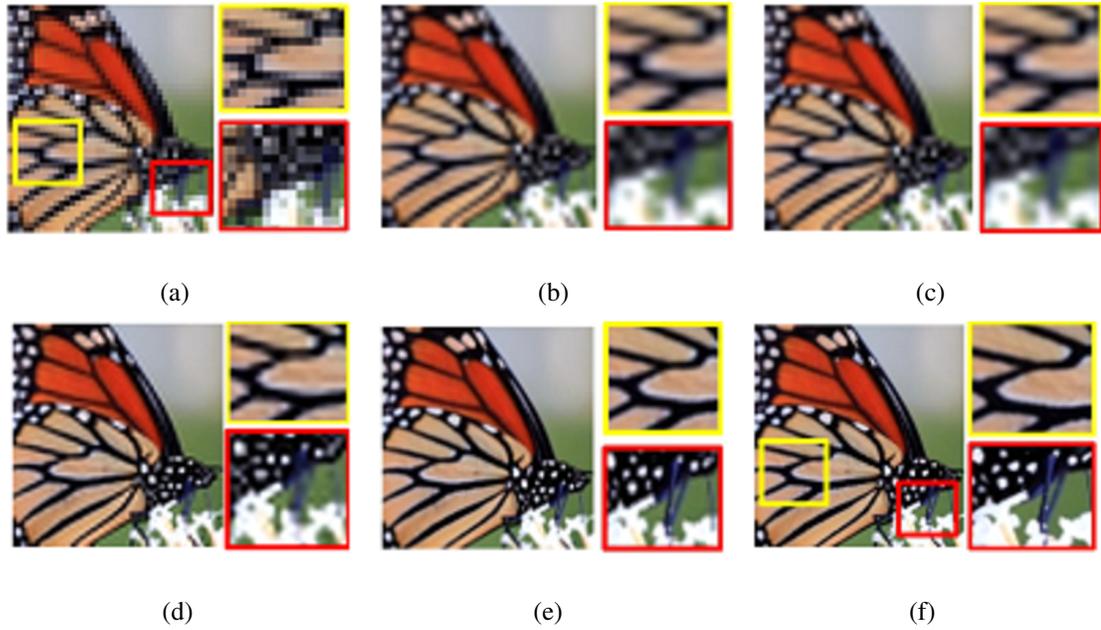


Figure 3.9: Depiction of LR, SR and original high-resolution images. (a)original LR image, (b) output of BCI, (c) output of SSR, (d) output of MSR, (e) output of our proposed algorithm, and (f) original high-resolution image. The solid rectangle boxes in yellow and red colors represent the regions that were magnified and displayed on the right side of each image for better visualization. One can see that the output of the proposed algorithm has better visual quality compared to other SR algorithms.

Table 3.2: Experimental results

Dist. Metric	SR Algorithms	BCI	SSR	MSR	HSR
	SNR (dB)		25.45	24.76	18.46
MSE		5.45	5.81	12.01	3.95

We also computed the average SNR and MSE to quantitatively measure the performance of the SR algorithms. Table 3.2 depicts the average SNR and MSE values for BCI, MSR, SSR, and HSR. Notice that our proposed algorithm has the highest signal-to-noise ratio and the lowest difference mean-square-error.

Chapter 4

Objective III: Face Recognition

4.1 Introduction

Face recognition has acquired increased importance in several real-life applications, namely video-based surveillance, human-machine interaction, biometric authentication, and others, due to increased demand for higher security [47]. With the latest improvements in image and video capturing device technology, the use of face recognition has become even more advantageous compared to other methods such as fingerprinting and iris matching [71]. In an environment where light, pose, facial expression, etc. can be controlled, automatic face recognition can outperform human-eye recognition especially when dealing with huge amounts of data [47]. On the other hand, in a typical surveillance scenario the environment is usually unconstrained where random lighting, posing, different facial expressions, make-up, weight gain or loss, aging, partial occlusions, and object motion are almost completely imponderable, which brings several challenges and open problems to

face recognition that need to be addressed [3, 5-7]. Lately, a wide range of different face recognition approaches have been introduced to address those challenges owing to the advances in computer vision techniques and increased interest in the field [1, 3, 7-12]. One of the most prominent algorithms for unsupervised face recognition is principal component analysis (PCA), a statistical method that determines a subspace representing most of the data variance [13, 14]. For supervised recognition, linear discriminant analysis (LDA) can find a subspace that maps the samples of the same class to a single spot and different classes as far apart from each other as possible [10, 14]. In this way the transformed low-dimensional vector can contain most discriminative features. However, PCA and LDA are very sensitive to noise in the images, which leads to a decrease in accuracy [10, 14]. Since the existing literature in the field is vast, we only focus on a few, directly related efforts.

Compressive sensing has emerged as a powerful approach for statistical signal modeling in computer vision and image processing and found successful use in face recognition [12, 15, 16]. The main idea behind compressive sensing, e.g., sparse representation, is to represent the input sample as a combination of a few atoms (columns) of an over-complete dictionary constructed from training samples. Recently, several methods have been proposed to estimate a sparse representation for the use of face recognition [17, 18]. Wright *et al.* introduced sparse representation-based classification to the face recognition area [19]. The classification is performed on residuals between input signal and the reconstruction of a sparse representation. As an over-complete dictionary, Wright *et al.* used the down-sampled images from the training dataset. Sparse representation-based classification demonstrated high recognition accuracy and also performed well, even when some facial features were occluded. This finding led others to investigate this approach further.

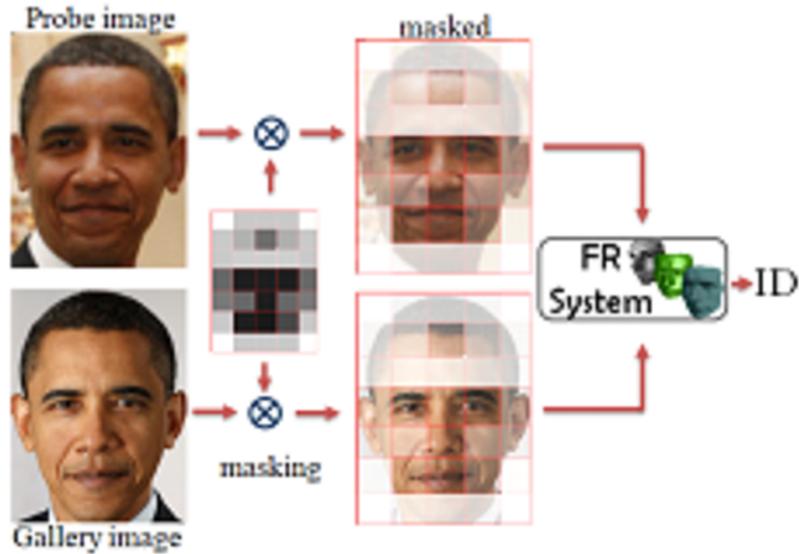


Figure 4.1: Depiction of proposed pipeline to use masked regions for face recognition.

Yang *et al.* used Gabor features in sparse representation-based classification to reduce computational complexity and improve recognition accuracy [20]. However, the main shortcoming of these methods is that many training images are required for good recognition performance, since the methods mentioned rely on the residuals, and thus use the images from the training dataset as an over-complete dictionary.

In this chapter, we introduce a novel methodology by modeling the sparse encoding with weighted patches to increase the robustness of face recognition even further. In the training phase, we define a mask (i.e., weight matrix) using a sparse representation selecting the facial regions, and in the recognition phase, we perform comparison on selected facial regions.

When the number of images in the training dataset is very large, the dictionary also

becomes very large, which makes the sparse process computationally expensive and real-time processing becomes very difficult. We present a novel approach to perform face recognition by using overlapped patches of a facial image as shown in Figure 4.1. To each patch we assign a weight that can maximize the recognition accuracy. Additionally, we propose a new classification approach, where instead of using the residual of the reconstruction from the sparse representation, (i) we use a sparse representation of an input image and a sparse representation of gallery images, and (ii) instead of constructing a dictionary by stacking the transformed images from training dataset, we build a compact dictionary [37]. This can result in faster recognition, where the reconstruction is independent of the database size.

The algorithm was evaluated both quantitatively and qualitatively using two comprehensive facial image databases, i.e., Face99 [53] and YFace [4], with the results clearly superior to common state-of-the-art methodologies in different scenarios.

4.2 Methodology

Table 4.1 summarizes the naming conventions used throughout the chapter. Let $x^i \in \mathcal{R}^{m \times n}$ denote a facial image of subject i , and let X be a collection of N facial images, $X = \{x^i : i = 1 \dots N\}$. Suppose that an image y can be represented as a weighted combination of training images in X , such that:

$$\left\| y - \sum_{i=1}^N \alpha_i x^i \right\|_2 \leq \varepsilon_0 \quad , \quad (4.1)$$

Table 4.1: Notations used in this chapter

Symbols	Description
X	Collection of gallery images
x	Any image in X ($\mathcal{R}^{m \times n}$)
x^i	i^{th} image in X
x_i	i^{th} patch of image x
α_i	i^{th} element of α (\mathcal{R}^1)
$\ \cdot\ _p$	l_p -norm
\mathcal{P}	Mask operator
\mathcal{D}	Compact dictionary

where $\alpha \in \mathcal{R}^{N \times 1}$ is a weighting vector, and ε_0 is a very small positive value. Ideally, we would want to represent an image y using as few training images from X as possible, thus an additional constraint is imposed:

$$\min_{\alpha} \|\alpha\|_0 \quad \text{s.t.} \quad \left\| y - \sum_{i=1}^N \alpha_i x^i \right\|_2 \leq \varepsilon_0 . \quad (4.2)$$

The optimization problem of Eq. 4.2 is extremely difficult (NP hard), due to the non-convexity of the l_0 norm [20]. Recent methods [19] suggest that as long as the desired coefficients of α are sufficiently sparse, they can be efficiently recovered by minimizing the l_1 norm; using Lagrange multipliers, Eq. 4.2 can be written as follows:

$$\alpha = \min_{\alpha} \left\| y - \sum \alpha_i x^i \right\|_2 + \lambda \|\alpha\|_1 , \quad (4.3)$$

where the parameter λ balances the sparsity of the solution. The main disadvantage of this method is that with the increase of the number of images (N) in the training database X ,

the computation time increases dramatically. A compact dictionary can be built from the training images [37] such that each image x in the training database X can be represented as:

$$x \cong \alpha \mathcal{D} \quad , \quad (4.4)$$

where α is a vector with few nonzero elements, and \mathcal{D} is a compact dictionary which is estimated from the training database X as:

$$\mathcal{D} = \min_{\alpha_{i=1 \dots N}, \mathcal{D}} \left\| \sum_{i=1}^N x^i - \alpha^i \mathcal{D} \right\|_2 + \lambda \left\| \sum_{i=1}^N \alpha^i \right\|_1 \quad . \quad (4.5)$$

Let P_i denote a mask operator that extracts the i^{th} patch ($x_i = P_i x$); then Eq. 4.5 can be written as:

$$\alpha = \min_{\alpha} \left\| \bigcup_i (x - \alpha \mathcal{D}) \right\|_2 + \lambda \left\| \bigcup_i \alpha \right\|_1 \quad . \quad (4.6)$$

Since the patches overlap we use union instead of sum, where the union of patches x_i is the original image x ($x = \bigcup_{i=1}^K \mathcal{P}_i x = \bigcup_{i=1}^K x_i$, where K is the number of patches). We need to assign a weight for each patch such that overall the representation α will be sparse. Thus, we introduce a weight parameter w_i for a patch operator \mathcal{P}_i in Eq. 4.6:

$$\alpha = \min_{\alpha, w} \left\| \bigcup w_i \mathcal{P}_i (x - \alpha \mathcal{D}) \right\|_2 + \lambda \left\| \bigcup w_i \mathcal{P}_i \alpha \right\|_1 \quad . \quad (4.7)$$

Instead of calculating the weights during the sparse representation estimation, we can estimate them during the training step where we can estimate both compact dictionary \mathcal{D} and weight parameter w iteratively:

$$\mathcal{D}, w, \mathcal{A} = \min_{\mathcal{A}, \mathcal{D}, w} \|w \mathcal{P}(X - \mathcal{A} \mathcal{D})\|_2 + \lambda \|w \mathcal{P} \mathcal{A}\|_1 \quad . \quad (4.8)$$

where $\mathcal{A} = \{\alpha^i : i = 1 \dots N\}$. Since we build a compact dictionary to speed up the sparse representation estimation, we cannot use the residual based recognition [75, 24]. Instead, after building the compact dictionary \mathcal{D} and estimating the weight w for each patch, we will estimate α^i for each image $x^i \in X$.

Let y be an input image, let α^* be its sparse representation, and let $\xi_i = \alpha^i - \alpha^*$ be the residual. We assume that $\xi_i \{i = 0 \dots N\}$ is independently and identically distributed with a variance σ_0 , and the maximum likelihood estimation $\mathcal{L}_{\sigma_0}(\xi_0 \dots \xi_{N-1})$ of a distribution is $\prod_{i=0}^N \mathcal{F}_{\sigma_0}(\xi_i)$ where $\mathcal{F}_{\sigma_0}(\cdot)$ is a probability distribution function with variance σ_0 . To maximize the likelihood \mathcal{L}_{σ_0} , we can take the negative logarithm of \mathcal{L}_{σ_0} , $(-ln)$, which will be similar to objective minimization problem, where the output will be the ID of an input image y :

$$\min_i \sum_{i=1}^N \|\alpha_i - \alpha^*\|_2 \quad \text{s.t.} \quad \|\alpha^*\|_1 \leq 1. \quad (4.9)$$

4.3 Implementation Details and Experimental Results

Algorithm 1 Training

Input: $X_i, i \in \mathcal{R}^N$

Output: $\mathcal{D}, w, \mathcal{A}$

Pick landmarks of reference image x^{ref}

for each image $X_i, i \in \mathcal{R}^N$

Register $X_i \rightarrow x^{ref}$

Extract patches for X_i ($X_i = \bigcup \mathcal{P}_i X_i$)

end for

Compute \mathcal{D}, w , and \mathcal{A} from $X_i, i \in \mathcal{R}^N$ (Eq. 4.7)

Algorithm 2 Recognition

Input: $y, \mathcal{D}, w, \mathcal{A}_i, i \in \mathcal{R}^N$

Output: i

Register $y \rightarrow x^{ref}$

Extract patches for y ($y = \bigcup \mathcal{P}_i y$)

Compute sparse representation α^ of y (Eq. 4.8)*

Estimate i for α^ from $\mathcal{A}_i, i \in \mathcal{R}^N$ (Eq. 4.9)*

In the training step, all the facial images are registered to a reference facial image using the detected landmarks by the Active Shape Model (see Section 4.3.2), and resized to 60x60 pixels. Given the input LR image y , landmarks are detected and registered to reference facial image, and resized. Algorithm 1 depicts the *pseudo code* for the training step, and Algorithm 2 depicts the *pseudo code* for the recognition step.

4.3.1 Experimental Databases

To evaluate the performance of the proposed methods and other existing methods we used two facial image datasets. We used two image datasets, namely Frontal Facial Dataset (Face99) collected by Perona *et al.* at California Institute of Technology [53] and Yale Face Database (YFace) collected by Belongie *et al* [4].



(a)



(b)

Figure 4.2: Depiction of probe and gallery images from the Face99 dataset. (a) Images used as a gallery, and (b) as a probe (testing).

Face99: Frontal face dataset collected by Markus Weber at California Institute of Technology consists of total 450 facial images of 27 subjects captured under different lighting, expressions, and backgrounds. Each image has 896 x 592 pixels and is stored in JPEG format. We used normal images (without illumination and expression) of each

subject for the gallery, and the rest of the images as a probe to test the face recognition algorithms.

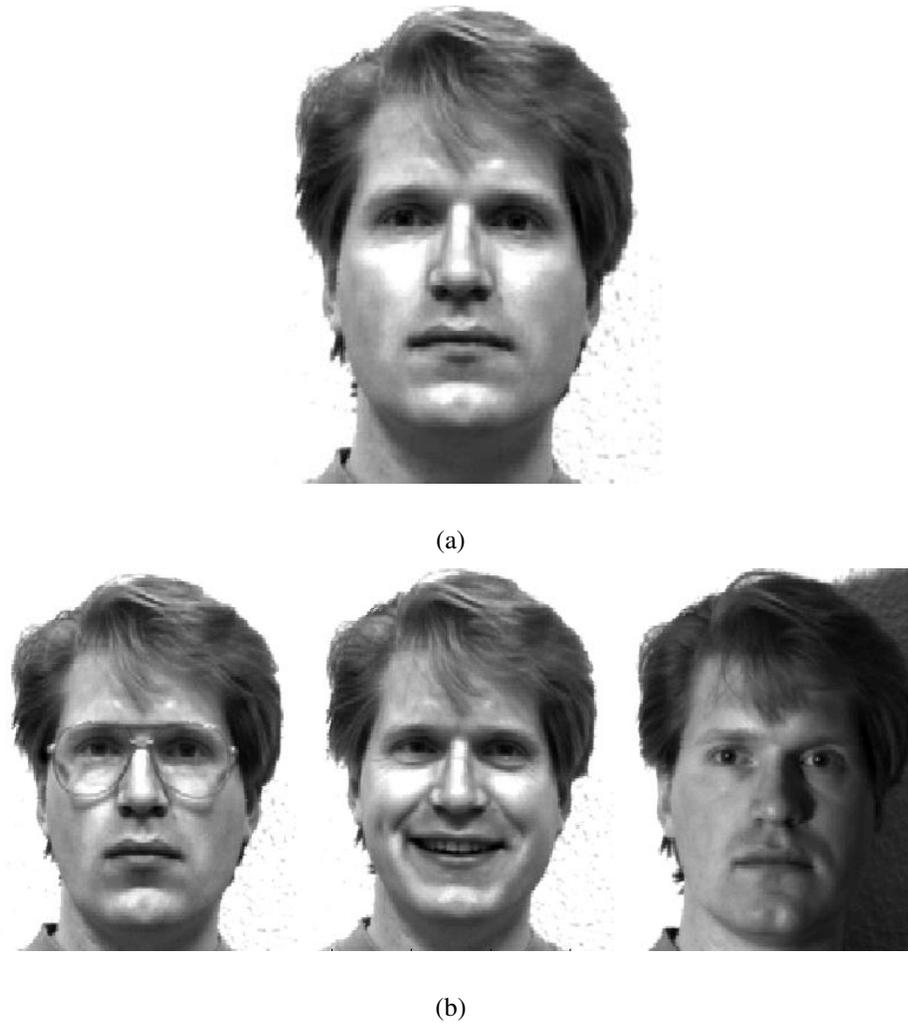
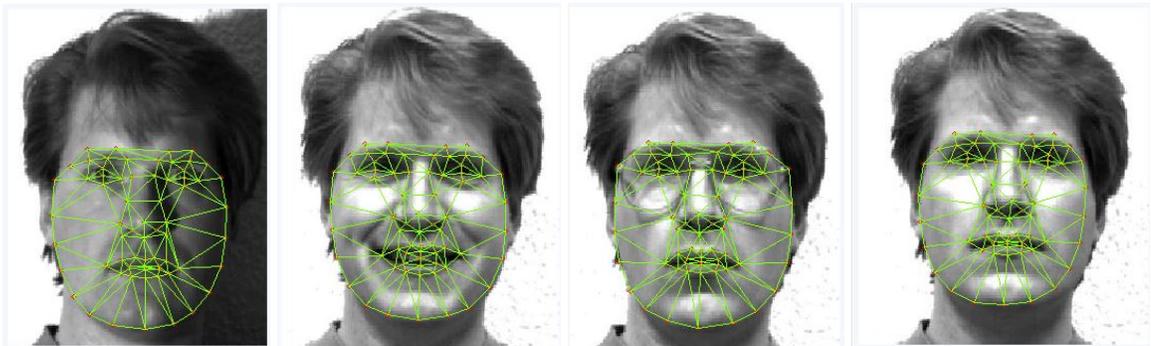


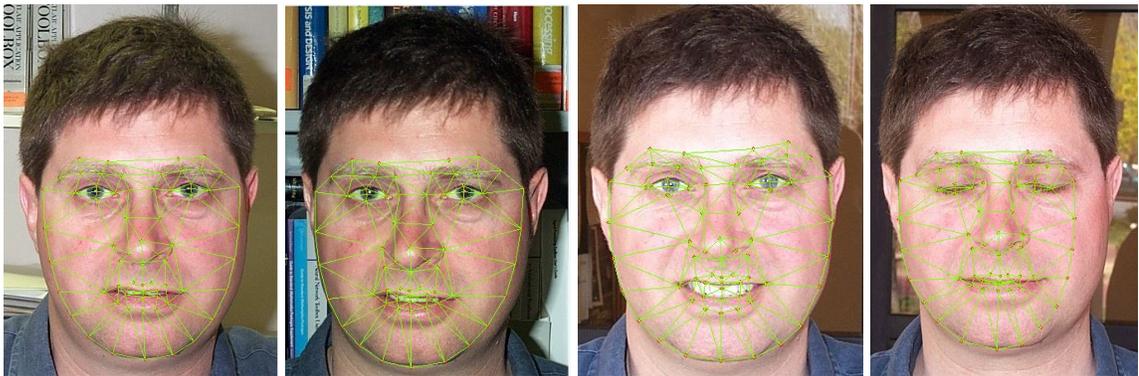
Figure 4.3: Depiction of probe and gallery images from the YFace dataset. Images used (a) as a gallery, and (b) as a probe (testing).

YFace: The Yale Face Database contains 165 images of 15 subjects stored as GIF format. There are 11 images per subject, one for each of the following facial expressions

or configurations: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink. We used normal images (without occlusion, expression, and lighting conditions) as a gallery and rest of the fourteen images as a probe to test the face recognition algorithms.

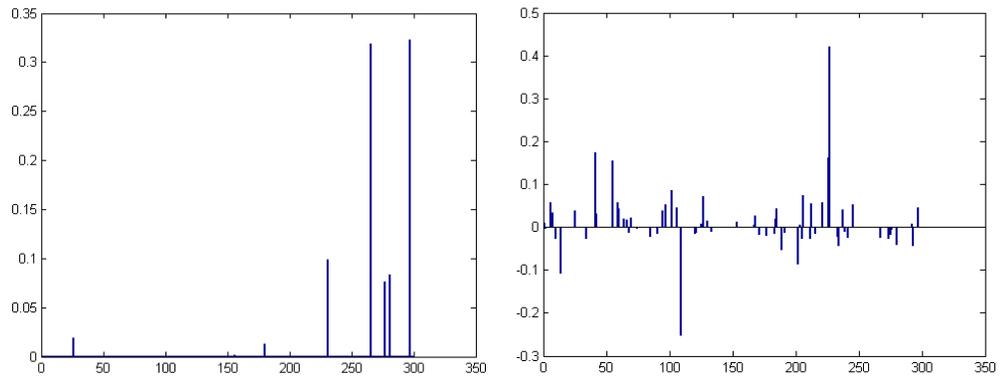


(a)



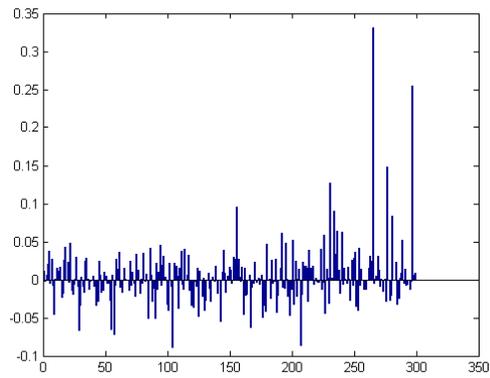
(b)

Figure 4.4: Depiction of Active Shape Model based facial landmark detection on facial images.



(a)

(b)



(c)

Figure 4.5: Sparse representation of the input image when face recognition: a) correctly identifies the subject, b) misidentifies the subject. When the input image is not a human, then the sparse representation will be random values, and the output will not be sparse. (c) Randomly generated output gives this result.

4.3.2 Preprocessing

We leveraged the Active Shape Model (ASM) to detect key anatomical landmarks in each image [16, 11, 56]. Figure 4.4 depicts the landmarks detected by the Active Shape Model on facial images. ASM may not always give good results. In such cases, we manually annotated the two eyes and the nose tip.

Using the feature points detected by ASM (or manually provided), we register all images to a reference image. To register all images to a reference image, scale, rotation and translation parameters are estimated using Procrustes Analysis [19], and warped accordingly. After registration all images are cropped and resized to 60x60 pixels.

4.3.3 Experimental Results

When the face recognition correctly identifies the subject, then the sparse representation of the input low-resolution image will be sparse; when the face recognition cannot identify the subject the sparse representation of an input image will not be sparse, it will be distributed. Figure 4.5 depicts the sparse representation output when the face recognition correctly identifies the subject. Notice that the output is sparse. However, when the face recognition fails to identify the subject, then the output of the sparse representation will not be sparse (Figure 4.5(b)). We performed another test to see the output of the sparse representation when we generate a random image, and to use it as an input image. Figure 4.5(c) depicts the output of the sparse representation when we use a random image. Notice that, when the face recognition correctly identifies the subject, the sparse representation output is sparse, and when it cannot identify the subject, the output is partially sparse. However,

when the input image is not a human facial image (random image), the output of sparse representation is not sparse at all.

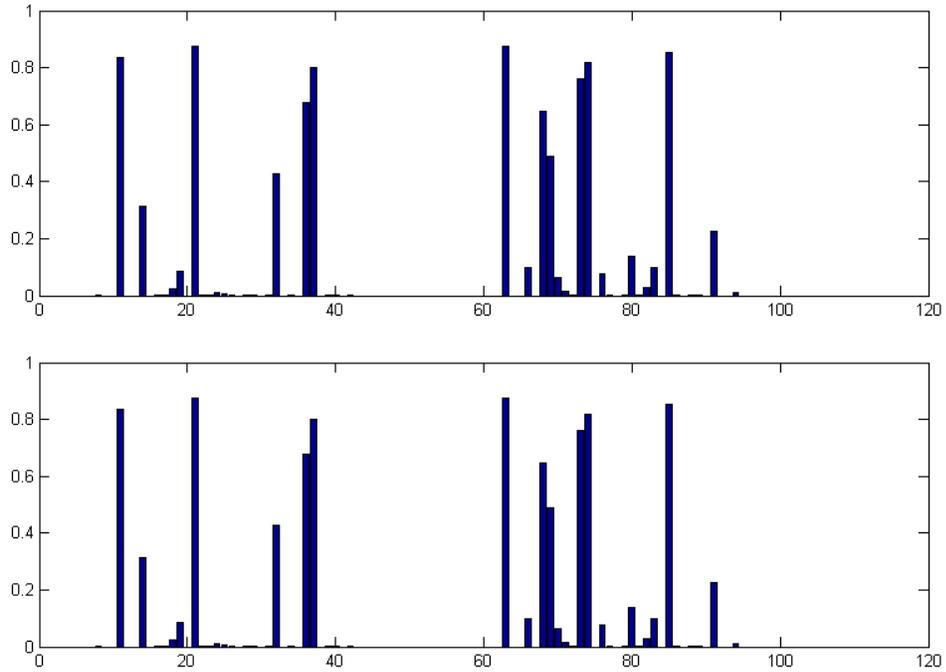


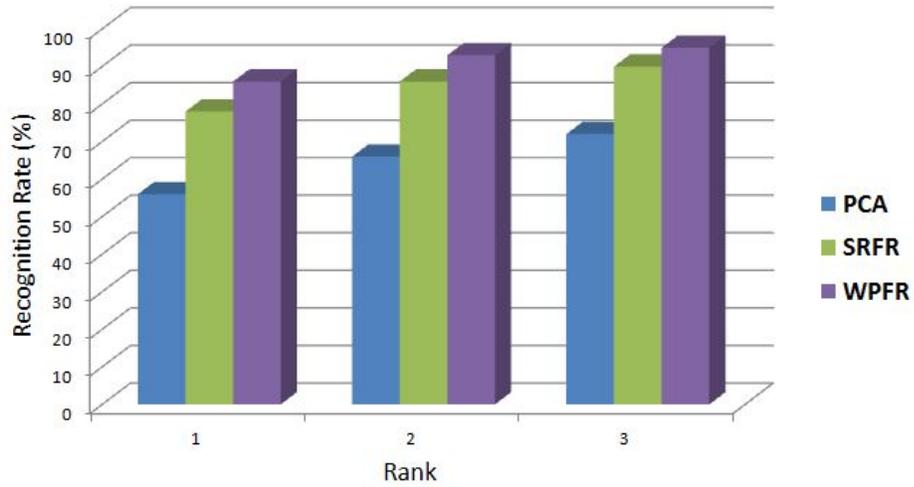
Figure 4.6: Depiction of weight matrix output during the training (two different iterations).

Figure 4.6 depicts the weight matrix that is used during the recognition stage. Since we vectorize the input data into one dimension and work in one dimension, the weight parameter also becomes one dimensional. Notice that, at two different iterations, the weight vector has only small change. Most interesting in this picture is the location of the high weights. These weights ($w_i > 0$) determine which region (indices) to trust, and how much to trust; the zero weight regions are ignored, because they mislead the recognition accuracy.

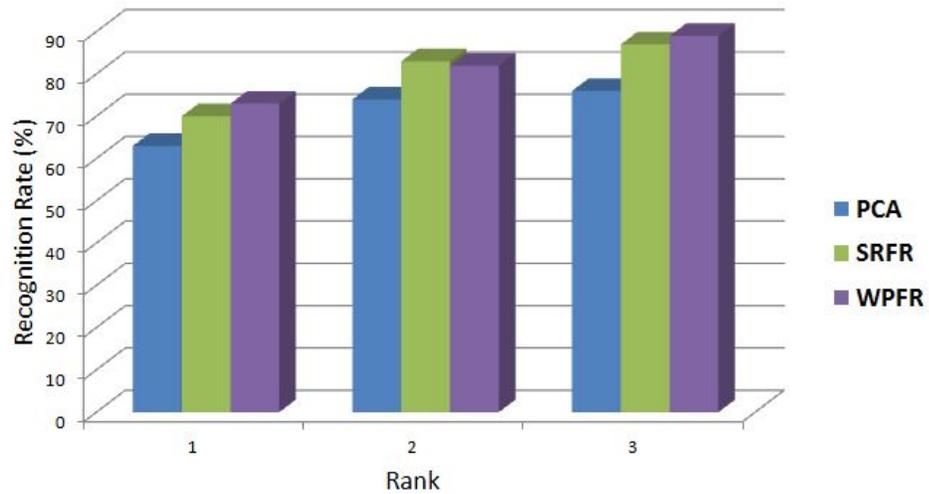
We exhaustively tested the performance of the proposed algorithm, henceforth referred

to as WPFR (Weighted Patching-based Face Recognition), for recognition accuracy and compared the results with the following standard face recognition algorithms: PCA [47] and SRFR [72]. Figure 4.7(a) depicts the rank vs. recognition rate performance plot of the proposed (WPFR) and the other FR methods for the Face99 dataset, and Figure 4.7(b) depicts the results for the YFace dataset. Note that on average the recognition rates of all FR methods are high on both datasets. This is because the resolution of probe and gallery images is high. In addition, this might be because the acquisition times between probe and gallery images are very small (acquired in the same week or month), and the number of the images per subject in the gallery is more than one. However, one can see that our proposed method, WPFR, outperforms the other FR algorithms, and has the highest Rank-1 recognition rate.

In addition, we have captured the computation times for the proposed face recognition algorithm. Overall, our algorithm is the slowest one among the PCA and SRFR due to the use of sparse representation and iterations. As mentioned earlier, the sparse presentation depends on the balancing parameter λ . When λ is low, then we run more iterations; if λ is high, then we run fewer iterations to estimate the sparse representation. For $\lambda = 0.1$ it took 0.18s, $\lambda = 0.01$ took 0.29s, and $\lambda = 0.001$ took 0.36s to identify one subject in the dataset.



(a)



(b)

Figure 4.7: Depiction of the Rank vs. Recognition Rate of face recognition algorithms on (a) Face99, and (b) YFace datasets. The Rank-1 for PCA, SRFR and WPFR on the Face99 dataset are 56%, 78%, and 86%, and the Rank-1 on the YFace dataset are 63%, 70%, and 73%, respectively.

Chapter 5

Conclusion

Image quality is an important role in image processing, computer vision, pattern recognition, etc. We have image quality.

In this dissertation we studied the image quality and its effects. We proposed a new face recognition algorithm to improve the recognition accuracy. In addition, we proposed a new algorithm to improve the resolution quality of the input low-resolution image where the output (super-resolved) image has better visual quality, as well as higher resolution compared to the input low-resolution image. We also proposed a new algorithm to reconstruct diastole/systole images from a series of MRI images.

We studied the effect face recognition, where we could measure the difference between the original (ground truth) and input (probe) image; as a distance metric we used face recognition. Our main focus in this research was to automatically recognize human subjects from a database under different image quality, occlusion, and expression.

We have used a sparse representation method in our algorithm. The whole image in the training database as well as input images are divided into patches, and the proposed face recognition using sparse representation is performed using those patches. For each patch we assigned a value (weight); based on the patch weight, the face recognition algorithm makes a global decision. Basically, the weight for each patch tells the proposed face recognition algorithm to focus on it more or less. If it is the region of interest, where it gives the correct subject id most of the time, then it will have a high weight value, otherwise it will be low. The weight parameter for each patch is computed from training images, where we know the id's of each subject in the training database. We tested the proposed face recognition algorithm on two publicly available datasets [53, 4]. These datasets contain high resolution facial images with varying lighting conditions, expression and occlusions. In terms of computation time, the proposed face recognition algorithm is slower compared to other face recognition algorithms, but the proposed face recognition algorithm performs better in terms of Rank-1 recognition accuracy.

By introducing a new face recognition algorithm to improve the recognition accuracy, we do not improve the resolution of the image for better visualization. First we studied the limitation of the existing algorithms, then we proposed a new super-resolution algorithm to improve the resolution of the input image, where the output image has higher resolution compared to the input low-resolution image. The proposed algorithm can be parallelized which can be implemented on a GPU, where the computation speed would be fast. In addition, we proposed a new approach for this algorithm to reduce the storage space for the training database, where we could reduce the computation time due to the reduction of the bandwidth and the small training database. The new super resolution

algorithm adds high-frequency information that is extracted from natural high-resolution images in the training dataset to the input image to get a super-resolution image. The selection of the high-frequency information from the training dataset is accomplished in two steps: a nearest-neighbor search algorithm is used to select similar images from the training dataset, which can be implemented on the GPU, and a sparse-representation algorithm is used to estimate a weight parameter to combine the high-frequency information of selected images. Quantitatively and qualitatively we demonstrated the superiority of the proposed super-resolution algorithm on various datasets.

In addition to the research on face recognition and super-resolution algorithms on natural and facial images, we performed research on medical images (e.g., MR images acquired from a Siemens MRI scanner). We proposed a computationally efficient method for MRI reconstruction [45]. Furthermore, it reduces the artifacts and mis-registrations. The proposed method was first tested with a synthetic dataset and then using actual data captured by an MRI scanner; its results were quantitatively validated. Also, the proposed method effectively estimated the heart rate signal from the MR image sequence thereby assisting in multi-frame reconstruction. In fact, this method could be used in clinical settings as a diagnosis tool. Importantly, there are no additional peripheral probes required during the MRI acquisition process. Knowledge of the cardiac motion patterns in real-time can be extremely valuable during an intervention. A high degree of accuracy in the reconstruction is particularly critical in the evaluation of any of existing cardiac disease. In addition, this research opens up opportunities for future investigations for diagnosis such as the assessment of patients with physiological or pathological irregularities.

5.1 Future Work

We have introduced a new approach to reconstruct a diastole/systole images from the MRI image sequence. However, we couldn't test our algorithm on large datasets with data from more than five patients. In addition, we want to compare our approach against recent state of the art reconstruction algorithms.

We also would like to continue to improve the face recognition system. Currently, we have tested it on two low-resolution datasets acquired by surveillance cameras. We would like to test it on other challenging datasets and build better face recognition algorithm that will not be tuned for the specific dataset. In addition, we would like to create a large facial image dataset, which will include all publicly available facial datasets. This dataset will be very large, and hardly any methods have been tested on this type of dataset. This might be due to the large number of the training images in the database, where *(i)* computation time will be very high, *(ii)* recognition accuracy will be low. To reduce the computation time of our proposed face recognition algorithm, we will try to parallelize it so it can be implemented on GPUs.

Chapter 6

Publications during Ph.D. Study

1. **E. Bilgazyev**, I. Uyanik, M. Unan, D. Shah, N. V. Tsekos, and E. L. Leiss. "Using motion correction to improve real-time cardiac MRI reconstruction", *In 6th International Conference on Machine Vision, London, UK, Nov. 16-17 2013.*
2. **E. Bilgazyev**, E. Yeniaras, I. Uyanik, M. Unan, and E. L. Leiss. "Quality enhancement of low-resolution image by using natural images", *In 6th International Conference on Machine Vision, London, UK, Nov. 16-17 2013.*
3. **E. Bilgazyev**, U. Kurkure, S. K. Shah, and I. A. Kakadiaris, "ASIE: Application specific image enhancement for face recognition", *In Proc. SPIE Biometric and Surveillance Technology for Human and Activity Identification X, Baltimore, ND, April 29 - May 3 2013.*
4. B. Efraty, **E. Bilgazyev**, S.K. Shah, and I.A. Kakadiaris, "Profile-based 3D-aided face recognition", *Pattern Recognition, vol.45, no. 1, pp. 43-53, 2012.*

5. M. Amer, **E. Bilgazyev**, S. Todorovic, S.K. Shah, I.A. Kakadiaris, and L. Ciannelli, "Fine-grained categorization of events in underwater videos by tracking fish", in *Proc. IEEE International Conference on Computer Vision Workshop, Barcelona, Spain, Nov. 6-13, 2011*.
6. **E. Bilgazyev**, B. Efraty, S.K. Shah, and I.A. Kakadiaris, "Improved face recognition using super-resolution", In *Proc. IEEE International Joint Conference on Biometrics, Washington DC, Oct. 11-13, 2011*.
7. **E. Bilgazyev**, B. Efraty, S.K. Shah, and I.A. Kakadiaris, "Sparse representation-based super-resolution for face recognition at a distance", In *Proc. IEEE British Machine Vision Conference, Dundee, UK, Aug. 29 - Sep. 2, 2011*.
8. **E. Bilgazyev**, S.K. Shah, and I.A. Kakadiaris, "Comparative evaluation of wavelet based super-resolution from video for face recognition at a distance", In *IEEE International Conference on Automatic Face and Gesture Recognition, Santa Barbara, CA, Mar. 21-25, pp. 559-565, 2011*.
9. B.A. Efraty, **E. Ismailov**, S. Shah, and I.A. Kakadiaris, "Towards 3D-aided profile-based face recognition", In *IEEE International Conference on Biometrics: Theory, Applications and Systems, Arlington, VA, Sep. 28-30, 2009*.
10. B. Efraty, D. Chu, **E. Ismailov**, S. Shah, and I.A. Kakadiaris, "3D-aided profile-based face recognition", In *IEEE International Conference on Image Processing, Cairo, Egypt, Nov. 7-11, 2009*.

Bibliography

- [1] A. Ashraf, S. Lucey, and T. Chen. Learning patch correspondences for improved viewpoint invariant face recognition. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, Jun. 24-26 2008.
- [2] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(9):1167–1183, 2002.
- [3] A. Bartoli. Groupwise geometric and photometric direct image registration. In *Proc. British Machine Vision Conference*, page I:157, 2006.
- [4] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997. Special Issue on Face Recognition.
- [5] O. Bernard, D. Friboulet, P. Thevenaz, and M. Unser. Variational b-spline level-set: A linear filtering approach for fast deformable model evolution. *Image Processing, IEEE Transactions on*, 18(6):1179–1191, 2009.
- [6] E. Bilgazyev, B. Efraty, S. Shah, and I. Kakadiaris. Improved face recognition using super-resolution. In *Biometrics (IJCB), 2011 International Joint Conference on*, pages 1–7, 2011.
- [7] E. Bilgazyev, S. Shah, and I. Kakadiaris. Comparative evaluation of wavelet-based super-resolution from video for face recognition at a distance. In *Proc. 9th IEEE International Conference on Automatic Face and Gesture Recognition*, pages 559–565, Santa Barbara, CA, March 2011.
- [8] E. Bilgazyev, I. Uyanik, M. Unan, D. Shah, N. V. Tsekos, and E. L. Leiss. Using motion correction to improve real-time cardiac MRI reconstruction. In *6th International Conference on Machine Vision*, London, UK, Nov. 16-17 2013.

- [9] E. Bilgazyev, E. Yeniaras, I. Uyanik, M. Unan, and E. L. Leiss. Quality enhancement of low-resolution image by using natural images. In *6th International Conference on Machine Vision*, London, UK, Nov. 16-17 2013.
- [10] E. Bilgazyev, E. Yeniaras, I. Uyanik, M. Unan, and E. L. Leiss. Sparse representation based face recognition using weighted regions. In *6th International Conference on Machine Vision*, London, UK, Nov. 16-17 2013.
- [11] A. Blake and M. Isard. Active shape models. In *Active Contours*, pages 25–37. Springer London, 1998.
- [12] Y. H. C. Ryu and H. Kim. Super-template generation using successive bayesian estimation for fingerprint enrollment. In *Proc. AVBPA 2005*, volume 3546 of *Lecture Notes in Computer Science*, page 710719, 2005.
- [13] T. Chan and L. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, 2001.
- [14] A. Chand and G. Kapoor. Generalized cubic spline fractal interpolation functions. *SIAM Journal on Numerical Analysis*, 44(2):655–676, 2006.
- [15] H. Chang, D. Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, Jun. 20-25 2005.
- [16] T. Cootes and C. Taylor. Active shape models: smart snakes. In *Proc. British Machine Vision Conference*, pages 266–275. Leeds, UK, Sep. 22-24 1992.
- [17] D. Cremers, F. Tischhuser, J. Weickert, and C. Schnrr. Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional. *International Journal of Computer Vision*, 50(3):295–313, 2002. 10.1023/A:1020826424915.
- [18] J. Felblinger, J. Slotboom, R. Kreis, B. Jung, and C. Boesch. Restoration of electrophysiological signals distorted by inductive effects of magnetic field gradients during mr sequences. *Magnetic Resonance in Medicine*, 41(4):715–721, 1999.
- [19] L. Feng, M. B. Srichai, R. P. Lim, A. Harrison, W. King, G. Adluru, E. V. R. Dibella, D. K. Sodickson, R. Otazo, and D. Kim. Highly accelerated real-time cardiac cine MRI using kt sparse-sense. *Magnetic Resonance in Medicine*, 70(1):64–74, 2013.
- [20] W. Freeman, T. Jones, and E. Pasztor. Example-based super-resolution. *Computer Graphics and Applications, IEEE*, 22(2):56–65, 2002.

- [21] W. T. Freeman and C. Liu. *Markov random fields for super-resolution and texture synthesis*, chapter 10, pages 1–30. MIT Press, 2011.
- [22] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 349–356, 2009.
- [23] L. Gorelick, A. Delong, O. Veksler, and Y. Boykov. Recursive mdl via graph cuts: application to segmentation. In *Proc. 13th IEEE International Conference on Computer Vision*, pages 890–897, Barcelona, Spain, Nov. 6-13 2011.
- [24] R. Gottumukkal and V. K. Asari. An improved face recognition technique based on modular pca approach. *Pattern Recognition Letters*, 25(4):429–436, 2004.
- [25] E. Heijman, W. de Graaf, P. Niessen, A. Nauerth, G. van Eys, L. de Graaf, K. Nicolay, and G. J. Strijkers. Comparison between prospective and retrospective triggering for mouse cardiac MRI. *NMR in Biomedicine*, 20(4):439–447, 2007.
- [26] P. Hennings-Yeomans, S. Baker, and V. Bhagavatula. Simultaneous super-resolution and feature extraction for recognition of low-resolution faces. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, Jun. 24-26 2008.
- [27] S. K. Hobbs, G. Shi, R. Homer, G. Harsh, S. W. Atlas, and M. D. Bednarski. Magnetic resonance image-guided proteomics of human glioblastoma multiforme. *J Magn Reson Imaging*, 18(5):530–6, 2003. Ca86312-0/ca/nci P41 rr09784/rr/ncrr T32 ca09696/ca/nci Journal Article Research Support, Non-U.S. Gov’t Research Support, U.S. Gov’t, P.H.S. United States Jmri.
- [28] Y. Hori, N. Yamada, M. Higashi, N. Hirai, and S. Nakatani. Rapid evaluation of right and left ventricular function and mass using real-time true-fisp cine mr imaging without breath-hold: Comparison with segmented true-fisp cine MR imaging with breath-hold. *Journal of Cardiovascular Magnetic Resonance*, 5(3):439–450, 2003.
- [29] J. Hsu, C. C. Yen, C.-C. Li, M. Sun, B. Tian, and M. Kaygusuz. Application of wavelet-based pocs super-resolution for cardiovascular MRI image enhancement. In *Multi-Agent Security and Survivability, 2004 IEEE First Symposium on*, pages 572–575, 2004.
- [30] G. Hua, M.-H. Yang, E. Learned-Miller, Y. Ma, M. Turk, D. J. Kriegman, and T. S. Huang. Introduction to the special section on real-world face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1921–1924, 2011.

- [31] K. Jia and S. Gong. Multi-modal tensor face for simultaneous super-resolution and recognition. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, Jun. 20-26 2005.
- [32] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1127–1133, 2010.
- [33] N. Kingsbury. A dual-tree complex wavelet transform with improved orthogonality and symmetry properties. In *Proc. IEEE Conf. on Image Processing*, Vancouver, September 11-13 2000. paper 1429.
- [34] N. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Appl. Comput. Harmon. Anal.*, 10:234–253, 2001.
- [35] M. Klodt and D. Cremers. A convex framework for image segmentation with moment constraints. In *Proc. 13th IEEE International Conference on Computer Vision*, pages 2236–2243, Barcelona, Spain, Nov. 6-13 2011.
- [36] M. Laudon, J. Webster, R. Frayne, and T. Grist. Minimizing interference from magnetic resonance imagers during electrocardiography. *Biomedical Engineering, IEEE Transactions on*, 45(2):160–164, 1998.
- [37] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems 19*, pages 801–808. 2007.
- [38] K. C. Li and M. D. Bednarski. Vascular-targeted molecular imaging using functionalized polymerized vesicles. *Journal of Magnetic Resonance Imaging*, 16(4):388–393, 2002.
- [39] S. Li and A. Jain. *Handbook of Face Recognition*. SpringerLink : Bücher. Springer, 2011.
- [40] Y. Li and W. A. P. Smith. Face recognition using patch-based spin images. In *Proc. Int. Conf. on Pattern Recognition*, volume 1, pages 20–24, 2006.
- [41] G. S. Mageras and E. Yorke. Deep inspiration breath hold and respiratory gating strategies for reducing organ motion in radiation treatment. *Seminars in Radiation Oncology*, 14(1):65–75, 2004.
- [42] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *Image Processing, IEEE Transactions on*, 17(1):53–69, 2008.

- [43] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, pages 416–423 vol.2, 2001.
- [44] R. Merrifield, J. Keegan, D. Firmin, and G. Yang. Reduced k-space encoding for dual-contrast truefisp imaging. *Magnetic Resonance in Medicine*, 49(3):595–598, September 25-28 2003.
- [45] F. Meyer. Learning and predicting brain dynamics from fmri: A spectral approach. In *Proc. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6701, 2007.
- [46] J. Milles, Y. M. Zhu, G. Gimenez, C. R. Guttmann, and I. E. Magnin. MRI intensity nonuniformity correction using simultaneously spatial and gray-level histogram information. *Comput. Med. Imag. Graph.*, 31(2):81–90, March 2007.
- [47] B. Moghaddam, W. Wahid, and A. Pentland. Beyond eigenfaces: probabilistic matching for face recognition. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 30–35, 1998.
- [48] F. Odille, S. Uribe, P. G. Batchelor, C. Prieto, T. Schaeffter, and D. Atkinson. Model-based reconstruction for cardiac cine MRI without ecg or breath holding. *Magnetic Resonance in Medicine*, 63(5):1247–1257, 2010.
- [49] T. Okabe, I. Sato, and Y. Sato. Spherical harmonics vs. haar wavelets: basis for recovering illumination from cast shadows. In *Proc. Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–50 – I–57 Vol.1, jun 2004.
- [50] J.-S. Park and S.-W. Lee. An example-based face hallucination method for single-frame, low-resolution facial images. In *Proc. IEEE Transactions on Image Processing*, volume 17, pages 1806–1816, Oct. 2008.
- [51] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *Signal Processing Magazine, IEEE*, 20(3):21–36, 2003.
- [52] A. Pednekar, I. Kakadiaris, R. Muthupillai, and S. Flamm. Automatic segmentation of the left ventricle from dual contrast cardiac mr data. In *Proc. 4th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 991–992, Houston, TX, Oct. 23-26 2002.
- [53] P. Perona. Face (1999) Frontal, <http://www.vision.caltech.edu/archive.html>, 1999.

- [54] N. Phatak, S. Maas, A. Veress, N. Pack, E. D. Bella, and J. Weiss. Strain measurement in the left ventricle during systole with deformable image registration. *Medical image analysis*, 13(2):354–61, apr 2009.
- [55] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *Image Processing, IEEE Transactions on*, 12(11):1338–1351, 2003.
- [56] M. Rogers and J. Graham. Robust active shape model search. In *Computer Vision ECCV 2002*, pages 517–530. 2002.
- [57] A. D. Scott, J. Keegan, and D. N. Firmin. Motion in cardiovascular mr imaging. *Radiology*, 250(2):331–351, 2009.
- [58] D. Sebald and J. Bucklew. Support vector machine techniques for nonlinear equalization. *IEEE Transactions on Signal Processing*, 48(11):3217–3226, Nov 2000.
- [59] N. Shafique and M. W. Haering. A rank constrained continuous formulation of multi-frame multi-target tracking problem. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, Jun. 24-26 2008.
- [60] J. Sun, N.-N. Zheng, H. Tao, and H.-Y. Shum. Image hallucination with primal sketch priors. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, pages II–729–36 vol.2, 2003.
- [61] M. E. Tipping and C. M. Bishop. Bayesian image super-resolution. In *Advances in Neural Information Processing Systems*, pages 1303–1310, 2003.
- [62] M. Tistarelli, S. Z. Li, and R. Chellappa. *Handbook of Remote Biometrics: for Surveillance and Security*. Springer Publishing Company, Incorporated, 1st edition, 2009.
- [63] M. Tistarelli, S. Z. Li, and R. Chellappa. *Handbook of Remote Biometrics: for Surveillance and Security*. Springer Publishing Company, Incorporated, 1st edition, 2009.
- [64] C. Tu, W.-L. Hwang, and J. Ho. Analysis of singularities from modulus maxima of complex wavelets. *IEEE Transactions on Information Theory*, 51(3):1049–1062, 2005.
- [65] I. Uyanik, P. Lindner, P. Tsiamyrtzis, D. Shah, N. Tsekos, and I. T. Pavlidis. Applying a level set method for resolving physiologic motions in free-breathing and non-gated cardiac mri. In *Functional Imaging and Modeling of the Heart*, pages 466–473. 2013.

- [66] H. van Assen, M. Danilouchkine, A. Frangi, S. Ordas, J. Westenberg, J. Reiber, and B. Lelieveldt. SPASM: A 3D-ASM for segmentation of sparse and arbitrarily oriented cardiac MRI data. *Medical Image Analysis*, 10(2):286–303, April 2006.
- [67] J. Wang, S. Zhu, and Y. Gong. Resolution enhancement based on learning the sparse association of image patches. *Pattern Recognition Letters*, 31(1):1–10, Jan. 2010.
- [68] Q. Wang, X. Tang, and H. Shum. Patch based blind image super-resolution. In *Proc. IEEE International Conference on Computer Vision*, Beijing, China, Oct. 17-20 2005.
- [69] Y. Wang, P. J. Rossman, R. C. Grimm, S. J. Riederer, and R. L. Ehman. Navigator-echo-base real-time respiratory gating and triggering for reduction of respiration effects in three-dimensional coronary mr angiography. *Radiology*, 198(1):55–60, 1996.
- [70] F. W. Wheeler and A. J. Hoogs. Moving vehicle registration and super-resolution. *2012 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 0:101–107, 2007.
- [71] L. Wiskott. The role of topographical constraints in face recognition. *Pattern Recognition Letters*, 20(1):89–96, January 1999.
- [72] J. Yang, H. Tang, Y. Ma, and T. Huang. Face hallucination via sparse coding. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1264–1267, 2008.
- [73] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.
- [74] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *Image Processing, IEEE Transactions on*, 19(11):2861–2873, 2010.
- [75] M. Yang, D. Zhang, J. Yang, and D. Zhang. Robust sparse coding for face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 625–632, 2011.
- [76] L. Zhang, H. Zhang, H. Shen, and P. Li. A super-resolution reconstruction algorithm for surveillance images. *Signal Processing*, 90(3):848–859, 2010.
- [77] Y. Zheng, S. Englander, S. Baloch, E. Zacharaki, Y. Fan, M. Schnall, and D. Shen. STEP: Spatiotemporal enhancement pattern for MR-based breast tumor diagnosis. *Medical Physics*, 36(7):3192, 2009.

- [78] Y. Zhou and J. Bai. Atlas-based fuzzy connectedness segmentation and intensity nonuniformity correction applied to brain MRI. *IEEE Trans. on Biomedical Engineering*, 54(1):122–129, 2007.
- [79] Y. Zhou, N. Tsekos, and I. Pavlidis. Cardiac MRI intervention and diagnosis via deformable collaborative tracking. In *Functional Imaging and Modeling of the Heart*, pages 188–194. 2011.