MACHINE LEARNING AND ITS APPLICATION IN AUTOMATIC CHANGE DETECTION IN MEDICAL IMAGES

VARVARA NIKA

A DISSERTATION SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM MATHEMATICS & STATISTICS
YORK UNIVERSITY
TORONTO, ONTARIO

November 2014

Abstract

Change detection is a fundamental problem in various fields, such as image surveillance, remote sensing, medical imaging, etc. The challenge of change detection in medical images is to detect disease-related changes while rejecting changes caused by noise, patient position change, and imaging acquisition artifacts such as field inhomogeneity. In this thesis, first, we overview the existing change detection methods, their underlying mathematical frameworks and limitations. Second, we present our contributions in solving the problem. We design optimal subspaces to approximate the background image in more efficient fashion. This is based on our structure principal component analysis, aiming to capture the structural similarity between scans in the context of change detection. We theoretically and numerically discuss the proper choices of norms used in the subspace approximation.

The mathematical frameworks developed in this thesis consist of: (i) a new mathematical model to change detection by defining it as an optimization problem involving a cost function, input and output image sets, projection onto a subspace, and a similarity measure; (ii) development and implementation of numerical pipelines to compute the clinical changes by designing four mathematical algorithms; (iii) refining our algorithms by introducing the co-registration step utilizing the local dictionaries; and (iv) two new structure subspace learning models that are robust to outliers and noise, reduce the dimensionality of the dataset, and computationally efficient. We defined the co-registration step as a minimization problem involving a cost function, input and output image sets, a set of transform functions, projection onto a subspace, and a similarity measure.

Based on the mathematical frameworks discussed above, numerical schemes are developed to automatically filter out clinically unrelated changes and identify true structure

changes that may be of clinical importance. Our approaches are data-driven and utilize the knowledge of machine learning. We quantitatively analyze the performance of these algorithms using both synthetic and real human data. Our work has the potential to be used in computer aided diagnosis.

Dedication

This thesis would have not been possible without the great support provided by many people in my life and I would like to take the opportunity to thank them all.

I am very grateful to my supervisor, Dr. Hongmei Zhu for being present in this great journey. I look back to all these years together and I am very grateful and forever in debt to her for introducing me to the beautiful field of Medical Imaging. I am thankful for her patience, the advice, the unlimited support, the readiness to help, her kindness and her great attitude. She has guided me during the past five years in both the dissertation work and the professional presentations. She always encouraged and supported my research ideas. Her constant support helped me to overcome many obstacles during my thesis. Her insightful comments and constructive criticisms have been invaluable to me. I have learned a great deal and because of her I felt calm, relaxed, and had a lot of joy doing the research, even when I had to stay late at night. I will cherish these times forever.

My co-supervisor, Dr. Paul Babyn, has been very helpful throughout my thesis. He always made time for my weekly presentations, despite his overwhelming schedule. He encouraged me in every step of my work and provided the necessary medical expertise with his invaluable perspective, as a very well known radiologist. He helped me consider different viewpoints in my research.

I am grateful to all my graduate studies professors and staff at York University. Professor Walter Whiteley has been a very inspiring figure throughout my time here at York. It all started with him and Margaret Sinclair who saw potential and believed in me. Walter introduced me to the graduate program and taught me very valuable lessons on how to conduct mathematical research.

Professor Michael Chen has provided the necessary guidance on optimization topics

and has been very supportive and helpful especially during the last year of my studies. Professor Michael Zabrocki has been very helpful and supportive during my graduate years at York University. He also taught me a very interesting course in Image Processing, where my research interest has its beginnings.

All my undergraduate and graduate studies professors at Tirana University have had a great impact on my progress. Their instruction style provided an inspiration to me. I strive to make my classroom environment a great place to learn as they made theirs.

My colleagues and staff at Toronto Catholic District Board, the Mathematics department at TCDSB, especially my math colleagues and administration at Francis Libermann CHS, who have helped me throughout the graduate years by providing the necessary flexibility. I thank them for all they have done for me. I would like to thank specifically my good friend Angie Pizzirusso who taught me that there is always a bright side to everything, Fidelis McGrath for making me laugh with his sharp and very smart humor, Sona Kevork for her constant reminder to take care of myself, Stefana Penelea for her calm, caring, supportive attitude, and always cheering for me, Kathy Kubota who reminded me that anyone presenting a problem to the team should also present a solution on how to solve the problem.

This thesis would have been impossible without the love, believes, and sacrifices of my parents, especially during my high school and undergraduate years. My father Mosko provided me with the utmost support. My mother Jorgjia prepared me to overcome life challenges and was always there for me. I know that they are watching me from heaven and, once again, they are very proud of me. My warm and very kind family back home, especially my sister Margarita, my two brothers Theodhor and Gaqo, my three beautiful nieces Eva, Bianka and Lona, and nephews Juli, Reno, Kristian, Aldo, and Mateo. My very special thanks to my family: my sons, Ilvan and Enea, who have always been my inspiration and my number one supporters throughout the years; my husband, Ilia, who has always been beside me. His unlimited love, support and help during our life together has been critical to my success.

Acknowledgments

I would like to thank MITACS and NSERC for the financial support that helped fund parts of the research in this dissertation.

Contents

List of Tables

List of Figures

Related Publications

Related Journal Publications

1. Nika, V., Babyn, P., and Zhu, H., EigenBlock Algorithm for Change Detection An Application of Adaptive Dictionary Learning Techniques," Journal of Computational Science, Vol. 5, Issue 3, pp 527 - 535, 2013.

2. Nika, V., Babyn, P., Zhu, H., Change detection of medical images using dictionary learning techniques and principal component analysis. J. Med. Imag., Vol. 1, issue 2, pp 024502, 2014.

Related Conference Proceedings

1. Nika, V., Babyn, P., Zhu, H. AEDL Algorithm for Change Detection in Medical Images An Application of Adaptive Dictionary Learning Techniques, in Proc. 15th Int. Conf. on Medical Imaging Computing and Computer Assisted Intervention, MICCAI: STMI, pp 14-21, 2012.

2. Nika, V., Babyn, P., and Zhu, H., Change Detection of Medical Images Using Dictionary Learning Techniques and PCA, Medical Imaging 2014: Computer-Aided Diagnosis, in Proc. of SPIE Vol. 9035, 903506, 2014.

Chapter 1

Introduction

1.1 Background

Change detection is the process of identifying differences in the state of an object or phenomenon by observing it at different times [28]. As illustrated in Fig. 1.1.1, change detection algorithms are used to identify regions of changes in multiple images of the same scene taken at different times. Change detection techniques and algorithms are critical to many applications in several areas, such as remote sensing, video surveillance, medical imaging, etc. These techniques have witnessed continuous development over the years, and as a result, there are now various algorithms, methods, and automated systems available.

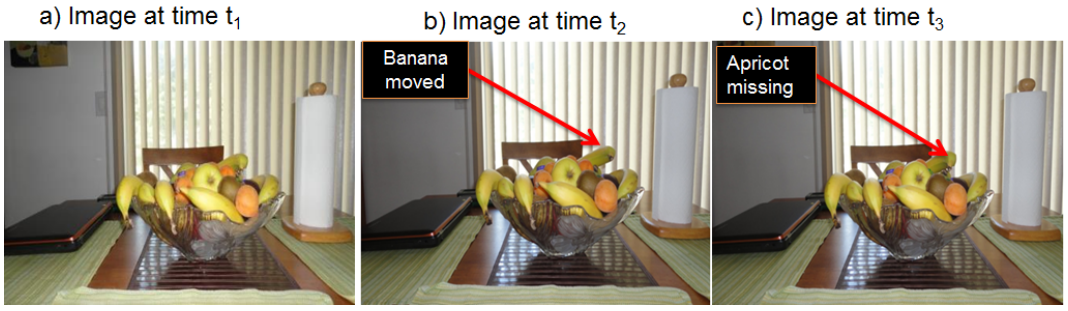

Figure 1.1.1: a) through c) images of the same scene, my kitchen table, taken at three different times t_1, t_2, and t_3 respectively. At time t_2, the only change is the position of one of the fruits, i.e., the banana has been rotated, or the tip of the banana has being pulled up. One of the fruits is not present at time t_3, mimicking a lesion that has disappeared.

In particular, identifying the changes in medical images taken at different times is of great relevance in clinical practice. Change detection is relevant to all imaging modal-

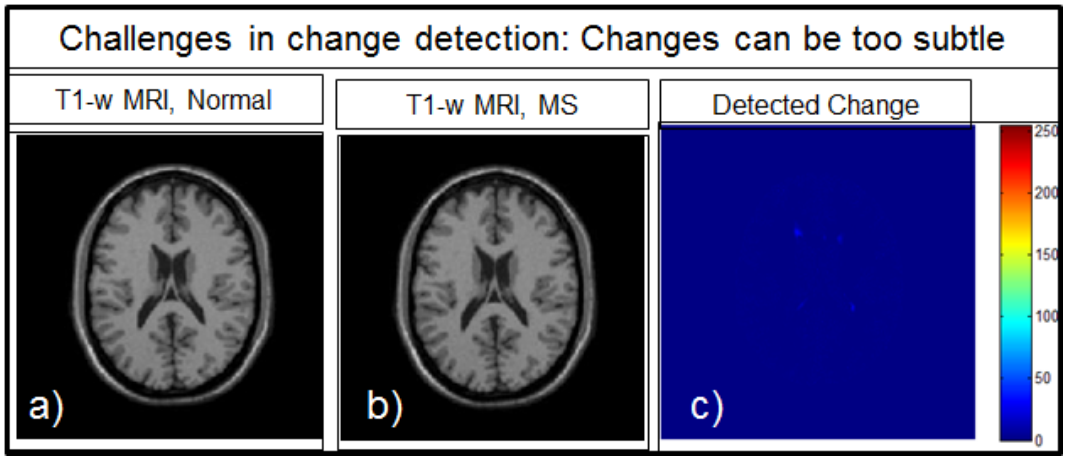

Figure 1.1.2: a) T1-weighted MRI of a normal brain. b) T1-weighted MRI of a brain with MS lesions, c) image containing only disease related changes. Images from http://mouldy.bic.mni.mcgill.ca/brainweb.

ities such as MRI, CT, Sonography etc, but we focus here mostly on MRI. Imaging datasets may include multiple sequences each consisting of many images obtained at one time needing comparison with the immediate previous study or multiple prior studies. The size of datasets that radiologists are dealing for CT or MRI can be hundreds or thousands of images for each time point. The key issue of change detection algorithms in serial MR images is to detect disease-related changes while rejecting 'unimportant ones' induced by noise, mis-alignment, and other acquisition-related artifacts, such as intensity inhomogeneity,[4].

Despite the diversity of approaches [2, 3, 4], a change detection algorithm usually consists of many common pre-processing steps to suppress or filter out 'irrelevant changes' before making change detection decision and using the core algorithm to determine the set of pixels that are significantly different from the previous images and are disease-related. In medical diagnosis and treatment, serial MRI examinations are often performed on patients with diseases such as cancer or multiple sclerosis; radiologists routinely detect subtle changes in images of the same anatomical location that may be clinically significant, as shown in Fig. 1.1.2. Trying to find subtle changes in extent or character, by using a side-by-side presentation mode of films, can be very difficult. Radiologists usually try to visually correct errors, due to patient repositioning, and they use their professional knowledge to identify and reject certain artifacts. Challenges the radiologists face during visual comparison include:

- Scanner software and hardware related changes (variation in appearance due to pulse sequences, acquisition parameters, gradient coils, RF inhomogeneities, registration).
- Separation of acquisition changes from disease related changes
- Information overload
- Change Blindness, the inability to detect changes in objects or scenes being viewed, and the inability to make comparisons between two scenes
- Satisfaction of Search
- Change occurs in an unexpected location
- Change occurs eonly in a part of a complicated lesion
- Side-by-side presentation is very poorly matched to the human visual system.

Computer automated systems for identifying, correcting, or ignoring these artifacts, would be very helpful. Authors in [27] made the case that automated change detection systems would be a great help for the radiologists to correctly interpret data. In [7, 8, 9] the MRI examinations of brain tumors are studied and the automated techniques are used to improve the results. The main purpose of these studies was to reduce human error and minimize the enormous amount of data that radiologists have to process to reach a conclusion. The authors found that implementing a scientifically useful tool is most clinically viable when it is efficiently integrated into clinical work flow. They concluded that automated change detection can improve efficiency, accuracy and reader agreement.

An efficient computer-based system that automatically reduces the quantity of data and directs radiologists' attention to clinically-relevant areas would be very useful. The automated change detection system created by them,[7, 8, 9], was a great improvement over previous automated system, however, the process is inherently time-consuming and the tissue classification task itself remains very difficult.

Authors in [4] reviewed a large number of change detection algorithms and they

classified them into two groups, statistical analyses and background modeling techniques. Authors in [2] presented an automatic change detection system for serial MRI with applications in multiple sclerosis - follow ups. It is one of the statistical analysis methods and it is based on the use of multimodal information for change detection, Generalized Likelihood Ratio Test, and nonlinear joint histogram normalization. The performance of the algorithm is low when noise is not-stationary.

The work of [3, 7, 8] is among the statistical analysis methods that is perhaps better recognized in the medical field. They implemented an integrated system for detecting changes in serial multi-spectral MRI examination, based on post classification of image pixels in multi-spectral MR intensity feature space. Their rational for using multi-spectral space was based on the observation that an abnormal tissue may "look like" a tissue transitioning from one normal tissue to another in the feature space and an assumption that change tends to occur along lines connecting pairs of cluster centroids in the feature space. The detected changes were presented in the form of a color-coded change map, superimposed on the anatomical images. The system additionally formats the output as a quantitative summary. Preliminary clinical studies show that their system can visually identify subtle changes related to disease. However, the tissue classification task itself is very difficult; in addition, the whole process of calculating transition tissue types and fractional membership for each pixel is inherently time-consuming.

Another change detection method for interval MR images is proposed in [11] using a general nonparametric statistical framework based on local steering kernels. Calculation of test statistics was derived from cosine similarity. However, their work does not address registration nor alignment, and also it is limited to one imaging modality.

Among background modeling methods, background subtraction has been used especially in video surveillance. In [10], background subtracted images are recovered using compressive sensing (CS). This works when important changes occupy a small portion of the test image, and therefore the changed image is sparse in the spatial domain. Assuming that both background and foreground fulfills the sparsity criteria, they solve the problem using L_1 minimization with total variation (L1 TV) algorithm [5, 6].

Another breakthrough work [19] proposed the use of principle component pursuit method to detect foreground changes in video surveillance. The work is based on low-rank and sparse decomposition of image matrices. Robust Principal Component Analysis (PCA) has applications in many other areas, such as face recognition, etc. This method applies to a series of video frames and may be considered for a series of MR images as well.

Authors in [20] use robust dictionary learning to solve the background subtraction problem. Their approach appears to produce a better dictionary than more traditional K-SVD algorithm [21]. However, the same assumptions for sparsity must hold. Any application of CS to background subtraction models involves the use of various L_1 minimization algorithms. Many MRI reconstruction techniques employ compressive sensing methods. Among others, work in [18] is very well known for a direct application to MR images of the brain. It uses a well known fact that MR images are sparse on some domains such as wavelet, finite differences, etc. An undersampled MR image is recovered by using L_1 minimization which allows for more rapid MR imaging.

Authors in [22] applied dictionary learning techniques to solve the reconstruction problem. They proposed an adaptive patch-based sparsifying learning dictionary which is obtained using the k-space data and is used to remove aliasing and noise. The dictionary is created using the K-SVD algorithm and is updated for each image block.

One of the major challenges in change detection algorithms for medical images is to detect disease-related changes while rejecting changes caused by noise and acquisition-related artifacts such as mis-alignment and intensity inhomogeneity. Despite the diversity of approaches [2, 3, 4, 7, 8, 9, 10, 11], a change detection algorithm usually consists of many common pre-processing steps to suppress or filter out "unimportant" changes before making change detection decisions and using the core algorithm to determine the set of pixels that are significantly different from the initial reference image and are disease-related. The sequence of pre-processing steps complicate the algorithm as a whole, increase the processing time, and most important, may distort clinical relevant information in the original images.

Work in [15] addressed mis-alignment in change detection problem by employing

a series of sparse optimization problems. However, their method only works well for highly sparse images, such as synthetic aperture radar (SAR) images that are much sparser than most medical images.

Authors in [1, 13] used eigenfaces for face recognition and pattern recognition by finding the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images, where each image is treated as a vector. Eigenvectors are called the eigenfaces and each of them accounts for a different amount of the variation among the images. Then, each test image is compared with many training reference images from a database. In our work we consider a training set obtained from only one image.

These challenges motivated us in designing three algorithms that automatically tolerates noise and acquisition-related artifacts and capture subtle and important clinical changes between two or more medical images that are not necessary spatially sparse.

1.2 The Objectives of the Dissertation

Computer-aided diagnosis (CAD) has become one of the major research subjects in medical imaging and diagnostic radiology. Radiologists in collaboration with mathematicians, physicists, and researchers in other related fields are working on designing and developing new CAD methodologies [95, 96]. Creating an automated system for detecting the changes of consecutive scans in medical images remains an important area of computer-aided diagnosis. The increasing need for early, reliable and accurate disease detection and diagnosis generates interdisciplinary challenges in acquiring, processing and interpreting these medical images.

This thesis is inspired by the new developments in the mathematics of medical images, which have led to exciting and innovative ways to improve and develop medical imaging technology. The main objective of the dissertation is development of a new mathematical framework for change detection and design of new algorithms that can be used to detect clinical changes automatically while rejecting unimportant changes due to patient position, noise, and other related acquisition artifacts.

We developed three variants of automated change detection systems for medical

images: Adaptive EigenBlock Dictionary Learning (AEDL), EigenBlock Change Detection (EigenBlockCD), and EigenBlock Change Detection 2 (EigenBlockCD-2). They take two or more images as input and automatically produce an image which contains only clinically related changes. The core ideas of these algorithms are based on local dictionary learning and background modeling.

Our key technical contributions are in the area of subspace learning and the proper selection of similarity measure in subspace learning. We use principal component analysis (PCA) to re-express the columns of the dictionary learning in more meaningful ways, i.e., for dimensionality reduction, feature extraction, computational efficiency, background modeling, sparse representation, and for reducing the redundancy among columns of the dictionary. We developed a new Structure Principal Component Analysis method, S-PCA, more robust to outliers and better selects the features required in our change detection algorithms.

Our first algorithm, (Adaptive EigenBlock Dictionary Learning (AEDL), performs image registration locally to capture the local spatial changes in the test image using a series of local sparse minimization processes and the knowledge of Compressive Sensing. The method takes two images as input, and gives two images as output, the sparse recovery of the reference image aligned with the test image, and the change detection image. The reconstructions of overlapping blocks using L_1 minimization algorithms makes the AEDL algorithm computationally expensive.

To reduce the running time of the algorithm, we designed the second algorithm, EigenBlock Change Detection (EigenBlockCD), which uses the L_2 norm as a similarity measure to learn the dictionary. Both, the AEDL and the EigenBlockCD algorithms detect clinical relevant changes while ignoring insignificant changes due to patient's position, noise, and other acquisition related artifacts, when these unrelated disease changes are within a small neighbourhood. To account for large sizes of insignificant changes, i.e., large shifts and rotations, we designed the third algorithm, EigenBlock Change Detection 2 (EigenBlockCD-2). We also extended all three algorithms for 3 dimensional volumetric MR scans.

1.3 Dissertation Outline

The rest of this dissertation is organized as follows. Problem formulation is given in Section 1.4. In Chapter 2, we review the mathematical frameworks of existing change detection methods relevant to our research. These methods are classified into three broad overlapping groups: General change detection methods are discussed in Section 2.1, background modeling in Section 2.2, and change detection problems in medical images in Section 2.3. Some of the methods include differencing, statistical-based, background modeling, sparsity-based methods, etc. We emphasize the recent mathematical advancement in the area of compressed sensing and dictionary learning.

In Chapter 3, we introduce Adaptive EigenBlock Dictionary Learning Algorithm (AEDL), [23], the first algorithm that we propose in this thesis. We describe the mathematical foundation of the AEDL algorithm, which is inspired by the recent results in compressed sensing theory. We propose a solution based on dictionary learning via projection onto Eigen-subspace. To demonstrate the accuracy of the algorithm, we present simulations with both synthetic and real MR images of human brain.

The second algorithm, EigenBlock Change Detection (EigenBlockCD), [24], is presented in Chapter 4. We build the EigenBlock dictionary by applying principal component analysis (PCA) to achieve dimensionality reduction and feature extraction.Then, we make the case for selecting the L_2 norm as similarity measure between corresponding blocks in reference and test images, and demonstrate the properness and effectiveness of using L_2 minimization. The performance of the EigenBlockCD algorithm is demonstrated through numerical simulations on synthetic and real MR images of human brain.

In Chapter 4 we also provide a thorough analysis of different similarity measures, such as L_1 and L_2. We first analyze the properties of both norms related to the projection onto orthonormal subspaces, and rotation. Then, we provide theoretical and numerical investigation of the proper selection of similarity measures in change detections, i.e., L_2 versus L_1 norm.

To account for large ranges of shifts and rotations, and to improve computational efficiency, an improved version of the EigenBlock CD algorithm, EigenBlockCD-2,

[25, 26], is presented in Chapter 5. We introduce an additional step, pre-alignment of two consecutive images named as reference and test image, before detecting changes. This co-registration algorithm ultilizes EigenBlockCD to find the optimal values of shifts and rotation that co-registers the two sets of images the best. We numerically demonstrate the advantages of this pre-processing step to change detection. Finally, we describe the EigenBlockCD-2 algorithm to detect clinical relevant changes of two consecutive MR scans of human brain. Numerical simulations with both synthetic and real MR images are provided.

In Chapter 6, we provide the performance analysis of the AEDL, the EgenBlockCD, and the EigenBlockCD-2 algorithms, [25, 26]. We describe the quantitative and qualitative measures that can be used to validate our algorithms. Numerical simulations between two consecutive synthetic images and serial MR images are presented. We compare the performance analysis of our algorithm with the other ones in the literature.

In Chapter 7 we extend our algorithm to three dimensional volumetric data. Numerical simulations with real MR images are shown.

Chapter 8 is dedicated to our newest contribution, Structure Principal Component Analysis (S-PCA). We provide the properties of S-PCA and compare it with some other recent non weighted L_2 and $L_{(2,1)}$ norm-based PCA. We investigate the performances of standard PCA and the SPCA using L_2 or $L_{(2,1)}$ minimizations, and introduce the weighted L_2 and $L_{(2,1)}$ - S-PCA , as a better principal component-based analysis for detecting the changes of MR images.

In Chapter 9, we provide the conclusions of this dissertation research and discuss the road map for future work.

1.4 Problem Formulation

Let $\mathbf{I}_1, \mathbf{I}_2, \cdots, \mathbf{I}_n$, represent images of size $N_1 \times N_2$, of the same anatomic location, taken at different times, t_1, $t_2, \cdots, t_n$ respectively. Let $\mathbf{I}_{cd}(1,2)$, $\mathbf{I}_{cd}(2,3), \cdots, \mathbf{I}_{cd}(n-1,n)$ represent the true changes between $\mathbf{I}_1$ and $\mathbf{I}_2$, $\mathbf{I}_2$ and $\mathbf{I}_3, \cdots, \mathbf{I}_{n-1}$ and $\mathbf{I}_n$, where true changes are not caused by patient position or noise as shown in Fig. 1.4.1. Let $\mathbf{I}_s(i,j)$,

$s = 1, 2, \cdots, n$, $i = 1, \cdots, N_1, j = 1 \cdots, N_2$, be image intensity of a pixel at i-th row and j-th column.

The question is: Can we automatically detect true changes while ignoring the changes that are clinically unrelated?

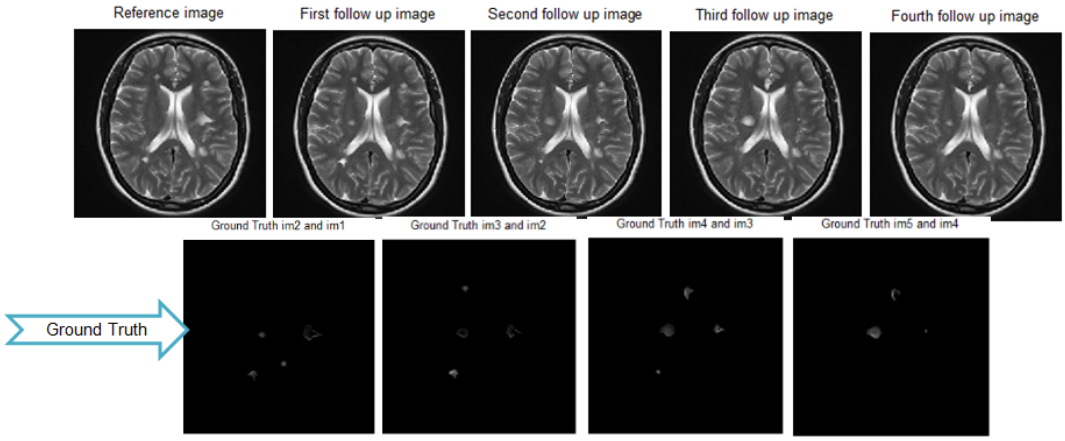

Figure 1.4.1: Top row: T2-weighted MR images of the brain taken at different times, t_1, $t_2, \cdots, t_5$. Bottom row: Four images representing the ground truth (true changes) between two consecutive MR images in the first row

.

Definition of change detection of an image set: Let the set of input images of the same size, $N_1 \times N_2$, be $\mathcal{I}_{in} = \{\mathbf{I}_i, i = 1, 2, \cdots, n\}$, where $\mathbf{I}_i$ is the image corresponding to time t_i. We define the change image set by $\mathcal{I}_{out}$:

$$\mathcal{I}_{out} = \mathcal{G}(\mathcal{I}_{in}, \mathcal{T}, \mathcal{C}, \rho) \tag{1.4.0.1}$$

where:

- $\mathcal{G}$: The mathematical model.
- $\mathcal{I}_{in}$: Input image set.
- $\mathcal{T}$: A set of transform functions, i.e., change of basis functions, such as those of wavelet transform, Fourier transform, etc.
- $\mathcal{C}$: The cost function to be minimized using a similarity measure, which compares and checks feature changes between $\mathbf{I}_{t_1}$ and $\mathbf{I}_{t_2}$. Features in spatial domain for some applications include the intensity of a pixel/voxel or a block of pixels/voxels from the same location.

- ρ: Similarity measure, i.e., distance measures such as L_1, L_2, $L_{(2,1)}$, cosine similarity, similarity maps, etc.

- $\mathcal{I}_{out}$: Output, a set of images with changes between $\mathbf{I}_{i-1}$ and $\mathbf{I}_i$, that is, $\mathcal{I}_{out} = \{\mathbf{I}_{cd}(i-1,i), i = 2, 3, \cdots, n\}$.

For simplicity, we aim to detect changes between two consecutive images, $\mathbf{I}_1$ and $\mathbf{I}_2$, initial (reference) and follow-up (test), taken at two different times. Similar to Eq. (1.4.0.1) we define $\mathbf{I}_{cd}$ as the image of changes between $\mathbf{I}_1$ and $\mathbf{I}_2$:

Definition of change detection between two images:

$$\mathbf{I}_{cd} = \mathcal{G}(\mathbf{I}_1, \mathbf{I}_2, \mathcal{T}, \mathcal{C}, \rho) \tag{1.4.0.2}$$

where:

- $\mathcal{G}$: The mathematical model.

- $\mathbf{I}_1$, $\mathbf{I}_2$: Input images of the same size.

- $\mathcal{T}$: A set of transform functions.

- $\mathcal{C}$: The cost function to be minimized.

- ρ: Similarity measure.

- $\mathbf{I}_{cd}$: Output, the image of changes between $\mathbf{I}_1$ and $\mathbf{I}_2$.

An automated change detection algorithm between two images, $\mathbf{I}_1$ and $\mathbf{I}_2$, takes these two images as input and produces a change image, $\mathbf{I}_{cd}$, as an output. A change detection algorithm should also be able to identify whether the change is considered significant (important) or insignificant according to a particular application, therefore the algorithm should maximize the number of significant changes found, while minimizing the insignificant ones. Furthermore, the output should be the one that is the closest to the real (true) change image, from all the change images determined. The problem can be formulated as:

Problem Formulation:

Given $\mathbf{I}_1$, $\mathbf{I}_2$, find the optimal $\mathcal{G}$, $\mathcal{T}$, $\mathcal{C}$, and ρ which are then used to compute $\mathbf{I}_{cd}$.

Desirable solutions require an automated system to:

- Detect changes in medical images which are subtle and disease related, while rejecting changes caused by patient position, noise, and other acquisition related artifacts.
- Reduce the quantity of data presented to the radiologist.
- Present changes in the way that matches the perception of the human eye, are easy to read, see, and understand.

Chapter 2

Literature Review

We discuss the major approaches towards the categorization or classification of change detection methods in general. This task has been undertaken several times and we will present relevant previous work, as well as our perspective on the recent developments in the field.

Authors in [4] and [29] surveyed a large number of image change detection algorithms and created a very useful classification. Most methods surveyed in their work are based on either statistical analyses or background modeling techniques. For the purposes of our work, in Sections 2.1 through 2.3 we provide a slightly different classification and describe those methods that are still relevant in current applications. We categorize these methods according to those used in general applications and those used particularly for medical images. We elaborate on the background modeling as it is one of the most popular methods and also is the fundamental approach of our algorithms.

We classify the change detection methods in three overlapping groups, change detection methods in general, background modeling methods, and change detection methods used in medical images. We focus more on the methods and algorithms described in Sections 2.2 and 2.3 as our research work presented in this thesis apply knowledge mostly from the background modeling methods and dictionary learning.

Change detection methods in general are discussed in Section 2.1, which include methods based on differencing and statistical hypothesis test methods. Background modelling methods are presented in Section 2.2. They include parametric, non-parametric,

sparsity and CS-based, and robust PCA methods. Change detection methods applied specifically to medical images are discussed in Section 2.3. We present the advantages and discuss the limitations of these approaches.

2.1 Change Detection in General Applications

We discuss various methods based on differencing in Section 2.1.1, and methods based on statistical hypothesis tests in Section 2.1.2.

2.1.1 Differencing

Simple differencing and its closely related methods such as image rationing, change vector analysis (CVA) and gradient images make a pixelwise comparison between images. Therefore, images must go through the preprocessing step of registration.

2.1.1.1 Image Differencing

As in Section 1.4, $\mathbf{I}_1$ and $\mathbf{I}_2$ represent two images of size $N_1 \times N_2$ of the same scene, taken at two different times, t_1 and t_2 respectively. The change image between $\mathbf{I}_1$ and $\mathbf{I}_2$ is denoted as $\mathbf{I}_{cd}$. A changed mask $B(i,j)$ is obtained by a simple thresholding as:

$$B(i,j) = \begin{cases} 1 & \text{if } |\mathbf{I}_{dif}(i,j)| \geq \tau; \\ 0 & otherwise. \end{cases} \tag{2.1.1.1}$$

where $\tau \rightarrow$ represent the threshold value and $I_{dif}(i,j)$ is the image difference obtained by pixelwise simple differencing method as:

$$\mathbf{I}_{dif}(i,j) = \mathbf{I}_2(i,j) - \mathbf{I}_1(i,j) \tag{2.1.1.2}$$

$\mathbf{I}_s(i,j)$ represent the image intensity of a pixel at row i and column j, $s = 1, 2$, $i = 1, \cdots, N_1$ and $j = 1, \cdots, N_2$.

The value of τ is chosen empirically or statistically by a specialist using several

thresholding selection methods. It is preferred that either both images are normalized with zero mean, or only the second image $\mathbf{I}_2(i,j)$ is normalized to have the same mean and the same standard deviation as $\mathbf{I}_1(i,j)$, i.e.,

$$\widetilde{\mathbf{I}}_2(i,j) = \frac{\sigma_1}{\sigma_2}(\mathbf{I}_2(i,j) - \mu_2) + \mu_1 \tag{2.1.1.3}$$

Then, the image difference is computed as:

$$\mathbf{I}_{dif}(i,j) = \widetilde{\mathbf{I}}_2(i,j) - \mathbf{I}_1(i,j) \tag{2.1.1.4}$$

Finally, the change image pixel $I_{cd}(i,j)$ is equal to the absolute difference $I_{dif}(i,j)$ everywhere the values of the mask $B(i,j)$ are one and is zero otherwise:

$$I_{cd}(i,j) = \begin{cases} |\mathbf{I}_{dif}(i,j)| & \text{if } B(i,j) = 1; \\ 0 & otherwise. \end{cases} \tag{2.1.1.5}$$

Simple differencing is sensitive to misregistration, to noise, variation in illumination. Global thresholding does not take into consideration local features of the change, and it might result in false positives or false negatives.

2.1.1.2 Image Rationing

Image rationing method calculates $\mathbf{I}_{ratio}(i,j)$, the ratio between the first image and the normalized second image.

$$\mathbf{I}_{ratio}(i,j) = \frac{\widetilde{\mathbf{I}}_2(i,j)}{\mathbf{I}_1(i,j)} \tag{2.1.1.6}$$

If the values of $\mathbf{I}_{ratio}(i,j)$ is close to one, then pixel (i,j) appears in both images with the same intensity, which suggests the pixel intensity has not changed and the pixel is part of the background: other values suggest a change. Similar to image differencing, to decide whether (i,j) pixel belongs to the change image or not, we need to know how big the change is, or if it is greater than a selected threshold. The choice of the threshold depends on the application as before. The changed mask is created

by thresholding $\mathbf{I}_{ratio}(i,j)$:

$$B(i,j) = \begin{cases} 0 & \text{if } 1-\tau \leq \mathbf{I}_{ratio}(i,j) \geq 1+\tau; \\ 1 & otherwise. \end{cases} \tag{2.1.1.7}$$

The image difference $\mathbf{I}_{dif}$ is computed as in Eq. (2.1.1.4), then the $I_{cd}(i,j)$ is determined with a very small value of τ as in Eq. (2.1.1.5).

2.1.1.3 Change Vector Analysis

Change Vector Analysis (CVA) was first used by [32] to identify and analyze changes between pairs of multi-date spectral data. A change vector is the difference vector between two vectors in N-dimensional feature space defined for two images of the same scene during two different dates as shown in Fig. 2.1.1 a). CVA computes spectral change vectors and compares their magnitudes to a specified threshold as shown in Fig. 2.1.1 b).

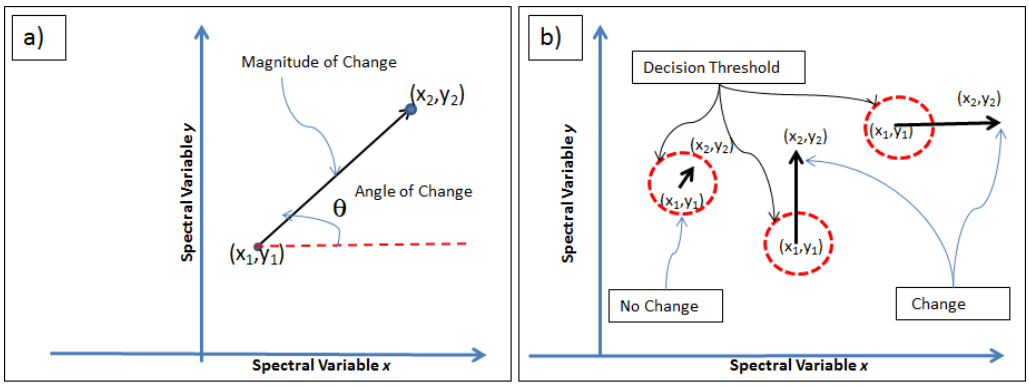

Figure 2.1.1: a) An illustration of change vector analysis; b) An illustration of decision making constraint by a specific threshold. The magnitude of change equals to $\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$, and the angle θ equals to $\tan^{-1}\left(\frac{y_2-y_1}{x_2-x_1}\right)$

.

The method processes and analyses change in all multi-spectral input data concurrently. Although CVA method for detecting changes avoids compounding of spatial-spectral errors which may be inherited from multi-date classifications, it requires spatial registration, spectral transformation, accurate geometric registration and radiometric normalization of the input data, as indicated in [29] and [33].

2.1.1.4 Gradient Images

Authors in [34] applied gradient image methods for automatically detecting changes in video surveillance applications. Gradient image method introduces structural information within the background model of a change detection algorithm. It captures the structural information by means of low-level features which is done by estimating the gradient components and then analyzing the edges.

The first step of this method is to obtain a reduced-resolution image, $\mathbf{I}_{1R}$ and $\mathbf{I}_{2R}$, from $\mathbf{I}_1$ and $\mathbf{I}_2$ respectively, by a δ-scale factor as Fig. 2.1.2 shows. $\mathbf{I}_{sR}(i,j)$ is the average gray-level of a $\delta \times \delta$ block centered at (i, j) position from $\mathbf{I}_1$ and $\mathbf{I}_2$ respectively. The reduced-resolution image $\mathbf{I}_{sR}(i,j)$ is computed as:

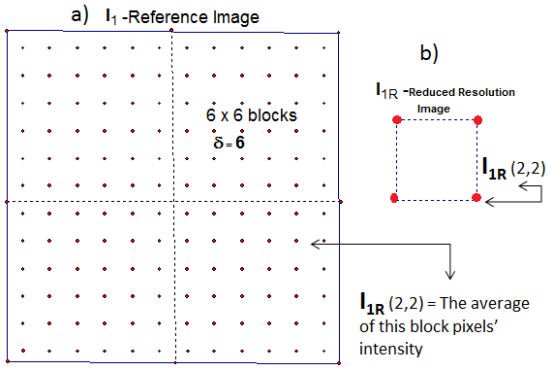

Figure 2.1.2: a) Reference image of size 12×12; b) The reduced-resolution image of size $\frac{12}{\delta} \times \frac{12}{\delta}$ with $\delta = 6$.

$$\mathbf{I}_{sR}(i,j) = \frac{1}{\delta^2} \sum_{k=0}^{\delta-1} \sum_{l=0}^{\delta-1} \mathbf{I}_s(\delta \times i + k, \delta \times j + l) \tag{2.1.1.8}$$

where $s = 1, 2$. Each reduced-resolution image $\mathbf{I}_{sR}$ is now of size $\frac{N_1}{\delta} \times \frac{N_2}{\delta}$.

Secondly, the structures of $\mathbf{I}_1$ and $\mathbf{I}_2$ is determined, by computing the gradient in the i and j directions of the reduced-resolution images for all (i,j) pixels as:

$$D_i\mathbf{I}_s(i,j) = \mathbf{I}_{sR}(i+1,j) - \mathbf{I}_{sR}(i,j) \tag{2.1.1.9}$$

$$D_j\mathbf{I}_s(i,j) = \mathbf{I}_{sR}(i,j+1) - \mathbf{I}_{sR}(i,j) \tag{2.1.1.10}$$

where $s = 1, 2$ corresponds to the reference and test image respectively. The structure of each image is then calculated a set of all the ordered pairs:

$$S(\mathbf{I}_s) = \{(D_i\mathbf{I}_s, D_j\mathbf{I}_s)|\forall(i,j) \in \mathbf{I}_s\} \tag{2.1.1.11}$$

Thirdly, a delta-structure image, $\Delta S[\mathbf{I}_1, \mathbf{I}_2]$, is computed:

$$\Delta S[\mathbf{I}_1, \mathbf{I}_2] = \max_{d\in\{x,y\}} \{|D_d\mathbf{I}_1(i,j) - D_d\mathbf{I}_2(i,j)|\} \tag{2.1.1.12}$$

Finally, the delta-structure image is thresholded to discriminate between the true changes and those caused by changes in the brightness. This method is designed to detect changes in video surveillance where the initial background structure has been estimated by using many frames. Medical image modalities do not have the number of frames which image surveillance methods require. Therefore, in our change detection problem we aim to detect changes between two or more consecutive frames in 2 or 3 dimensional spaces.

2.1.2 Statistical Hypothesis Test

In this section, we will discuss three of the many available statistical hypothesis test methods: significance tests, likelihood ratio tests, and probabilistic mixture models.

2.1.2.1 Significance Test

Deciding if a pixel belongs to the foreground (change) or the background (no change) is related to choosing one out of two hypothesis, the null hypothesis $\mathcal{H}_0$ implies no change and $\mathcal{H}_1$ implies change. Under null hypothesis, the only change appearing in $I_{dif}(i,j)$ is due to noise [30]. If we model the noise with Gaussian distiribution with zero mean and σ_0^2 variance, then the conditional probability of a pixel (i,j) from the

difference image I_{dif} under the null hypothesis is:

$$p(I_{dif}(i,j)|\mathcal{H}_0) = \frac{1}{\sqrt{2\pi\sigma_0^2}} exp\left(-\frac{I_{dif}^2(i,j)}{2\sigma_0^2}\right) \tag{2.1.2.1}$$

Then the significance test $S(i,j)$ is:

$$S(i,j) = p(I_{dif}(i,j)|\mathcal{H}_0) \lessgtr \tau \tag{2.1.2.2}$$

This shows that a pixel (i,j) belongs to the background if $p(I_{dif}(i,j)|\mathcal{H}_0) < \tau$, or to the foreground otherwise. This can easily be extended to a block-based significance test. Let $\mathbf{B}_m$ be a block with m pixels from $I_{dif}(i,j)$, and $\mathrm{I}_{dif}^{(m)}(i,j)$ be the pixel values in the block centered at position (i,j). The significance test for the block can be calculated by:

$$p(I_{dif}^m(i,j)|\mathcal{H}_0) = \left(\frac{1}{\sqrt{2\pi\sigma_0^2}}\right)^m exp\left(-\frac{\sum\limits_{(i,j)\in\mathbf{B}_m} I_{dif}^2(i,j)}{2\sigma_0^2}\right) \tag{2.1.2.3}$$

To eliminate block artifacts, overlapping blocks are considered, although it is more computationally expensive.

2.1.2.2 Likelihood Ratio Test (LRT)

Likelihood ratio test is based on the comparison of **likelihood ratio** to a threshold τ. The likelihood ratio is calculated as:

$$\mathcal{L}(i,j) = \frac{p(I_{dif}(i,j)|\mathcal{H}_1)}{p(I_{dif}(i,j)|\mathcal{H}_0)}$$

Calculating $p(I_{dif}(i,j)|\mathcal{H}_1)$ is not straightforward. In some applications [30], both hypothesis are modelled by a Gaussian distribution with zero mean and σ_0^2 and σ_1^2 standard deviation respectively. σ_0^2 can be estimated from a number of very small unchanged regions in I_{dif}, and similarly, σ_1^2 is calculated from a very large changed

region of I_{dif}. We can then compute

$$p(I^m_{dif}(i,j)|\mathcal{H}_s) = \left(\frac{1}{\sqrt{2\pi\sigma_s^2}}\right)^m exp\left(-\frac{\sum\limits_{(i,j)\in\mathbf{B}_m} I^2_{dif}(i,j)}{2\sigma_s^2}\right)$$

for $s = 0, 1$. The likelihood ratio of a block $\mathbf{B}_m$ is:

$$\mathcal{L}^m(i,j) = \frac{p(I^m_{dif}(i,j)|\mathcal{H}_1)}{p(I^m_{dif}(i,j)|\mathcal{H}_0)} = \left(\frac{\sigma_0}{\sigma_1}\right)^m exp\left(-\sum_{(i,j)\in\mathbf{B}_m} I^2_{dif}(i,j)\left(\frac{1}{2\sigma_1^2} - \frac{1}{2\sigma_0^2}\right)\right)$$

One of the limitations of this method is that properly selecting threshold τ is a difficult task.

2.1.2.3 Probabilistic Mixture Models

Authors in [31] and [36] proposed an automatic technique for the analysis of the difference image based on the Bayes theorem, under the assumption that pixels in the difference image are independent of one another. The method minimizes the overall change detection error probability. It performs an unsupervised estimation of means and variances of the changed and unchanged pixels in the difference image by an iterative method based on the Expectation-Maximization (EM) algorithm. The method assumes that probability density function $p(\mathbf{I}_{dif})$ of pixel values in the difference image $\mathbf{I}_{dif}$ can be modeled as a mixture density distribution:

$$p(\mathbf{I}_{dif}(i,j) = p(\mathbf{I}_{dif}(i,j)|\mathcal{H}_0) \times P(H_0) + p(\mathbf{I}_{dif}(i,j)|\mathcal{H}_1) \times P(H_1)$$

where $p(\mathbf{I}_{dif}|\mathcal{H}_0), P(H_0), p(\mathbf{I}_{dif}|\mathcal{H}_1)$ and $P(H_1)$ are estimated by applying EM algorithm as a generalization of maximum-likelihood (ML) estimation for incomplete data.

Authors in [35] took the idea of probability mixture model framework further. They propose a method which decomposes the appearance change into multiple causes due to object or camera motion, illumination phenomena, specular reflections, and "iconic

or pictorial changes". Motion is modeled as warpping, illumination change by a smooth function that amplifies image contrast, specular reflection, which are local and located near regions of high curvature, are modeled by near saturation of image intensity, and iconic or pictorial (i.e; eye blinking) are considered to have high correlation with the object and therefore are learned using eigenspace techniques. Iconic change is represented as a linear combination of learned basis images. Also, changes that do not belong to any class or that are not modelled by four classes are called outliers.

Let $\mathbf{I}((i,j),t)$ represent an image at location (i,j) at time t. They assume that this image can be explained using only five causes, $\mathbf{I}_M$ for motion, $\mathbf{I}_L$ for illumination variations, $\mathbf{I}_S$ for specular reflections, $\mathbf{I}_P$ for pictorial changes, and lastly, $\mathbf{I}_O$ for outliers. Let a_k, $k = 1, ..., 5$, be the k^{th} cause, $\mathbf{I}_k((i,j),t;a_k)$ be the predicted model image at time t caused by a_k, $p_k(\mathbf{I}_k(i,j),t)|a_k,\sigma_k)$ be the correspondent component probability, and the residual $\Delta\mathbf{I}_k$ be calculated as the difference of the image and one of the predicted image models:

$$\Delta\mathbf{I}_k = \mathbf{I}((i,j),t) - \mathbf{I}_k((i,j),t;a_k)$$

then the probability mixture is:

$$p_k(\mathbf{I}_k(i,j),t)|\{a_k,\sigma_k\}_{k=1}^5) = \sum_{k=1}^{5} w_k(i,j)p_k(\mathbf{I}_k(i,j),t)|a_k,\sigma_k)$$

where σ_k is a scale parameter, and $\sum_{k=1}^5 w_k(i,j) = 1$

$$p_k(\mathbf{I}_k(i,j),t)|a_k,\sigma_k) = \frac{2\sigma_k^3}{\pi(\sigma_k^2 + \Delta\mathbf{I}_k^2)^2}$$

with $\pi = \sum_{(i,j)} \frac{w_k(i,j)}{N}$. The EM - algorithm calculates $w_k((i,j);\sigma_k)$:

$$w_k((i,j);\sigma_k) = \frac{p_k(\mathbf{I}(i,j),t)|a_k,\sigma_k)}{\sum_{l=1}^{5} p_l(\mathbf{I}(i,j),t)|a_l,\sigma_l)}$$

This method incorporates many appearance changes caused by multiple object motions, shadows, specular reflections and deformable models in a single framework. The

likelihood function is robust and drops more rapidly than a Gaussian distribution toward the tales, as shown in Fig. 2.1.3. This speeds up the separation of the appearance changes into more distinct classes.

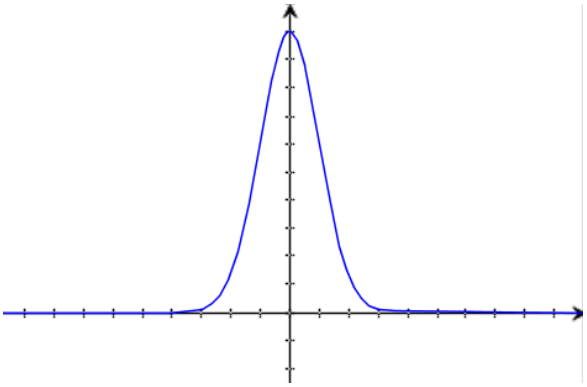

Figure 2.1.3: Robust likelihood probability distribution

2.2 Background Modeling

Background modeling and background subtraction are well known techniques that are used in video surveillance systems. In such systems the detection of changes is achieved by comparing each new frame with a modeled background image [38]. This is known as background subtraction. Background modeling is in fact the selection of features which represent the background (i.e., pixel intensity).

There are various classifications for background modeling methods, [4] and [38]. We introduce the following classification: parametric, non-parametric, sparsity based, compressed sensing based, and robust PCA methods, which are discussed in Sections 2.2.1 through 2.2.5.

2.2.1 Parametric Methods

The most well known method of the group is Mixture Of Gaussian (MOG) techniques and its derivations. The MOG is a parametric method that models the values of a particular pixel as mixture of Gaussians. Then, as described in [41], based on the persistence and the variance of each of the Gaussians of the mixture, the method determines whether a pixel belongs to background or foreground. There are two parameters used in this method: α - the learning constant and T - the proportion of data that

should be accounted for the background. The model uses a series of pixel values for gray values or a vector for color images:

$$\{x_1, ..., x_t\} = \{\mathbf{I}(i_0, j_0, k) : 1 \leq k \leq t\}$$

where $\mathbf{I}$ is the image sequence. The recent history of each pixel $\{x_1, ..., x_t\}$ is modeled by a mixture of s Gaussians. The probability of observing the current pixel is:

$$P(x_t) = \sum_{k=1}^{s} \omega_{k,t} * \eta(x_t, \mu_k, t, \mathbf{\Omega}_{\omega_{k,t}})$$

where $\omega_{k,t}$ is the weight (portion of delta for this Gaussian) of this k^{th} Gaussian in the mixture at time t, μ_k is the mean value of the k^{th} Gaussian in the mixture at time t, $\mathbf{\Omega}_{\omega_{k,t}}$ is the covariance matrix of the k^{th} Gaussian in the mixture at time t, and $\eta(x_t)$ is the Gaussian probability density function:

$$\eta(x_t, \mu_k, t, \mathbf{\Omega}_{\omega k,t}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\mathbf{\Omega}_{\omega_{k,t}}|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_t - \mu_t)^T \mathbf{\Omega}_{\omega_{k,t}}^{-1}(x_t - \mu_t)}$$

where $\mathbf{\Omega}_{\omega_{k,t}} = \sigma^2 \mathbf{I}$ for simplicity.

2.2.2 Non-Parametric Methods

In this section, we describe background subtraction methods based on kernel density estimation (KDE, [38]) and also an adaptive version of the KDE method from [37]. The background is modeled as a probabilistic model by kernel density estimation. A probability distribution function is estimated for each pixel and it is computed directly from background data, without prior knowledge of the underlying distribution. There is no model selection or distribution parameter estimation, unlike the MOG which tries to fit Gaussians to each pixel.

Kernel density estimation techniques [38] are non-parametric methods with many applications in video surveillance. The idea behind the KDE is to construct a statistical representation of the background scene that does not change over time and to build statistical representations of the moving objects or the scene that changes over time,

the foreground. This technique assumes that the intensity value of a pixel over time in a static scene is modeled with a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, and that noise is modeled by a zero mean Gaussian distribution $\mathcal{N}(0, \sigma^2)$.

Let $\mathbf{I}_1$, $\mathbf{I}_2$,..., $\mathbf{I}_N$ be N frames from video surveillance, then x_1, x_2, $\cdots$,x_N, represents a sequence of samples of intensity values for a pixel x at time point t, where $t = 1, 2, \cdots, N$. The method uses kernel density estimation to estimate the probability density distribution of a pixel intensity at any given time t. The probability of observing x_t is:

$$\mathcal{P}(x_t) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{K}_\sigma(x_t - x_i) \tag{2.2.2.1}$$

where $\mathcal{K}_\sigma$ is a kernel function and σ is the bandwidth. By choosing a Gaussian kernel function, the probability density distribution is given as:

$$\mathcal{P}(x_t) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{1}{2}\frac{(x_t - x_i)^2}{\sigma^2}\right) \tag{2.2.2.2}$$

The method uses a probability estimate to determine whether a pixel x_t belongs to foreground or background.

$$x_t \in \begin{cases} \text{Foreground} & \text{if } \mathcal{P}(x_t) < \tau; \\ \text{Background} & otherwise. \end{cases}$$

where τ is a global threshold. The choice of a suitable kernel bandwidth is very important and may be difficult to make. Larger values of σ lead to over-smoothing and smaller values of σ lead ragged density estimate. For an infinite number of samples, the choice of σ is insignificant. In practical applications, the number of samples is finite; therefore, [38] proposed to use a different kernel bandwidth σ for each pixel: $\sigma = \frac{m}{0.068\sqrt{2}}$. This value derives from the equation:

$$\mathcal{P}(N(0, 2\sigma^2) > m) = 0.25 \tag{2.2.2.3}$$

where $N(0, 2\sigma^2)$ is the distribution for the deviation $(x_i - x_{i+1})$ and m is the median of absolute deviation of all the consecutive intensity values of pixels in the sample, i.e.,

$$m = median\{|x_{i+1} - x_1|, i = 1, 2, ..., N\}$$

Two major problems associated with the non-parametric kernel density estimation methods are the long processing time and the large memory requirement. Authors in [37] modified the original KDE method by using the first frame to initialize the KDE background model. The KDE Gaussian model is updated at every frame and the probability obtained by the KDE method is added to the prior probability density at every frame. The probability $\mathcal{P}(x_t)$ is based on each pixel and can be expressed as:

$$\mathcal{P}(x_t) = \hat{\mathcal{P}}(x_{t-1}) + \frac{1}{G_t\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left(\frac{x - x_t}{\sigma}\right)^2\right) \tag{2.2.2.4}$$

2.2.3 Sparsity Based Methods

Sparsity

Background modeling methods which are reviewed in our work, are based mainly on sparse representations and compressive sensing theory. In this section we include the necessary definitions and main results for both sparse representations and compressive sensing relevant to background modeling methods and our algorithms in particular. Authors in [62] discussed sparse solutions to the systems of equations derived from sparse modeling of signals and images. We present the problem definition as outlined in [62].

Sparse Solutions

Let $\mathbf{A} \in \mathcal{R}^{n \times m}$, with $n < m$, be a full-rank matrix. Consider the underdetermined linear system of equations $\mathbf{A}\ \mathbf{x} = \mathbf{b}$. As we know, there are an infinite number of solutions for such a system. In many problems it is reasonable to look for those solutions based on some criteria. For example, if we are interested in a solution of $\mathbf{A}\ \mathbf{x} = \mathbf{b}$, which minimizes a given function $G(\mathbf{x})$, then the general optimization problem can be

expressed as follows:

$$\text{Problem P:} \quad \min_{\mathbf{x}} G(\mathbf{x}) \quad \text{subject to: } \mathbf{Ax} = \mathbf{b} \tag{2.2.3.1}$$

The type of function $G(\mathbf{x})$ determines the uniqueness of the solution. If function $G(\mathbf{x})$ is convex, the above problem has a unique solution. For example, if $G(\mathbf{x})$ is the squared Euclidean norm of x, i.e., $||x||_2^2$, the minimization problem in Eq. (2.2.3.1), namely problem P_2, can be written as:

$$\text{Problem } P_2: \quad \min_{\mathbf{x}} ||x||_2^2 \quad \text{subject to: } \mathbf{Ax} = \mathbf{b} \tag{2.2.3.2}$$

Problem P_2 in Eq. (2.2.3.2) has a unique solution $\hat{\mathbf{x}}$ which is called the minimum-norm solution and it is computed explicitly as:

$$\hat{\mathbf{x}} = \mathbf{A}^{\dagger}\mathbf{b} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b} \tag{2.2.3.3}$$

However, L_2 norm measures the energy of the signal. Minimizing the L_2 norm will result in minimizing the energy of the system. When the signal is sparse, minimizing the L_2 norm is equivalent to minimizing the energy of the signal. This may not lead to the solution with the expected sparsity.

Natural images are not necessary sparse in the spatial domain. However, they can be sparsely represented in a transformed domain such as Fourier, Wavelet, etc. Therefore, many image processing tasks can be transformed into finding sparse solutions to underdetermined systems of linear equations. Simple examples include using the discrete wavelet transform (DWT) in image coding (JPEG-2000). Approximating the image by zeroing the coefficients with small values and quantizing the large ones produces a smaller number of coefficients to be efficiently stored. Also, sparse coefficients can be inverted easily to obtain a very good approximate representation of the original image. The success of the DWT in image coding is thus directly tied to its ability to sparsify image content more than discrete cosine transform (DCT) representation used in JPEG. The sparsity of signal representation is used in image compression, image denoising, image deblurring and many other image processing tasks.

The L_0 "norm" of a signal or vector $\mathbf{x}$ is the number of nonzero elements in $\mathbf{x}$ and is given as:

$$||\mathbf{x}||_0 = \#\{i : x_i \neq 0\}$$

A signal or a vector is sparse if many of its elements are zero. That is, a signal $\mathbf{x}$ is sparse if $||\mathbf{x}||_0 << m$ where m is the dimension of the signal. If we substitute the function $G(\mathbf{x})$ with L_0 norm of x, i.e., $G(\mathbf{x}) \equiv ||\mathbf{x}||_0$, then the general minimization problem in Eq. (2.2.3.1) becomes:

$$\text{Problem P}_0\text{:} \qquad \min_{\mathbf{x}} ||\mathbf{x}||_0 \;\; \text{subject to: } \mathbf{b} = \mathbf{A}\mathbf{x} \tag{2.2.3.4}$$

While the solution to Problem P_2 is easy, the solution to Problem P_0 is, in general, NP-hard. In many applications, instead of solving the P_0 Problem, an approximate solution is obtained by minimizing the L_1 norm: $||\mathbf{x}||_1 = \sum |x_i|$ [112, 111, 113]. In this case the Problem P_1 is considered:

$$\text{Problem P}_1\text{:} \qquad \min_{\mathbf{x}} ||\mathbf{x}||_1 \;\; \text{subject to: } \mathbf{b} = \mathbf{A}\mathbf{x} \tag{2.2.3.5}$$

This is a convex optimization problem and among convex problems it is in some sense closest one to Problem P_0. For matrices $\mathbf{A}$ with incoherent columns, whenever Problem P_0 has a sufficiently sparse solution, that solution is unique and is equal to the solution of Problem P_1 [62]. Since Problem P_1 is convex, the solution can thus be obtained by standard optimization tools, such as linear programming. Also there are various well-established greedy algorithms that find the sparsest solution to Problem P_0. Donoho and Huo [112] have shown that in case $\mathbf{b}$ has a "very" sparse representation, i.e., when there exists $\mathbf{x}$ so that $\mathbf{b} = \mathbf{A}\ x$ and $||\mathbf{x}||_0 \ll n$, where n is dimensions of the vector $\mathbf{x}$, then this sparse representation is the unique solution of not only P_0 Problem but also of the P_1 Problem.

Sparse Dictionaries

Representing a signal involves the choice of a dictionary, the set of elementary signals called atoms. A signal can be decomposed into linear combination of these atoms.

When the dictionary forms a basis, such a linear combination is unique and can be expressed as:

$$\mathbf{x} = \Phi\alpha \tag{2.2.3.6}$$

where $\mathbf{x}$ is a signal or an image, Φ is the dictionary, and α are the coefficients or the linear combination representation.

Dictionary learning can be built by a) analytic modeling, b) data-driven or c) hybrid approaches [63].

a) Analytic dictionaries model a signal by using a class of mathematical functions. For example, the Fourier dictionary is designed around smooth functions, while the Wavelet dictionary is designed around piecewise-smooth functions with point singularities. Such dictionaries are formulated analytically and therefore enjoy a fast implicit implementation which does not involve multiplication by the dictionary matrix. On the other hand, the dictionary can only be as successful as its underlying model, and indeed, these models tend to be over-simplistic compared to the complexity of natural phenomena.

b) Trained dictionaries utilize the learning approach. The structure of the signal is extracted directly from the data rather than by using a mathematical function. This leads to finer adaptation to specific instances of the data. A key contribution to the area of dictionary learning was provided by Olshausen and Field [106, 107, 108]. They proposed to train a dictionary from small image patches collected from a number of natural images. The use of such a trained dictionary possesses a higher degree of statistical independence among its output scans, and therefore can produce a sparse and more efficient representation for the full set that spans the image space.

c) Hybrid dictionaries learn the sparse dictionary using knowledge of dictionary's specific structure [109]. Such dictionaries often benefits of both analytic and trained dictionaries

Note that dictionary training is a much more recent approach to dictionary design, and has been strongly influenced by the latest advances in sparse representations theory and algorithms. The main advantage of trained dictionaries is that they lead to state-of-the-art results in many practical signal processing applications.

Sparsity based methods are based on the assumption that the foreground image is sparse and the background can be learned from a dictionary [13, 14, 15, 20]. Authors in [13] proposed a general classification algorithm for image-based object recognition. They argued that a test sample can be represented as a sparse linear combination of training samples:

$$\mathbf{y} = a_1\mathbf{x}_1 + a_2\mathbf{x}_2 + \cdots + a_n\mathbf{x}_n, \tag{2.2.3.7}$$

where $a_j \in \mathcal{R}, j = 1, 2, ...n$. Then the linear representation of $\mathbf{y}$ can be rewritten in terms of all training samples as

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{2.2.3.8}$$

The above system is typically under-determined and so its solution is not unique. The sparsest solution to Eq. (2.2.3.8) can be obtained by solving the following optimization problem:

$$\widehat{\mathbf{x}}_0 = arg\min_{\mathbf{x}} ||\mathbf{x}||_0 \quad \text{s.t} \quad \mathbf{y} = \mathbf{A}\mathbf{x} \tag{2.2.3.9}$$

The problem of finding the sparsest solution of an under-determined system of linear equations is NP-hard. However, for $\mathbf{x}$ sparse, the solution of the L_0 minimization problem is equal to the solution of the following L_1 minimization problem:

$$\hat{\mathbf{x}}_1 = arg\min_{x} ||\mathbf{x}||_1 \quad \text{s.t} \quad \mathbf{y} = \mathbf{A}\mathbf{x} \tag{2.2.3.10}$$

The sparse modeling and dictionary learning approach proposed in [13] has been successfully applied to classification and detection problems.

Authors in [20] proposed a robust learning approach which simultaneously detects foreground pixels as outliers and builds a correct background model at the learning stage. Their approach appears to produce a better dictionary that the more traditional K-SVD algorithm does [21]. The method takes into consideration two sparsity assumptions, the background has a sparse linear representation with respect to a learned dictionary, and that the foreground is sparse since most of the pixels belong to the background. The background in the frames is a low-dimensional structure and therefore can be expressed by a linear combination of basis vectors. From the above

assumption, $\mathbf{x_B}$ has a sparse representation over a dictionary $\mathcal{D}$ as:

$$x_B = \mathcal{D}\alpha \tag{2.2.3.11}$$

Each frame $\mathbf{x}$ can be decomposed as a sum of two components: the background $\mathbf{x}_B$ and the foreground $\mathbf{x}_F$, i.e.,

$$\mathbf{x} = \mathbf{x}_B + \mathbf{x}_F \tag{2.2.3.12}$$

From Eq. (2.2.3.12), $\mathbf{x}_F = \mathbf{x} - \mathbf{x}_B$ has a sparse representation as well. The method aims to find α with the most sparsity by solving the L_0 minimization problem.

$$\widehat{\alpha} = arg\min_{\alpha} ||x - \mathcal{D}\alpha||_0 + \lambda||\alpha||_0 \tag{2.2.3.13}$$

Since the problem (2.2.3.13) is NP hard, authors solved the corresponding L_1 minimization problem (2.2.3.14) instead

$$\widehat{\alpha} = arg\min_{\alpha} ||x - \mathcal{D}\alpha||_1 + \lambda||\alpha||_1 \tag{2.2.3.14}$$

Designing an appropriate dictionary $\mathcal{D}$ is very important to the success of this method. Therefore, the dictionary is learned from training samples by collecting a few representative background frames as training samples, and then find an optimal dictionary $\mathcal{D}$ satisfying the following:

$$\mathcal{D} = arg\min_{\mathcal{D},\alpha} ||\mathbf{X} - \mathcal{D}A||_1 + \lambda||A||_1 \tag{2.2.3.15}$$

where $\mathbf{X}$ is the matrix of training samples, and $\mathbf{A}$ is the matrix of coefficient vectors.

Authors in [15] applied the knowledge of dictionary learning techniques to determine changes between a pair of images of the same scene taken at different times. Their method, **S**parsity-driven joint **I**mage **RE**gistration and **C**hange **D**etection (SIRE-CD), constructs the dictionary from training samples taken from one of the images, then find the sparsest representation that best approximates the second image. One of the assumptions is that both images must be very sparse in some domain. They

successfully applied the SIRE-CD to SAR images that are very sparse spatially. Images are registered at first, then one of the images is divided into many overlapping patches of the same size, which then are stacked as a column vectors, $\mathbf{y}$ and can be represented as a sparse linear combination over a dictionary Φ.

$$\mathbf{y} = \Phi\mathbf{x} \tag{2.2.3.16}$$

They find the best sparse approximation of $\mathbf{y}$ by solving the under-determined problem for the sparsest vector $\widehat{\mathbf{x}}$

$$\widehat{\mathbf{x}} = arg\min_{\mathbf{x}} ||\mathbf{x}||_0 \quad \text{s.t} \quad \mathbf{y} = \Phi\mathbf{x} \tag{2.2.3.17}$$

A subspace pursuit algorithm has been used to solve Eq. (2.2.3.17).

Many background modeling methods utilize knowledge of both sparsity and compressive sensing [10]. Sparsity is a key concept of compressive sensing framework and compressive sensing exploits the fact that many signals are sparse in the sense that they have concise representations when expressed in the proper basis [110].

2.2.4 Compressed Sensing Based Methods

Compressive sensing theory (CS) was introduced by Candes et al [5, 6, 61]. Using compressive sensing theory one can recover certain signals and images from far fewer samples or measurements than needed by traditional methods. Compressive sampling relies on two concepts: sparsity that pertains to the signals of interest and incoherence that pertains to the sensing modality.

- **Sparsity**: Compressive sampling exploits the fact that many natural signals are sparse or compressible. That is, that they have concise representations when expressed in the proper basis Ψ [110].

- **Incoherence** extends the duality between time and frequency and expresses the idea that objects having a sparse representation in Ψ must be spread out in the domain in which they are acquired, similar to a Dirac or a spike in the

time domain is spread out in the frequency domain [110]. In other words, a low coherence value indicates that a signal which is sparse in one of the bases will have a dense representation in the other one.

Candes, Romberg and Tao [58, 59, 60] showed that sparse representations can be recovered exactly from redundant dictionaries thanks to uncertainty principles. They proved that, in general, undetermined systems can be inverted if the solution is sparse: Given $\mathbf{y} = \mathbf{\Phi}\ \mathbf{x}$ and $\mathbf{x}$ is sparse, where $\mathbf{\Phi}$ is a matrix of size $m \times n$ $(m < n)$, we say that $\mathbf{x}$ is S-sparse if S of its components are non-zero.
According to [58], if the following inequality (2.2.4.1) holds,

$$(1-\delta)||\mathbf{x}||_2^2 \leq ||\Phi\mathbf{x}||_2^2 \leq (1+\delta)||\mathbf{x}||_2^2 \tag{2.2.4.1}$$

then we say that matrix $\mathbf{\Phi}$ satisfies the restricted isometry property (RIP) for sets of size 2S columns. It is also called the "2S-RIP". It is clear that we need $m > 2S$ rows in Φ to have the 2S-RIP.
Theorem (Candes, Romberg): A $m \times n$ sub Gaussian matrix (entries are i.i.d random variables) obeys the 2S-RIP for $m \geq C \cdot S\log(n/S)$ with high probability. This means that for $\mathbf{y} = \mathbf{\Phi}\ \mathbf{x}$ with $\mathbf{\Phi}$ Gaussian matrix and $\mathbf{x}$ a S-sparse signal then the signal $\mathbf{x}$ can be recovered from $\simeq S \cdot \log(n/S)$ Gaussian measurements. Given measurement $\mathbf{y}$, $\mathbf{x}_0$ can be recovered by solving:

$$\min_{\mathbf{x}} ||\mathbf{x}||_0 \ \text{ subject to: } \mathbf{y} = \Phi\mathbf{x} \tag{2.2.4.2}$$

Eq. (2.2.4.2) is hard to solve. However, if $\mathbf{\Phi}$ obeys the 2S-RIP, we can recover an S-sparse $\mathbf{x}_0$ from $\mathbf{y} = \Phi\mathbf{x}_0$ by solving Eq. (2.2.4.3):

$$\min_{\mathbf{x}} ||\mathbf{x}||_1 \ \text{ subject to: } \mathbf{b} = \Phi\mathbf{x} \tag{2.2.4.3}$$

Not only Ł$_1$ minimization perfectly recovers the signals that are exactly sparse, but it also recovers good approximations to signals that are approximately sparse. Robust recovery in presence of noise is also possible.

CS theory is recently used in background modeling methods. In applications of CS theory for background modeling, the foreground objects can be recovered by learning a sparse representation of the background image in some other domains.

The method in [10] recovered background subtracted images in video surveillance with the use of CS theory. The background subtraction problem can be seen as a sparse signal recovery problem where L_1 minimization can be applied. They utilize the sparsity of the foreground to perform background subtraction by using fewer measurements than the conventional methods.

If vector $\mathbf{y}$ of size $M \times 1$ represents the compressive samples and $\mathbf{\Phi}$ of size $M \times N$ is the measurement matrix, based on CS theory, $\mathbf{y}$ can be expressed as:

$$\mathbf{y} = \Phi\mathbf{x} = \Phi\Psi\alpha \tag{2.2.4.4}$$

where Ψ is the sparsity basis that provides a S-sparse representation of $\mathbf{x}$, and α is an $N \times 1$ column vector with K-nonzero elements. α can be recovered by solving the L_1 minimization problem [5, 6]:

$$\widehat{\alpha} = arg\min ||\alpha||_1 \quad \text{s.t} \quad \mathbf{y} = \Phi\Psi\alpha \tag{2.2.4.5}$$

Let $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_{dif}$ be the background, test and their difference image respectively. Note that $\mathbf{x}_{dif} = \mathbf{x}_2 - \mathbf{x}_1$ is sparse in the spatial domain. We have $\mathbf{y}_1 = \Phi\ \mathbf{x}_1$, $\mathbf{y}_2 = \Phi\ \mathbf{x}_2$. Samples $\mathbf{y}_{dif} = \mathbf{y}_2 - \mathbf{y}_1$ can be used to recover the foreground objects. Authors in [5, 6] reconstructed the background, test and difference images using L1TV minimization.

2.2.5 Robust PCA Methods

Robust PCA has applications in many other areas, such as video surveillance, face recognition, etc. In [19], the use of principle component pursuit (PCP) method was proposed to detect foreground changes in video surveillance. They assumed that each image matrix $\mathbf{M}$ can be written as a sum of a low-rank matrix $\mathbf{L}$ and a sparse matrix $\mathbf{S}$, i.e., $\mathbf{M} = \mathbf{L} + \mathbf{S}$. Video frames are stacked as columns of a matrix M, then the low-rank component $\mathbf{L}$ corresponds to the stationary background and the sparse component $\mathbf{S}$

captures the moving objects in the foreground. A simple principal component pursuit (PCP) solution recovers the low-rank and the sparse components, provided that the rank of the low-rank component is not too large, and that the sparse component is reasonably sparse:

$$\begin{aligned} &\text{minimize } ||\mathbf{L}||_* + \lambda||\mathbf{S}||_1 \\ &\text{subject to:} \quad \mathbf{L} + \mathbf{S} = \mathbf{M} \end{aligned} \tag{2.2.5.1}$$

It is worth emphasizing that in both sparsity and CS based background modeling, the solution involves the use of various L_1 minimization algorithms. Also, recently, many MRI reconstruction techniques employ compressive sensing methods. Work of [5, 6] is very well known among others for a direct application to MR images of the brain. They use the well known fact that MR images are sparse on some domains such as wavelet, finite differences, etc. Recovery of undersampled MR image using L_1 minimization allows for rapid MR imaging.

Work in [22] applied dictionary learning techniques to solve the reconstruction problem. They proposed an adaptive patch-based sparsifying dictionary learning in the k-space (i.e., the Fourier frequency domain) and used it to remove aliasing and noise. The dictionary is created using the K-SVD algorithm and is updated for each image block. In most recent research, the CS sparse models, and dictionary learning algorithms are used often for classification and detection problems.

2.3 Change Detection in Medical Images

Some of the methods described in the previous section have also been used for detecting changes of medical images. In this section we review several methods or change detection systems that have been used in MR imaging during the last decade. The authors in [27] presented a detailed review of change detection methods in serial imaging studies of the brain, and classified them in manual inspection, measurement sampling, volumetric, warping, and temporal analysis methods.

Measurement sampling methods include one-diameter measure and two-diameter

measure. The one-diameter measure method [48] measures the maximum diameter and the maximum corresponding perpendicular of a tumor to assess the progression or regression of the disease. In the two-diameter approach [49] the radiologists compute the maximum area which is used to rate each patient's progression or regression of the disease. One of the drawbacks of such a method is the ability to detect small changes, because of the difficulty of the radiologists to reach a consensus.

In some of volumetric methods, the global volume of each anatomical tissue of interest at each acquisition time is calculated, and the results are subtracted [50, 51]. This increases the uncertainty and error is additive when regions are large. Two other volumetric methods [52, 53] identify the tumor first and then compute its volume each time. The growth rate is computed as a ratio of the difference between the computed volumes and time interval.

Some nonlinear linear registration methods (warping) [54] utilize an elastic image registration algorithm to match surfaces between acquisitions and to compute the volumetric deformation fields such as dilation and contraction. One of the problems with the nonlinear registration is that it is not a one-to-one tissue correspondence based on intensity, since there is an infinite number of displacement fields corresponding to a given pair of acquisitions.

Temporal analysis methods [55, 56] are based on computing various metrics for each voxel. Based on the computed metric, these methods calculate the probability of a voxel belonging to the change over time series. By analyzing these methods and pointing out their drawbacks, authors in [3, 7] made the case that automated change detection systems would be a great help for the radiologists to correctly interpret data.

The work of [57] provides a review of the methods and algorithms for detection of multiple sclerosis lesions (MS) in serial MR images of the brain. They also make the case for an automated change detection system. Visual inspection is still the most common approach in change detection. According to [57], methods for detecting MS are classified into two main categories: intensity based methods, and deformation based approaches. Each of these categories is sub-classified into sub-categories, as shown in Fig 2.3.1.

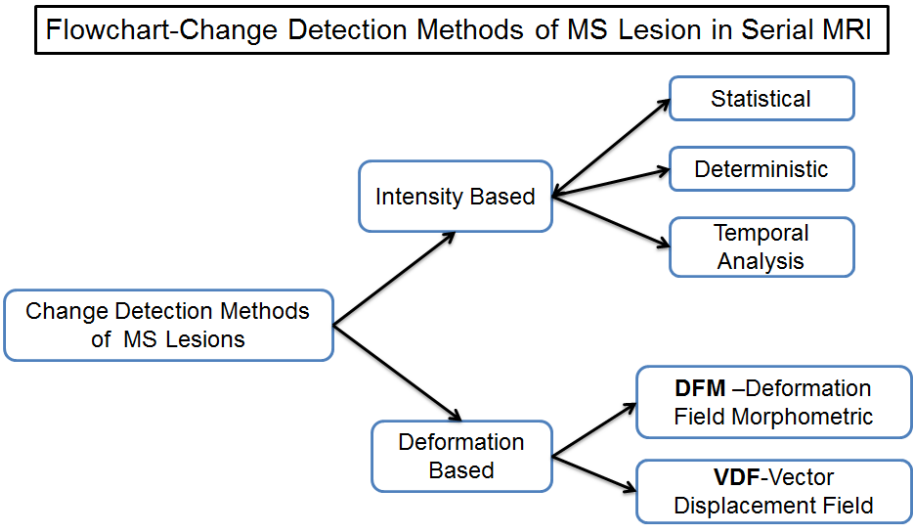

Figure 2.3.1: A flowchart showing the classification of change detection methods in in serial MR brain scans of multiple sclerosis patients, according to Lladò et al.

2.3.1 Intensity based approaches

Intensity based approaches are based on comparison between the baseline's voxels intensities and its corresponding follow ups. Patient's position change and movements lead to errors which make the registration a very important pre-processing step. One possible way to reduce this miss-alignment caused by the lesion evolution is to use a similarity metric robust to local differences. The group of intensity based methods is sub-classified into: statistical, deterministic and temporal change methods [57].

2.3.1.1 Statistical change detection techniques

Statistical change detection techniques are an alternative to simple differencing. Some of statistical methods [2, 68] deal with multi-modal image data, using multidimensional Gaussian models of image intensities. False detection rate is lower with respect to standard single-modal methods. The work in [68] presents an automated system to detect subtle changes in mixed-signal MR brain lesions by scan matching and spatial normalization. "Match scan" represents the scan to be registered to the "base scan", a scan chosen as the registration target. Scan matching or co-registration is performed by multi-scalar iterative searches through six parameters (three for translations and three for rotations) at the multi-resolution levels which are done by Gaussian blurring

of images. This is followed by global intensity matching, subtraction of the images, and filtering out the difference image. Finally, in order to detect active lesions and their evolutions, a statistical model of intensity changes between successive scans is built, and the quantification of changes is performed by structured difference filtering based on the statistical likelihood of "structure" in a Gaussian field, followed by thresholding the structure differences. The methods rely on the changes of the lesions not on the changes of the individual voxels.

Registration is a very important component of medical image analysis. Registered images are easier to compare as only differences between scans can be evaluated to further facilitate the detection of change. Errors are introduced during the registration process as the accuracy of registration techniques relies on the applied interpolation methods, motion artifacts, and image resolution [66]. Many treat the registration problem as a mathematical optimization, using a cost function to quantify the quality of the registration for some given transformations. A thorough review of how the choice of the cost function and the number of transformational parameters effect the quality of registration is discussed in [69, 70].

2.3.1.2 Deterministic approaches

Deterministic approaches are based on subtraction of two consecutive registered temporal volumes. Unchanged voxels may represent normal tissues or unchanged lessons, or the background, while changed areas may be due to appearance, disappearance, shrinkage, or enlarging lesions. Positive values of changes represent new or enlarging lesions, while negative values of changes are the result of shrinkage or disappearance.

Another example in the intensity-based methods proposed in [64] initially aligns two volume sets of data to a fraction of the linear dimension of a voxel, and then utilizes by simple differencing of the aligned source images to find intensity differences due to evolving lesions. The registration step involves finding the solution of the least

square error of the difference volume.

$$\min_{\alpha,\beta,\gamma,T_x,T_y,T_x} \frac{1}{N_{voxels}} \sum_{voxels} (I_2 - I_1)^2 \tag{2.3.1.1}$$

where $\alpha, \beta, \gamma, T_x, T_y, T_x$ are rotation angles and translations with three directions x, y, and z. The authors in [65] found that it is unlikely to have a perfect interobserver agreement when new lesions have only moderately altered signal intensity, and that the use of subtraction results in moderate interobserver agreement for enlarging lesions.

2.3.1.3 Temporal analysis

Temporal analysis is based on the analysis of series of MR images over time, with at least more than two examinations. Time series scans are used to monitor disease activity and progression and provide information about subtle morphological and functional changes [72, 73]. These methods are based on the fact that lesions found by segmentation methods are significantly correlated over time. By introducing the time as the fourth dimension, the segmentation of series of 3D data sets of the temporal change at each voxel is replaced with a 4D image set [72]. In temporal analysis, 3D MR data sets acquired at different time interval, are combined to 4D (x, y, z, t) data sets, allowing for the intensity and the brightness of each voxel to be considered as a function of time. The dynamic changes in lesion voxels are detected by analyzing the time series of each voxel over time. The method assumes that data sets are perfectly registered and normalized.

This is a big issue in MR images as the MR intensity is not absolute. The same tissue may have different image intensities at different scans. Both normalization and regististration are necessary steps. Preprocessing includes registration, normalization of image brightness, brain mask within brain white and gray matter by segmentation system [71], and sets of corrected 3D data are combined to form 4D data sets. Time series were analysed by developing features to describe variation of lesions. They validated their methods using simulated lesion pattern in 4D data sets.

Authors in [73] presented a method of time-series analysis (TSA) that performs

direct quantitative analysis of signal intensity in the time domain. Their method seeks the integration into a 4D spatio-temporal data set by spatial and intensity normalization, artifact filtering, fusion into a 4D volume, and analysis by creating parametric images and feature maps. They claim that correct 4D integration minimizes artifacts due to mis-registration, partial-volume effects, and scanner float.

2.3.2 Deformation field-based approaches

Deformation field-based approaches refer to the modification of the surrounding tissues, due to lesion expansion or contraction. A non-linear registration is performed between successive scans, and the structural changes are determined based on the local deformation of voxels. These methods are divided into deformation field morphometry methods and vector displacement fields [76, 77, 57].

2.3.2.1 Deformation field morphometry

Deformation field morphometry [76] calculates the deformation vectors related to the grid points between the images to indicate shifted voxels in the source image (a deformed image to target image). A local volume ratio is defined as the volume of the deformed voxels in the source image divided by the volume of the non-deformed voxels in the target image. A local volume ratio greater than one shows a local increase of an existing lesions and vice-versa, a local volume ratio less than one shows a local decrease [76].

2.3.2.2 Vector displacement fields

Vector displacement fields are a semi-automatic approach using vector displacement fields obtained by a non-rigid registration of two successive scans to track MS lesions. Both the divergence and the norm of the displacement vector fields are used in order to detect deformations and intensity changes. Therefore, high values of the norm indicate large deformation areas, while high divergence indicates evolving lesions, where the

sign of the divergence operator shows whether the lesion was growing or shrinking. As showed in [77], noise can be characterized by high divergence and low norm while the norm is large and the divergence low in the case of a translation.

2.3.3 Automated Change Detection System

An automatic change detection system for serial MRI with applications in MS - follow ups is presented in [2]. The method is based on the use of multi-modal information for change detection, generalized likelihood ratio test (GLRT), and nonlinear joint histogram normalization. This method does not work well when noise is not-stationary. Their change detection algorithm goes through four steps:

1. Brain extraction. In this step, the method extracts the head from the background noise using Otsu's thresholding method at first, and then the brain extraction is done using an atlas-based segmentation.

2. Repositioning and correction of deformation artifacts. In this step, an image is chosen randomly from the database as a reference image, and all other images of the dataset are aligned according to the reference image.

   The algorithm applies the iterative affine registration techniques. In each iteration an affine registration is followed by a re-computation of the brain extractions, which is computed as the intersection of the brains of the registered images at that step.

   Deformable registration is done by the deformable matching method which estimates a parametric displacement field decomposed on multiresolution B-spline basis functions. Then, image resampling is performed using interpolation techniques. A global geometric transform is computed by composing the transforms obtained during the affine and deformable registration steps. A 5-degree box spline kernel is used in the last deformable registration step [101].

3. Intensity normalization is achieved using a nonlinear normalization method which relies on the estimation of joint probability distributions of intensities in two

images;

$$j(u,v) = p\Big(\mathbf{I}_1(s) = u, \mathbf{I}_2(s) = v\Big) : s \in \Omega \tag{2.3.3.1}$$

After images are registered and their intensities are normalized, the obtained difference image is composed of noise, residual artifacts and the real image changes which are small in scale compared to the RF artifacts. The method applies a large low-pass Gaussian filter with a large standard derivation $\sigma = 15mm$ on the difference image, to isolate the RF artifacts.

4. A single or multimodal statistical change detection test is the last step. The generalized likelihood ratio test (GLRT) is used to compute the ratio of the probabilities of two hypotheses: the null hypothesis $\mathcal{H}_0 \rightarrow$ there is no change between $\mathbf{I}_1$ and $\mathbf{I}_2$ inside a window W, or $\mathcal{H}_1 \rightarrow$ there is significant change between $\mathbf{I}_1$ and $\mathbf{I}_1$ inside W.

   The generalized likelihood ratio is defined as:

$$R_{GLRT} = \frac{p(\mathbf{I}_1, \theta_1)p(\mathbf{I}_2, \theta_2)}{p(\mathbf{I}_1, \theta_0)p(\mathbf{I}_2, \theta_0)} \tag{2.3.3.2}$$

   For a single modality the average $\theta = \mu$, the maximum likelihood estimate of μ is $\widehat{\mu}$, where $\widehat{\mu} = \sum_{(i,j)\in W} \frac{\mathbf{I}(i,j)}{n}$ and n is the number of voxels in the window W. The logarithm of Eq. (2.3.3.2) is computed. The expression used to decide whether a voxel belongs to the change or no change is:

$$\begin{aligned} (\mathcal{H}_1 : \text{change}) &\Longleftrightarrow \tfrac{\sqrt{n}}{2\sigma}|\widehat{\mu}_2 - \widehat{\mu}_1| > \lambda \\ (\mathcal{H}_0 : \text{no change}) &\Longleftrightarrow \tfrac{\sqrt{n}}{2\sigma}|\widehat{\mu}_2 - \widehat{\mu}_1| < \lambda \end{aligned} \tag{2.3.3.3}$$

   where λ is a threshold

A similar approach is taken for the multimodality case. The probability of observing intensity values $\mathbf{I}(i,j)$, for $(i,j) \in W$, for a given mean μ, for m modalities is:

$$p(\mathbf{I}, \mu) = \left(\frac{1}{\sqrt{(2\pi)^m |C|}}\right)^m exp\left(-\frac{1}{2}\sum_{(i,j)\in W}(\mathbf{I}(i,j) - \mu)^T C^{-1}((\mathbf{I}(i,j) - \mu)\right) \tag{2.3.3.4}$$

Authors in [7, 8, 9] developed an automated system to examine the MR scans of brain tumors. The main purpose of these studies is to reduce human error and minimize the enormous amount of data that radiologists have to process to reach a conclusion. They found that implementing a scientifically useful tool is most clinically viable when it is efficiently integrated into clinical work-flow, and that automated change detection improves efficiency, accuracy and agreement. Therefore, it is very useful to have an efficient computer-based system that automatically reduces the quantity of data and directs radiologists' attention to clinically-relevant changes.

Their integrated system for detecting changes in serial multi-modal MRI examinationa or scans is based on post classification of image pixels in multi-spectral MR intensity feature space. The rational behind their method is based on the observation that an abnormal tissue is a tissue transitioning from one normal tissue to another in the feature space and an assumption that change tends to occur along lines connecting pairs of cluster centroids in the feature space. The system additionally formats the output as a quantitative summary. The change detection algorithm goes through the following steps:

1. **Pre-processing**: multi-contrast images, such as T1-weighted, FLAIR, or T1-weighted post Gd at least at two different times undergo four preprocessing steps. Inhomogeneity correction and registration were done with Insight Segmentation and Registration Toolkit (ITK) [102]. Registration of serial examination pairs, both within a particular exam and between serial exams, is important to establish a common spatial framework. In the third and the fourth pre-processing steps a brain mask is manually defined, separating the brain and cerebrospinal fluid (CSF) from the non-brain tissues. Samples of normal-appearing white matter (NAWM) are defined by using the image edit component of the "Analyze" software package.

2. Characterization of the contents of voxels based on image intensity (e.g., white matter, gray matter, or CSF). The algorithm locates sample points for CSF, normal gray matter, and also locates or synthetically creates sample points for three pathological "tissues". The algorithm makes assumptions that voxels don't

contain more than two different tissues, called a "tissue pair", and that changes between two different times occur between the pairs, forming duel tissue classes. For this algorithm, the dual tissue classes were: NAWM and NETTA, NAWM and enhancement, NETTA and enhancement, enhancement and necrosis, and NETTA and necrosis.

3. Feature extraction: A scatter-plot of the feature space and partial membership lines connecting the centroids of the contributing tissues in each of the relevant pairs is drawn in multispectral intensity space. They assume that the multispectral intensity of a voxel will follow such a line, thus all points in this feature space are projected onto this line, which results in the dimensionality reduction. The distance from a point P to its projection P_{proj} is calculated using Mahalanobis distance D. As shown in Fig 2.3.2, if the coordinates of any point P is known

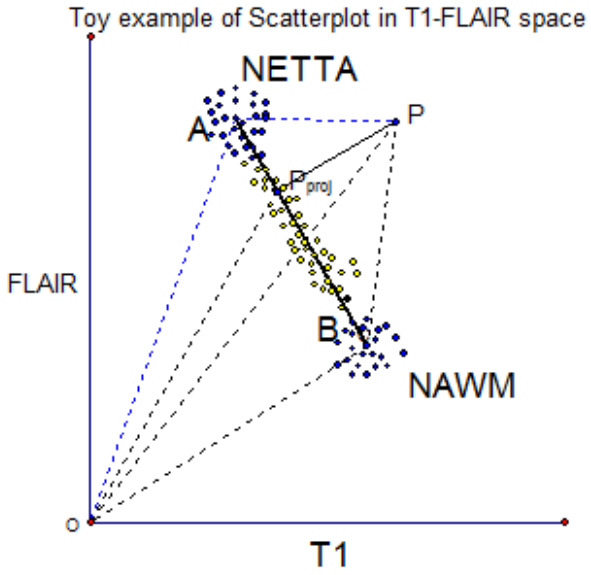

Figure 2.3.2: Points A and B are the centroid of NETTA and NAWM respectively. P_{proj} is the projection of point P onto the line segment joining centroids A and B.

.

then the coordinates of its projection P_{proj} can be found by Eq. (2.3.3.5).

$$|\overrightarrow{\mathbf{OP}}_{proj}| = \frac{(\overrightarrow{\mathbf{OB}} - \overrightarrow{\mathbf{OA}}) \bullet (\overrightarrow{\mathbf{OP}} - \overrightarrow{\mathbf{OA}})}{|\overrightarrow{\mathbf{AB}}|^2} = \frac{\overrightarrow{\mathbf{AB}} \bullet \overrightarrow{\mathbf{AP}}}{|\overrightarrow{\mathbf{AB}}|^2} \tag{2.3.3.5}$$

4. Membership assignment: each voxel is assigned to the closest dual tissue class. Let μ_A and μ_B represent the mean intensities of both tissues which make up the

partial membership line. Let x_P represents the intenisity of the voxel P. Then, V_A, and V_B represent the fractional volumes (or the membership) calculated using the following formulas:

$$V_A = \frac{x_P - \mu_A}{\mu_A - \mu_B}$$
$$V_B = \frac{x_P - \mu_B}{\mu_A - \mu_B}$$

5. The inverse of the square of Mahalanobis distance, $\frac{1}{D^2}$, is calculated to decide whether or not a point is a member of a given dual-tissue class. A point belongs to a given dual-class if Mahalanobis distance for that point is low.

6. This is the noise reduction step, which is done by thresholding the mean membership change. If the mean membership change exceeds the threshold then the the change is considered real, otherwise the change is ignored.

To validate and compare the performance of their algorithm the specificity and sensitivity measures on nine serial phantom pairs are computed. The values of specificities were in the range [0.8, 1] and sensitivities in [72, 96] with a p-value 0.008.

Preliminary clinical studies show that their system can identify visually subtle changes related to disease. However, tissue classification task itself is very difficult; in addition, the whole process of calculating transition tissue types and fractional membership for each pixel is inherently time-consuming.

Another change detection method for two MR images was introduced in [11]. The method is based on local steering kernels (LSK) and calculation of test statistics is derived from cosine similarity [47]. Their work does not address registration and alignment. They assume that images have undergone two preprocessing steps: geometric co-registration and the intensity adjustment.

For both the reference image $\mathbf{I}_1$ and the test image $\mathbf{I}_2$, the LSK are calculated as a measure of the local similarity of a pixel x_j to its neighbors within a window W of size $m \times m$, i.e.,

$$\mathcal{K}_{\mathbf{I}_s}(x_i - x_j) = \frac{\sqrt{det(\mathbf{\Omega}_i)}}{h^2} exp\left(\frac{(x_i - x_j)^T \mathbf{\Omega}_i (x_i - x_j)}{-2h^2}\right), \tag{2.3.3.6}$$

where $s \in \{1, 2\}$ and $i \in \{1, ..., m^2\}$, $\mathbf{\Omega}_i$ is the covariance matrix of spatial gradient vectors within W and around x_i, and h is a global smoothing operator.
Next, the LSK are normalized as:

$$\overline{\mathcal{K}}_{\mathbf{I}_s}(x_i - x_j) = \frac{\mathcal{K}_{\mathbf{I}_s}(x_i - x_j)}{\sum_{i=1}^{m^2} \mathcal{K}_{\mathbf{I}_s}(x_i - x_j)} = k^j_{\mathbf{I}_s}, \text{ for } s \in \{1, 2\} \tag{2.3.3.7}$$

For each pixel x_j, they will have m^2 of $k^j_{\mathbf{I}_s}$ which are stacked as column vectors. Next the cosine similarity is applied as the "distance measure" between pairs $(k^j_{\mathbf{I}_1}, k^j_{\mathbf{I}_2})$ as:

$$\rho(k^j_{\mathbf{I}_1}, k^j_{\mathbf{I}_2}) = cos\theta_j = \frac{(k^{j\ T}_{\mathbf{I}_1})(k^j_{\mathbf{I}_2})}{||k^j_{\mathbf{I}_1}||\ ||k^j_{\mathbf{I}_2}||} = \rho_j \tag{2.3.3.8}$$

Finally, the dissimilarity map (DM) is computed at each x_j as:

$$DM : f(\rho_j) = \frac{1 - \rho_j^2}{\rho_j^2} \tag{2.3.3.9}$$

The significance test is performed on the DM with a 99% confidence interval to detect regions with significant changes. Sensitivity, specificity and similarity index (SI) performance measures are computed to validate the method. They reported that their method achieved a sensitivity = 0.877, specificity = 0.998, and SI = 0.879.

2.4 Remaining Challenges

Change detection methods mentioned in Section 1 and Section 2 present an improvement over simple visual inspection and classical statistical approaches. However, most methods require many preprocessing steps such as registration before using the core algorithm [7, 8, 9]. Some methods require spatially sparse images [15] while the others require a large database of images [13, 14]. To the best of our knowledge there are no well known change detection methods that would work without such limitations. For this reason we have looked at mathematical frameworks that take into consideration the information obtained from measurements themselves. These approaches include dictionary learning techniques and subspace learning which will be discussed in our

contributions in Sections 3 through 8.

Chapter 3

The Adaptive EigenBlock Dictionary Learning Algorithm

3.1 An Introduction

Our approach to change detection problem is based on background modeling and background subtraction. Background subtraction is the separation of the moving object called the "foreground" from the static information called the "background" [105]. Background subtraction is a particular case of change detection when the reference image is the background and the second is the test image. Changes can be revealed by subtracting the image background from the test image. Background modeling describes the kind of model used to represent the background [105].

Most images are sparse in either spatial domain or some transformed domain. In particular, the background subtracted image is generally sparse spatially. With recent developments in the area of compressive sensing (CS) [5, 6], and availability of various sparse image reconstruction algorithms, it is natural to explore the use of CS in change detection as in [10, 15]. Any application of CS in background subtraction models involves the use of various L_1 minimization algorithms.

We first explored the possibility of utilizing image sparsity and compressive sensing to automatically detect changes between two consecutive images by designing the **A**daptive **E**igenBlock **D**ictionary **L**earning (AEDL) algorithm [23]. The AEDL algorithm captures local spatial differences between two or more consecutive images and

detects the significant changes between them. Changes are linearly modeled by a local dictionary trained from the images themselves and reconstructed locally by L_1 minimization. The AEDL algorithm is designed to ignore insignificant changes due to mis-alignment (spatial shift, rotation), field inhomogeneity and noise. We use the principle component analysis (PCA) to reduce the size of local dictionaries in the algorithm and to identify the linear relationship in the data. This helps to speed up the computations for practical applications. Performance of our algorithm is validated using synthetic and real images.

To simplify the problem, we focus on detecting differences between two consecutive images of the same anatomical location taken at times t_1 and t_2. The image corresponding to t_1 is referred to as the reference image, and similarly, the image corresponding to t_2 as the test image. Note that these two images are not necessarily aligned and may contain different noise levels, intensity inhomogeneity, and other acquisitions related artifacts. Roughly speaking, the AEDL algorithm consists of three stages as shown in Fig. 3.1.1: 1) Capture local spatial changes between the reference and test images by averaging all the absolute differences of test image blocks with their best linear approximations given by a local dictionary trained from the reference image. The local background image, modeled as a sparse linear combination of the image blocks in the dictionary, can be reconstructed by an L_1-minimization algorithm. 2) Repeat Stage one by averaging all the absolute differences of reference image blocks with their best linear approximations given by a local dictionary trained from the test image. 3) Finally, the change image is computed as the average of change images from Stages one and two.

The AEDL algorithm extends previous works of [1, 13, 14, 15, 16]. It is designed for medical images to automatically detect the anatomic changes in the test image. The AEDL algorithm is different from [1, 13, 14, 15, 16] in the following aspects:

i. We focus on change detection in medical images where we deal with mis-alignment at different scans and imaging acquisition related artifacts.

ii. We use only two consecutive MR images, hence our dictionary is not learned from a database of images. It is built on one image and trained from local blocks of

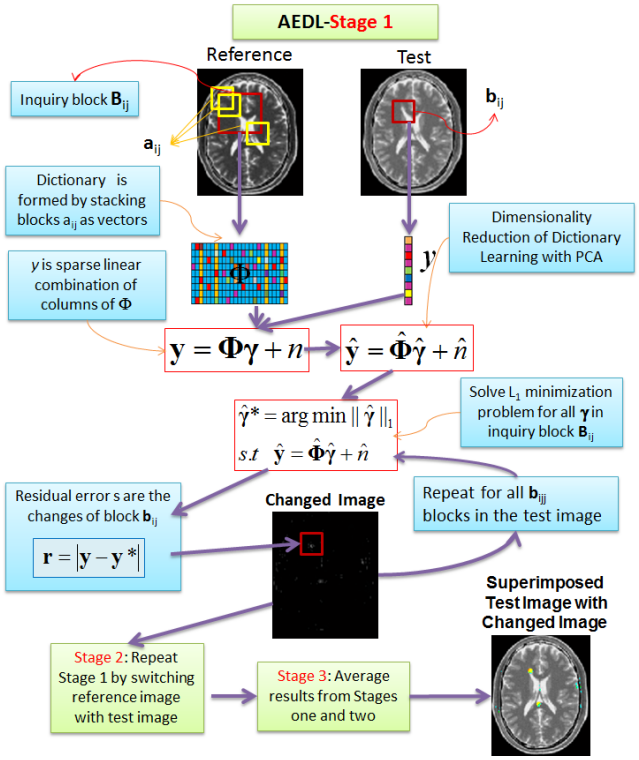

Figure 3.1.1: Three stages of the AEDL algorithm

that image.

iii. The size of the local dictionary is further reduced using PCA to improve computational efficiency.

3.2 The AEDL Algorithm With L_1 Minimization

Let $\mathbf{I}_1$ and $\mathbf{I}_2$ represent two images of size $N_1 \times N_2$ of the same anatomic structure at two different times, t_1 and t_2 respectively. $\mathbf{I}_1$ is referred as the reference image and $\mathbf{I}_2$ as the test image. $\mathrm{I}_s(i, j)$ is image intensity of the pixel at the i-th row and the j-th column, for $s = 1, 2$.

We assume that the differences of these two images include disease related changes that have occurred from time t_1 to time t_2 and also changes caused by registration shifts, rotations, noise, and other acquisition related artifacts. During this time interval most blocks from the reference images have undergone a few disease related changes. Most pixels of the image at time t_1 will appear again in the second image $\mathbf{I}_2$ either at the same or a nearby location. This means that a block from the reference image, with some structure, would appear again in the test image, having a similar

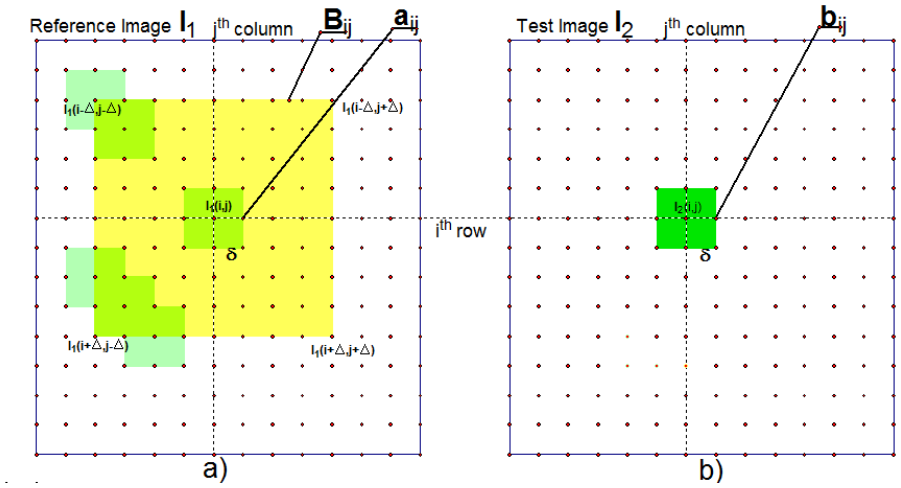

Figure 3.2.1: a) An example of an inquiring block $\mathbf{B}_{ij}$ in orange of size 9×9 from the reference image $\mathbf{I}_1$ corresponding to b) the test block $\mathbf{b}_{ij}$ from the test image $\mathbf{I}_2$ centered at i^{th} row and j^{th} column. Here δ=1 and $\Delta = 4$. There are 81 overlapping $\mathbf{a}_{ij}$ blocks of size 3×3, examples of them shown in light green in a), which form the local dictionary corresponding to the test block $\mathbf{b}_{ij}$.

structure, either at the same location or within the neighbourhood. In other words, the background of each block from $\mathbf{I}_2$ image can be learned by $\mathbf{I}_1$ image blocks. Therefore, the background of a block from the test image, located at (i, j), can be learned from the reference image blocks located in some Δ neighborhood of (i, j).

Let $\mathbf{b}_{ij}$ be a block of size $(2\delta+1)\times(2\delta+1)$ in the test image centered at pixel (i, j), where parameter δ is a positive integer, representing the radius of the $\mathbf{b}_{ij}$ block. There are $(2\delta+1)\times(2\delta+1)$ pixels in $\mathbf{b}_{ij}$. Let $\mathbf{a}_{ij}$ be a block in the reference image of the same size as $\mathbf{b}_{ij}$, centered at the same location. It follows from our assumptions that for a test block $\mathbf{b}_{ij}$, we can define a neighborhood of length Δ and centered at position (i, j) in the reference image, such that the background of $\mathbf{b}_{i,j}$ can be learned from the dictionary blocks within the neighborhood Δ. Let $\mathbf{B}_{ij}$ be the inquiring block corresponding to the test block $\mathbf{b}_{i,j}$, which is given by the sub-image defined on the neighborhood Δ in the reference image. The background of $\mathbf{b}_{i,j}$ can be mathematically expressed by blocks extracted from the block $\mathbf{B}_{ij}$ in the reference image centered at (i, j). Fig. 3.2.1 shows an example of a block $\mathbf{b}_{ij}$ in the test image of size $(2\delta+1)\times(2\delta+1)$ with $\delta = 1$, its corresponding overlapping training blocks $\mathbf{a}_{ij}$ in a neighbourhood Δ of (i, j) in the reference image with $\Delta = 4$.

Basic Conventions and Notations:

To simplify the notation, we drop the subindex ij and we set the following notations for the rest of this section:

- $\mathbf{I}_1$ and $\mathbf{I}_2$ images corresponding to time points t_1 and t_2 are referred as reference and test image, respectively.
- The block $\mathbf{b}_{ij}$ from test image, located at pixel (i, j), is named as the **Block of Interest**, and is denoted by $\mathbf{b}$.
- The Δ neighbourhood of (i, j) pixel in reference image, $\mathbf{B}$, is called the "**Inquiry Block**", and is given by:

$$\mathbf{B} = \left\{ \mathbf{I}_1(\tilde{i}, \tilde{j}) \colon \tilde{i} \in [i - \Delta, i + \Delta], \tilde{j} \in [j - \Delta, j + \Delta] \right\}$$

- $\mathcal{D}$ is the set of all the blocks $\mathbf{a}_k$ of the same size as the block of interest $\mathbf{b}$, from inquiry block $\mathbf{B}$.

$$\mathcal{D} = \left\{ \mathbf{a}_k \colon k = 1, 2, ...(2\Delta + 1)^2 \right\}$$

- Blocks $\mathbf{a}_k$ from the inquiry block are called the "**Training Blocks**". Blocks $\mathbf{a}_k$ are of sizes $(2\delta+1) \times (2\delta+1)$ centered at (p, q), where $k = (2\Delta+1) \times (q-1)+p$.
- Let $m = (2\delta + 1)^2$ and $n = (2\Delta + 1)^2$. Then, the block of interest $\mathbf{b}$ and every training block $\mathbf{a}_k$ are of size $\sqrt{m} \times \sqrt{m}$. Similarly, the inquiry block $\mathbf{B}$ is of size $\sqrt{n} \times \sqrt{n}$.

From our assumptions, the block of interest is learned from training blocks in the inquiry block. In our model we express $\mathbf{b}$ as a linear combination of blocks $\mathbf{a}_k$:

$$\mathbf{b} = \sum_{k=1}^{(2\Delta+1)^2} \boldsymbol{\gamma}_k \mathbf{a}_k = \sum_{k=1}^{n} \boldsymbol{\gamma}_k \mathbf{a}_k, \quad \mathbf{a}_k \in \mathcal{D} \tag{3.2.0.1}$$

where $k \in \{1, 2, \ldots, n\}$

Training blocks $\mathbf{a}_k$ and the block of interest $\mathbf{b}$ are stacked as $m \times 1$ column vectors $\mathbf{x}_k$ and $\mathbf{y}$ respectively, where $m = (2\delta + 1)^2$ and δ is a positive integer representing the

radius of these blocks. Then, Eq. (3.2.0.1) can be written as:

$$\mathbf{y} = \sum_{k=1}^{n} \gamma_k \mathbf{x}_k, \quad k \in \{1, 2, \ldots, n\} \tag{3.2.0.2}$$

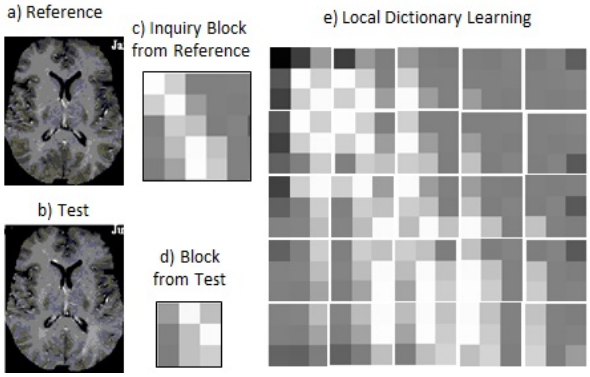

Figure 3.2.2: a) Reference and b) test images. c) Inquiry block **B** from reference image of size 5×5 ($\delta = 1, \Delta = 5$). d) The corresponding block of interest **b** of size 3×3 from test image, centered at 64^{th} row and 64^{th} column. e) Dictionary Φ of size 9×25, that is 25 blocks of size 3×3.

Let $\Phi = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k, \ldots, \mathbf{x}_n]$ be a matrix of all the training blocks $\mathbf{a}_k$ stacked as column vectors $\mathbf{x}_k$. We call matrix Φ the dictionary.

Remark 1. The dictionary Φ is used in our model to learn the background of the block of interest. Remark 2. The dictionary Φ is adaptive in a sense that the columns of Φ get updated every time we move to a different block of interest in the test image.

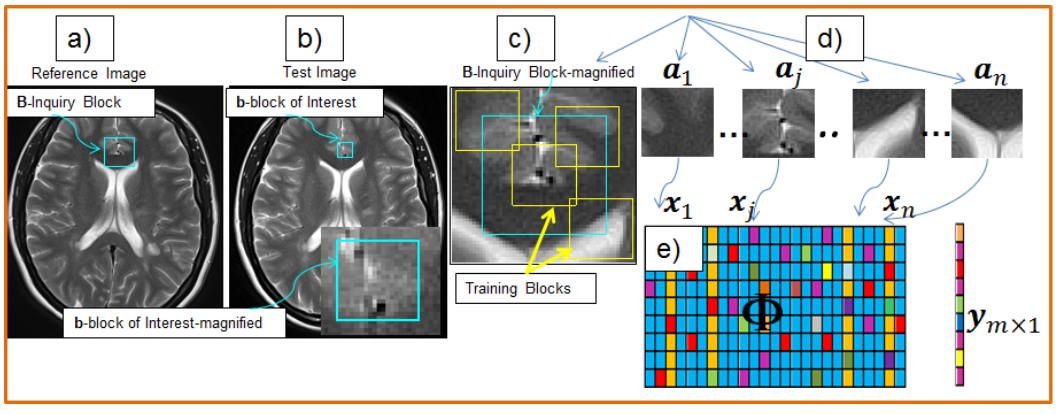

Figure 3.2.3: a) Reference image with an inquiry block **B**. b) Test image with a block of interest **b**. c) Magnified inquiry block **B** with many overlapping training blocks. d) Training blocks, $\mathbf{a}_k for k = 1, 2, ..., n$ from the inquiry block. e) Dictionary Φ formed by stacking training blocks as column vectors and vector **y** representing the block of interest.

Fig. 3.2.2 shows an example of how the dictionary Φ is created for a given block of interest $\mathbf{b}$ with $\delta = 3$, and $\Delta = 5$. Also, Fig. 3.2.3 illustrates how the block of interest $\mathbf{b}$ is extracted from the test image and stacked as a column vector $\mathbf{y}$, and how the training blocks are extracted from the inquiry block and stacked as column vectors to form the dictionary Φ. Then Eq. (3.2.0.2) can be written as:

$$\mathbf{y} = \Phi\boldsymbol{\gamma} \tag{3.2.0.3}$$

where:

$$\Phi = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k, \ldots, \mathbf{x}_n]$$

$\mathbf{x}_k \in \mathbb{R}^{m\times 1}, k = 1, 2, ...n$, obtained by stacking the entries of the training block $\mathbf{a}_k$, i.e., $\mathbf{x}_k = [x_{1k}, x_{2k}, \cdots, x_{mk}]^T$, $\mathbf{y} \in \mathbb{R}^{m\times 1}$ is the vector representation of the stacked block of interest, i.e., $\mathbf{y} = [y_1, y_2, \cdots, y_m]^T$ and $\boldsymbol{\gamma} \in \mathbb{R}^{n\times 1}$ is the vector of coefficients in Eq. (3.2.0.3), i.e., $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \cdots, \gamma_n]^T$.

Dictionary Φ is of size $m \times n$ where $m = (2\delta + 1)^2$ and $n = (2\Delta + 1)^2$. Columns of Φ represent training blocks, and rows are the number of pixels in each training block. Therefore, dictionary Φ can be interpreted as a 2-dimensional matrix with n samples (observations represented by training samples) in m dimensions. Finding $\boldsymbol{\gamma}$ involves solving an underdetermined set of m equations with n unknowns, and hence must be done subject to additional requirements on the solution.

We made the assumptions that there are only a few changes occurring from time t_1 to time t_2 and that all the training blocks are in a Δ neighbourhood of (i, j). It follows that the block of interest $\mathbf{b}$ can be sparsely represented as a linear combination of a few blocks $\mathbf{a}_k$ from the inquiry block $\mathbf{B}$ in the reference image for some small values of Δ. That is, $\mathbf{y}$ can be represented as a sparse linear combination of columns of dictionary Φ, which means that the vector of coefficients $\boldsymbol{\gamma}$ is sparse as Fig. 3.2.4 illustrates. The additional requirement is that $\boldsymbol{\gamma}$ should be sparse.

The problem then becomes finding the sparsest $\boldsymbol{\gamma}$ such that Eq. (3.2.0.3) holds.

Most existing methods such as those in image surveillance and face recognition [1, 13, 14, 15, 16] learn the dictionary by using training samples from a database of

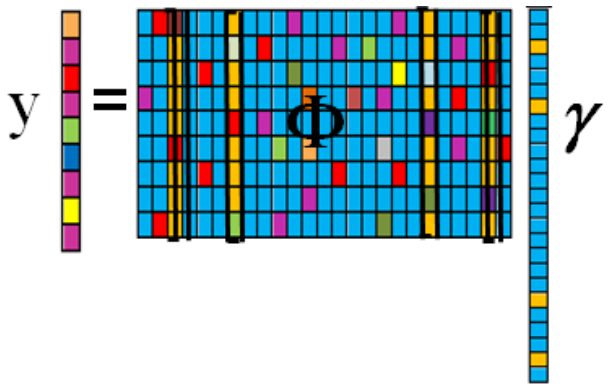

Figure 3.2.4: An illustration of Eq. (3.2.0.3), a block of interest $\mathbf{y}$ expressed as a linear combination of the columns from the dictionary Φ, which is formed by training atoms $\mathbf{a}_k$.

images. In our algorithm, during Stage one, we use the reference image to learn the dictionary. Our model is a block based dictionary which captures only local disease related changes and ignores the changes due to the patient positioning, etc. It uses overlapping blocks to reduce image block artifacts. From the assumption that the vector of coefficients $\boldsymbol{\gamma}$ is sparse, our method seeks to solve Eq. (3.2.0.3) for the sparsest $\boldsymbol{\gamma}$. Let's consider the following minimization problem:

$$\text{Problem } P_p: \quad \begin{aligned} \boldsymbol{\gamma}^* &= \arg\min_{\boldsymbol{\gamma}} ||\boldsymbol{\gamma}||_p \\ \text{subject to:} \quad & \mathbf{y} = \Phi\boldsymbol{\gamma} \end{aligned} \tag{3.2.0.4}$$

where $||.||_p$ represents the L_p norm. The L_p norm for a matrix $\mathbf{X}$ is:

$$||\mathbf{X}||_p = \Big(\sum_{j=1}^{n}\sum_{i=1}^{m} x_{ij}^p\Big)^{\frac{1}{p}} \tag{3.2.0.5}$$

For $p = 2$, we get the Frobenius norm or the Euclidean norm L_2:

$$||\mathbf{X}||_2 = \Big(\sum_{j=1}^{n}\sum_{i=1}^{m} x_{ij}^2\Big)^{\frac{1}{2}} \tag{3.2.0.6}$$

For $p = 1$, we get the L_1 norm as:

$$||\mathbf{X}||_1 = \sum_{j=1}^{n}\sum_{i=1}^{m} |x_{ij}| \tag{3.2.0.7}$$

Similarly, the L_p of a column vector $\mathbf{x} \in \mathbf{R}^{m\times 1}$ can be derived directly from Eq. (3.2.0.5), Eq. (3.2.0.6) and Eq. (3.2.0.7), i.e.,

$$||\mathbf{x}||_p = \Big(\sum_{i=1}^{m} x_{ij}^p\Big)^{\frac{1}{p}} \tag{3.2.0.8}$$

The Euclidean norm or L_2 norm of $\mathbf{x}$ is defined as:

$$||\mathbf{x}||_2 = \Big(\sum_{i=1}^{m} x_{ij}^2\Big)^{\frac{1}{2}} \tag{3.2.0.9}$$

The L_1 norm is the sum of absolute values of the elements of $\mathbf{x}$:

$$||\mathbf{x}||_1 = \sum_{i=1}^{m} |x_{ij}| \tag{3.2.0.10}$$

For $p = 0$, the L_0 norm of a vector $\mathbf{x}$ gives the total number of non-zero coefficients in $\mathbf{x}$:

$$||\mathbf{x}||_0 = \{\#i|x_i \neq 0\} \tag{3.2.0.11}$$

Finding a sparse $\boldsymbol{\gamma}$ can be done by minimizing the support of $\boldsymbol{\gamma}$, i.e., number of non-zero coefficients in $\boldsymbol{\gamma}$. It follows from the definition of $||\mathbf{x}||_0$ that the sparsest solution $\boldsymbol{\gamma}$ to Eq. (3.2.0.4) is obtained for $p = 0$. That is, by solving the minimization P_0 problem:

$$\text{Problem } P_0: \quad \begin{aligned} &\boldsymbol{\gamma}^* = \arg\min_{\gamma} ||\boldsymbol{\gamma}||_0 \\ &\text{subject to:} \quad \mathbf{y} = \Phi\boldsymbol{\gamma} \end{aligned} \tag{3.2.0.12}$$

If there is no change in the test image and in the ideal noiseless case, i.e, the reference and test image are identical, then the block of interest is represented by only one training block. Then, the solution $\boldsymbol{\gamma}^*$ to Eq.(3.2.0.12) is a vector of only one nonzero element. In reality, there is always noise, therefore solution $\boldsymbol{\gamma}^*$ vector will have a few nonzero elements of small magnitudes. To account for noise, Eq. (3.2.0.3) can be modified to:

$$\mathbf{y} = \Phi\boldsymbol{\gamma} + \mathbf{n} \tag{3.2.0.13}$$

where $\mathbf{n} = \mathbf{y} - \Phi\boldsymbol{\gamma}$ represents the random noise and the residual errors in the approximation algorithm. It is clear that we aim to compute the sparsest $\boldsymbol{\gamma}$ while minimizing the residual errors, $\mathbf{n}$. Then, the minimization problem P_0 in Eq. (3.2.0.12) can be written as:

$$\text{Problem } P_0: \quad \begin{aligned} &\boldsymbol{\gamma}^* = \arg\min_{\gamma} ||\boldsymbol{\gamma}||_0 \\ &\text{subject to:} \quad \mathbf{y} = \Phi\boldsymbol{\gamma} + \mathbf{n} \end{aligned} \tag{3.2.0.14}$$

The solution $\boldsymbol{\gamma}^*$ is sparse, that is, $||\boldsymbol{\gamma}^*||_0 << n$ where n is the dimension of the vector. The solution to P_0 problem is in general NP-hard (non-deterministic polynomial-time hard). In many applications, instead of solving the P_0 problem, an approximate solution is obtained by replacing the L_0 norm with L_1 norm [112, 111, 113]. In this case, the L_0 minimization problem in Eq. (3.2.0.14) can be relaxed into the following L_1 convex minimization problem P_1:

$$\text{Problem } P_1: \quad \begin{aligned} &\boldsymbol{\gamma}^* = \arg\min_{\gamma} ||\boldsymbol{\gamma}||_1 \\ &\text{subject to:} \quad \mathbf{y} = \Phi\boldsymbol{\gamma} + \mathbf{n} \end{aligned} \tag{3.2.0.15}$$

The solution to the convex problem P_1 can be computed by standard optimization tools such as linear programming. As described in Section 2.4, authors in [112] have shown that for very sparse representation of $\mathbf{y}$ when there exists $\boldsymbol{\gamma}$ so that $\mathbf{y} = \boldsymbol{\Phi}\ \boldsymbol{\gamma}$ and $||\gamma||_0 \ll n$, then the solution to problem P_0 is also a solution to the P_1 problem. In other words, Problem P_1 is among convex optimization problems that is the closest to the P_0 optimization problem under sparsity and incoherence conditions. For dictionary Φ with incoherent columns, if there is a sufficiently sparse solution to the P_0 problem, then the solution is unique and it is equal to the solution of the P_1 problem, [5, 6, 111, 112, 113].

We write Eq.(3.2.0.13) in matrix representation:

$$\mathbf{y} = [\Phi\ \mathbf{I}] \begin{bmatrix} \boldsymbol{\gamma} \\ \mathbf{n} \end{bmatrix} \equiv \mathbf{A}\boldsymbol{\alpha} \tag{3.2.0.16}$$

For every vector $\mathbf{y}$ from the test image we determine the sparsest representation $\boldsymbol{\alpha}^*$ of all $\boldsymbol{\alpha}$'s $\in \mathbf{R}^{n+m}$ which minimizes the error and satisfies Eq.(3.2.0.15). For efficient

computation, we solve the following minimization problem.

$$\text{Problem } P_1: \quad \begin{aligned} \boldsymbol{\alpha}^* &= \arg\min_{\boldsymbol{\alpha}} ||\boldsymbol{\alpha}||_1 \\ \text{subject to:} \quad \mathbf{y} &= \mathbf{A}\boldsymbol{\alpha} \end{aligned} \tag{3.2.0.17}$$

where

$$\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\gamma} \\ \mathbf{n} \end{bmatrix} \text{ and } \quad \boldsymbol{\alpha}^* = \begin{bmatrix} \boldsymbol{\gamma}^* \\ \mathbf{n}^* \end{bmatrix}$$

If there are changes in the test image then the block of interest may not be well represented as a linear combination of a few blocks from the inquiry block. Enforcing the sparsity conditions when solving the P_1 minimization problem leads to a noisy approximation of the block $\mathbf{y}^*$ from all the training blocks of $\boldsymbol{\Phi}$. To reduce the noise, we modify $\mathbf{A}$ slightly by spreading out the mean value μ of the inquiry block $\mathbf{B}$.

$$\mathbf{A} = [\Phi \ \mu\mathbf{I}] \tag{3.2.0.18}$$

where $\mathbf{I}$ is the $m \times m$ identity matrix. An orthogonal matching pursuit algorithm can be used to solve Eq.(3.2.0.17). In our numerical computation we use the compressive sampling matching pursuit (CoSaMP) algorithm [12] to compute $\boldsymbol{\alpha}^*$ by minimizing the L_1 norm. The CoSaMP algorithm compares with the best optimization-based approaches. Moreover, it is computationally more efficient, requires less storage, and is more efficient for practical problems since it requires only a matrix multiplied with the sampling matrix [12]. The best approximation $\mathbf{y}^*$ of the background of $\mathbf{y}$, the block of interest from the test image, can be computed as a linear combination of these training blocks from inquiry block in the reference image.

$$\mathbf{y}^* = \Phi \ \boldsymbol{\gamma}^* \tag{3.2.0.19}$$

and the foreground $\mathbf{r}$ which represent clinically relevant changes are computed as:

$$\mathbf{r} = |\mathbf{y} - \mathbf{y}^*| \tag{3.2.0.20}$$

3.3 Dictionary Learning and The AEDL Algorithm via Eigen-Subspace

In Eq.(3.2.0.18), dictionary Φ of size $m \times n$ where $n = (2\Delta + 1)^2$ composed of reference image atoms, presents high level of redundancy. This is because these atoms highly overlap and many atoms differ by only one row or column. We use principal component analysis, PCA, to reduce the dimensionality of the dictionary dataset and eliminate such redundancy. The general idea is to approximate each column $\mathbf{x}$ of Φ with its projected column $\widehat{\mathbf{x}}$ of $\widehat{\Phi}$ over a projection matrix $\mathbf{T}$, such that:

$$\widehat{\Phi} = \mathbf{T}\ \Phi \tag{3.3.0.21}$$

where $\mathbf{T}$ is a $m \times m$ linear transform. The best choice for the transform $\mathbf{T}$ would be the one that minimizes the residual error of the approximation in the least mean square sense (MSE) [79]. If $\mathbf{x}$ is approximated with its projection $\widehat{\mathbf{x}}$, then the estimated MSE of the approximation is defined in [115] as:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(\widehat{x}_i - x_i)^2 \tag{3.3.0.22}$$

and the unbiased MSE is defined in [114] as:

$$MSE = \frac{1}{m-1}\sum_{i=1}^{m}(\widehat{x}_i - x_i)^2 \tag{3.3.0.23}$$

Let $\mathbf{T}$ be the orthogonal projection onto the space spanned by $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_r$ eigenvectors of covariance matrix of Φ, i.e.,

$$\mathbf{T} = \sum_{s=1}^{r}\mathbf{U}_s\ \mathbf{U}_s^T \tag{3.3.0.24}$$

To do so, we first center the dictionary Φ to have zero mean, by subtracting $\mathbf{x}^c$ the mean column of Φ. The centered dictionary with zero mean Φ^c is:

$$\Phi^c = [\mathbf{x}_1^c, \mathbf{x}_2^c, \ldots, \mathbf{x}_n^c] \tag{3.3.0.25}$$

where $\mathbf{x}_i^c = \mathbf{x}_i - \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_k$. Φ^c can be written as:

$$\Phi^c = \Phi - \mathbf{x}^c = \Phi - \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_k \tag{3.3.0.26}$$

Eigenvalues and eigenvectors of the covariance matrix $\Omega = \Phi^c(\Phi^c)^T$ are then computed. We project the dictionary Φ^c onto eigen-subspace of Ω to obtain the projection $\widehat{\Phi}$ as:

$$\widetilde{\Phi} = \mathbf{U}^T\ \Phi^c \tag{3.3.0.27}$$

We use r eigenvectors to learn the dictionary, extract the features and reduce the dimensions of the dictionary, where $r \leq$ to the rank of Φ^c and $r \ll m$. The low dimensional dictionary $\widehat{\Phi}$ is obtained by projecting Φ^c onto eigen-subspace:

$$\widehat{\Phi} = \mathbf{U}_r^T\ \Phi^c \tag{3.3.0.28}$$

where $\mathbf{U}_r$ is the matrix of $\mathbf{r}$ eigenvectors corresponding to the r largest eigenvalues. The vector $\mathbf{y}$ is also projected into the subspace obtained by the eigenvectors of the covariance matrix as:

$$\widehat{\mathbf{y}} = \mathbf{U}_r^T\ \mathbf{y}^c \tag{3.3.0.29}$$

where $\mathbf{y}^c$ is the centered block of interest, obtained by subtracting $\mathbf{x}^c$ the mean column of the dictionary Φ. Then, the Eq.(3.2.0.18) can be written as:

$$\widehat{\mathbf{A}} = [\widehat{\Phi}\ \mu\mathbf{I}] \tag{3.3.0.30}$$

The initial problem (3.2.0.17) in eigen-subspace can be now written as:

$$\text{Problem } P_1 : \quad \begin{array}{c} \widehat{\boldsymbol{\alpha}}^* = \arg\min_{\widehat{\boldsymbol{\alpha}}} ||\widehat{\boldsymbol{\alpha}}||_1 \\ \text{subject to:} \quad \widehat{\mathbf{y}} = \widehat{\mathbf{A}}\widehat{\boldsymbol{\alpha}} \end{array} \tag{3.3.0.31}$$

$$\text{where} \quad \widehat{\boldsymbol{\alpha}} = \begin{bmatrix} \widehat{\boldsymbol{\gamma}} \\ \widehat{\mathbf{n}} \end{bmatrix} \text{ and } \quad \widehat{\boldsymbol{\alpha}}^* = \begin{bmatrix} \widehat{\boldsymbol{\gamma}}^* \\ \widehat{\mathbf{n}^*} \end{bmatrix}$$

Using the solution $\widehat{\boldsymbol{\alpha}}^*$ of Eq. (3.3.0.31) the AEDL algorithm computes $\widehat{\mathbf{y}}^*$, i.e., an estimate of the projected background of $\widehat{\mathbf{y}}$ in the selected eigen-subspace, by solving the following linear system:

$$\widehat{\mathbf{y}}^* = \widehat{\Phi}\, \widehat{\boldsymbol{\gamma}}^* \tag{3.3.0.32}$$

Therefore, we reconstruct $\mathbf{y}^*$, the background of $\mathbf{y}$ in spatial domain, by solving the following equation.

$$\widehat{\mathbf{y}}^* = \mathbf{U}_r^T \mathbf{y}^* \tag{3.3.0.33}$$

We observe that the matrix of eigenvectors $\mathbf{U}_r$ is full rank and the number of rows is greater or equal to the number of columns, $m \geq r$. Therefore, the (left) pseudo inverse of $\mathbf{U}_r^T$ exists. Note that the left pseudo inverse of a matrix $\mathbf{V}$ is: $\mathbf{V}^\dagger = (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T$. Hence, the left pseudo inverse of $\mathbf{U}_r^T$ can be computed as:

$$\left(\mathbf{U}_r^T\right)^\dagger = \left(\left(\mathbf{U}_r^T\right)^T \mathbf{U}_r^T\right)^{-1} \left(\mathbf{U}_r^T\right)^T \tag{3.3.0.34}$$

$$= \left(\mathbf{U}_r \mathbf{U}_r^T\right)^{-1} \mathbf{U}_r \tag{3.3.0.35}$$

The background in the spatial domain is then given as:

$$\mathbf{y}^* = \left(\mathbf{U}_r \mathbf{U}_r^T\right)^{-1} \mathbf{U}_r \widehat{\mathbf{y}}^* \tag{3.3.0.36}$$

and the residual error of the approximation $\mathbf{y}^*$ with the block of interest $\mathbf{y}$ is then computed as:

$$\mathbf{r} = |\mathbf{y} - \mathbf{y}^*| \tag{3.3.0.37}$$

The AEDL algorithm uses overlapping training blocks to learn the background of the block of interest. Therefore, to compute the foreground with significant changes corresponding to the block of interest $\mathbf{b}$, this process is repeated m times, the number of pixels in the block of interest. $\mathbf{F}^{[1]}$ represents the foreground of the block of interest $\mathbf{b}$ in the test image and is learned from training blocks in the reference image.

$$\mathbf{F}^{[1]} = \frac{1}{m} \sum_{k=j-\delta}^{j+\delta} \sum_{l=i-\delta}^{i+\delta} \mathbf{r}_{k,l}^{[1]} \tag{3.3.0.38}$$

where $\mathbf{r}_{k,l}^{[1]}$ represents residual vectors and $\mathbf{F}^{[1]}$ is the foreground represented by significant changes corresponding to the appearance or disappearance of a lesion, the growth or shrinkage of an existing lesion and also changes in the intensity level of a lesion. The foreground blocks are image blocks of size $\sqrt{m} \times \sqrt{m} = (2\delta + 1) \times (2\delta + 1)$, centered at the pixel (i,j), and divided by m, the number of blocks that contain the pixel (i,j). In Stage one the AEDL algorithm computes the change image $\mathbf{I}_{cd}^{[1]}$ between the reference and the test images using the following equation:

$$\mathbf{I}_{cd}^{[1]} = \frac{1}{m} \sum_{j=\delta}^{N_2-\delta} \sum_{i=\delta}^{N_1-\delta} \mathbf{r}^{[1]}(i,j) \tag{3.3.0.39}$$

In Stage two the background of each block of interest in reference image is learned from training blocks in the test image. Hence, we repeat the process of Stage one by switching the reference image with the test image. The foreground $\mathbf{F}^{[2]}$ of a block in the reference image is computed as:

$$\mathbf{F}^{[2]} = \frac{1}{m} \sum_{k=j-\delta}^{j+\delta} \sum_{l=i-\delta}^{i+\delta} \mathbf{r}_{k,l}^{[2]}(i,j) \tag{3.3.0.40}$$

The change image $\mathbf{I}_{cd}^{[2]}$ between the reference and test images is computed as:

$$\mathbf{I}_{cd}^{[2]} = \frac{1}{m} \sum_{j=\delta}^{N_2-\delta} \sum_{i=\delta}^{N_1-\delta} \mathbf{r}^{[2]}(i,j) \tag{3.3.0.41}$$

Then in Stage three, the two change images $\mathbf{I}_{cd}^{[1]}$ and $\mathbf{I}_{cd}^{[2]}$ are averaged to create the final change image $\mathbf{I}_{cd}$ between the reference and the test image:

$$\mathbf{I}_{cd} = \frac{1}{2} \left(\mathbf{I}_{cd}^{[1]} + \mathbf{I}_{cd}^{[2]} \right) \tag{3.3.0.42}$$

which can be simplified as:

$$\mathbf{I}_{cd} = \frac{1}{2} \left(\frac{1}{m} \sum_{j=\delta}^{N_2-\delta} \sum_{i=\delta}^{N_1-\delta} \mathbf{r}^{[1]}(i,j) + \frac{1}{m} \sum_{j=\delta}^{N_2-\delta} \sum_{i=\delta}^{N_1-\delta} \mathbf{r}^{[2]}(i,j) \right) \tag{3.3.0.43}$$

and finally the change image $\mathbf{I}_{cd}$ is

$$\mathbf{I}_{cd} = \frac{1}{2m} \sum_{j=\delta}^{N_2-\delta} \sum_{i=\delta}^{N_1-\delta} \Big(\mathbf{r}^{[1]}(i,j) + \mathbf{r}^{[2]}(i,j) \Big) \tag{3.3.0.44}$$

3.4 Simulations

First, we tested the AEDL method with synthetic images of size 128×128 as shown in Figs. 3.4.1, 3.4.2, 3.4.3, 3.4.4. In each reference image, there are three 12 sided polygons placed in the first 32 rows. The size of each polygon increases from left to right and pixels inside each polygon were given intensities using Gaussian distribution with maximum intensity 0.3, 0.2, 0.4 respectively, from left to right. To test the algorithm for different shift's sizes, rotation angles, noise and intensity levels, the first 32 rows have been repeated for the second, third and fourth 32 rows of the reference image.

The test image is created by making two types of changes to the reference image, significant and insignificant changes. Changes such as shifts, rotations, and different noise levels, added to different strips of the reference image, are considered as **insignificant** changes. In addition to insignificant changes, the reference image has gone under other changes, considered as **significant**, such as object disappearing (the second shape in the second 32 rows is deleted), a new object appearing (a new shape is added down-right), object enlarged (the shape in the fourth 32 rows and first 32 columns is enlarged) , and finally, object shrinkage (the third polygon of the second 32 rows is shrunk).

These significant changes make up our ground truth image. The desired outcome of a change detection algorithm is to detect only the significant changes that make up the ground truth image. The performance of the AEDL algorithm is compared with the ground truth and the absolute differencing method.

In Fig. 3.4.1 from the top to the bottom, the polygons in the strips of 32 rows of reference image are shifted 1-down and 1-left, 3-down and 2-right, 2- down and 3-left, and 3-up and 4-right respectively. All considered insignificant changes. The AEDL algorithm ignores the changes related to horizontal and vertical translations of polygons in the strips when the radius of the inquiry block $\mathbf{B}$ is no less than the size of the shift.

As Fig. 3.4.1 shows, it detects the real and significant changes closer to the ground truth, with $\delta = 1, c = 3$ and $\Delta = \delta + c = 4$, compared to the absolute difference.

In Fig. 3.4.2 from the top to the bottom, the polygons in the strips of 32 rows of reference image are rotated by 2 degrees clockwise, 3 degrees counter-clockwise, 4 degrees clockwise respectively and the polygons in the last 32 rows by 6 degrees counter-clockwise. All considered as insignificant changes. Results show that the AEDL algorithm ignores the changes related to rotation of strips when the radius of $\mathbf{B}$ is no less than the angle of rotation and detects the significant changes closer to the ground truth with $\delta = 2, c = 3$ and $\Delta = \delta + c = 5$.

In Fig. 3.4.3, Gaussian noise is added to the reference image, with SNR = 30, 40, 50, and 60 for the first, second, third, and fourth 32 rows respectively. Noise is considered as another type of insignificant change. The simulations show that the AEDL algorithm still detects the real changes closer to the ground truth for SNR$\geq$ 30 with $\delta = 1, c = 3$ and $\Delta = \delta + c = 4$.

In Fig. 3.4.4, starting from the second strip of 32 rows in the reference image, the intensities of shapes on the second, third, and fourth strips are increased by 0.1, 0.3, and 0.5 respectively. We want our algorithm to detect changes due to the intensity changes, as they are considered disease related by the radiologists. Note that the polygons at the bottom of the test image have a maximum intensity of 0.9 in the reference images. We added these changes to mimic different lesion activities. Fig. 3.4.4 shows that the AEDL algorithm detects the significant changes closer to the ground truth, with $\delta = 1, c = 1$ and $\Delta = \delta + c = 2$ compared to the absolute difference.

In Fig. 3.4.5, we also tested the AEDL algorithm with MR images. The image at row i) and column c) shows the results of two real T2-weighted MR images taken in 2010 and 2012. Both images have the same inactive and visible MS lesion, which hasn't grown or changed between two exams periods, as confirmed by the radiologist. The overlaid test image with the colored changed image is the same as the test image as shown in Column e). This confirms that there are no significant changes for the algorithm to detect, whereas the absolute difference shows changes due to the image registration which are insignificant and should be ignored. Row ii) columns a) and b)

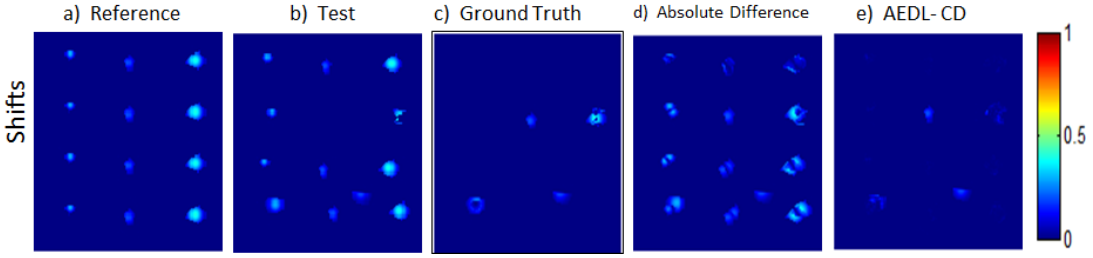

Figure 3.4.1: Testing the algorithm performance in the presence of object shifts. a) Reference image, b) Test image, c) Ground truth image showing only the significant changes, d) Absolute difference image between the test and reference images, e) Change image obtained by the AEDL algorithm shows that the algorithm ignores the changes related to translations that can be captured by local dictionaries, but detects the real and significant changes closer to the ground truth.

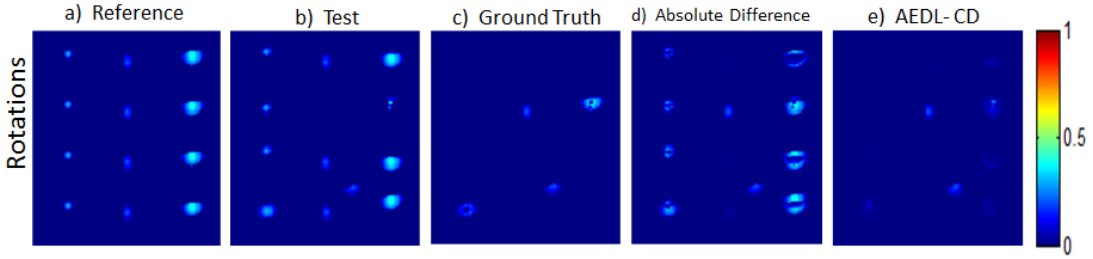

Figure 3.4.2: Testing algorithm performance in the presence of object rotations. a) Reference image, b) Test image, c) Ground truth image showing only the significant changes, d) Absolute difference image between the test and reference images, e) Change image obtained by the AEDL algorithm shows that our method ignores the changes related to rotations that can be captured by local dictionaries, but detects the significant changes closer to the ground truth.

show that two T2-weighted MR images of a normal brain and a brain with moderate MS lesions, with 1mm slice thickness, 3% noise, and 0% intensity non-uniformity (RF). Images are obtained from brainWeb: simulated brain database [17]. Column c) shows that the AEDL algorithm finds the significant changes related to the new MS lesion formation and ignores changes detected by the absolute difference method as shown in column d).

3.5 Summary

In this section we presented an application of adaptive dictionary learning techniques with our first change detection algorithm, the AEDL. The algorithm captures local spatial changes between the reference and test images and consists of three stages. In the first stage, the change image is obtained by averaging all the absolute differences

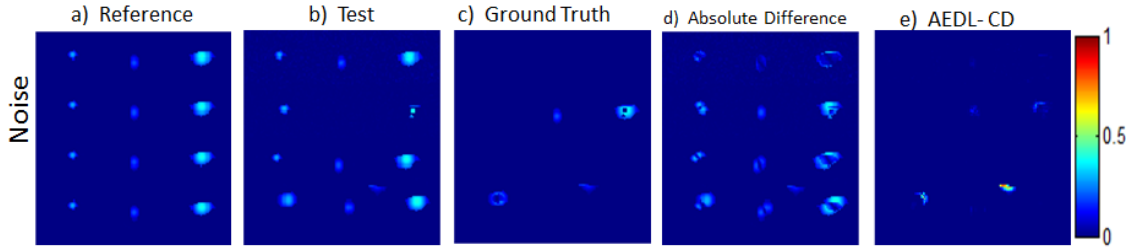

Figure 3.4.3: Testing the AEDL algorithm performance in the presence of different noise level. a) Reference image, b) Test image, c) Ground truth image showing only the significant changes, d) Absolute difference image between the test and reference images, e) Change image obtained by the AEDL is very closer to the ground truth and shows that our algorithm method detects the real changes for SNR ratio > 25.

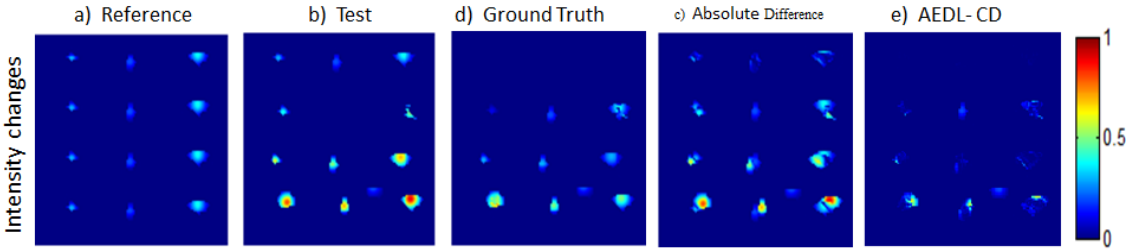

Figure 3.4.4: Testing the AEDL algorithm performance in the presence of different intensity level. a) Reference image. b) Test image c) Ground truth image showing only the significant changes. d) Absolute difference image between the test and reference images. e) Change image obtained by the AEDL algorithm detects the significant changes closer to the ground truth image.

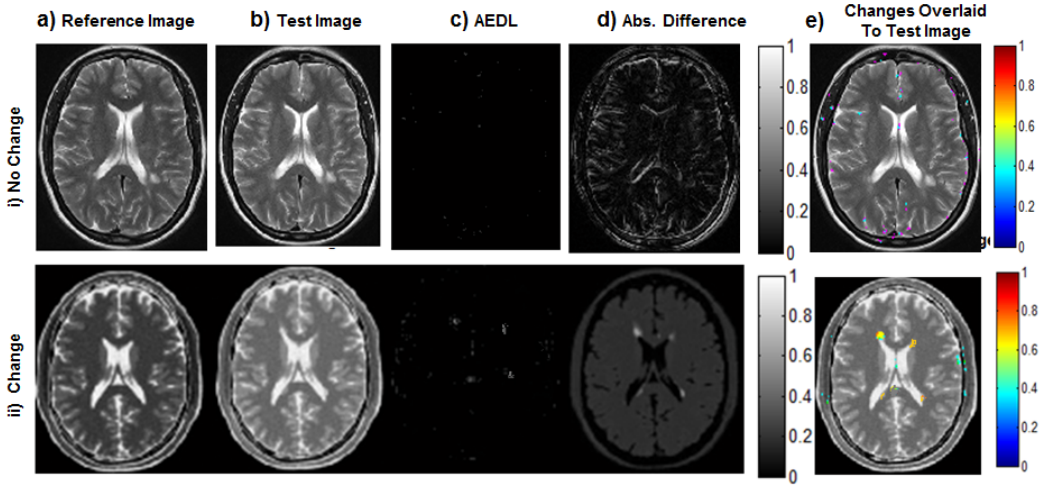

Figure 3.4.5: Row i) Columns a) and b) are the reference and test images, c) the change image computed by the AEDL method, d) the absolute difference image between the reference and test images, e) the test image overlaid with the coloured change image. Row ii) Columns a) and b) are T2-weighted MR images of a normal brain and a brain with moderate MS lesions respectively, simulated by BrainWeb [17]. Column c) the change image computed by the AEDL algorithm, d) the absolute difference image between the reference and test images, e) the test image overlaid with the coloured change image.

of a test image block with its best linear approximations given by a local dictionary trained from the reference image. In Stage two the process is similar, but the local dictionaries are learned from the test image blocks. In the third stage, the change image between the reference and test image is computed as the average of change images from Stages one and two.

The sparse approximation is computed by the L_1 - minimization algorithm. This algorithm automatically detects the significant structural changes in the test image while ignoring unimportant changes related to mis-alignments, noise and acquisition-related artifacts. Experiments on synthetic images illustrated that our algorithm detects changes due to appearance or disappearance of objects in the presence of object translations, rotations, intensity changes, and moderate noise level. Experiments on MR images showed that our algorithm identifies clinically significant changes and rejects clinically insignificant changes. Note that the rejection of unimportant changes has to be within areas of local dictionaries. In addition, we introduced the use of PCA to reduce the size of a local dictionary for each image block, to find the sparse representation of the background and improve computational efficiency.

The AEDL algorithm is able to detect changes in presence of small shifts and rotations, and noise. To account for large shifts, rotations, and noise, the AEDL algorithm employs large sizes for both block of interest and inquiry blocks. This requires reconstruction of relatively larger blocks, and hence the L_1 minimization becomes more computationally expensive. This limitation of the AEDL algorithm is addressed in Section 4 by designing our second adaptive dictionary learning based algorithm, named EigenBlock Change Detection algorithm (EigenBlockCD).

Chapter 4

EigenBlock Change Detection Algorithm In Two Dimensions

4.1 An Introduction

In Chapter 3 we described the AEDL algorithm for detecting changes between consecutive medical images. The AEDL algorithm detects relevant clinical changes close to the ground truth while ignoring changes due to patient position and other acquisition artifacts. The AEDL algorithm is based on local adaptive dictionary learning techniques and computes the background for each block of interest using L_1 minimization techniques. The reconstruction of the background for each block of interest makes this method computationally expensive. We adapted the idea behind AEDL and designed a new algorithm to overcome this limitation.

In this Chapter we describe our second algorithm EigenBlock Change Detection algorithm (EigenBlockCD) for detecting changes based on local dictionary learning techniques. Similar to AEDL algorithm, our goal is to perform a local image registration for identifying important structural changes such as appearing of a new lesion or disappearing of an existing one, and the growth or the shrinkage of a lesion. Meanwhile our method aims to automatically reject unimportant changes due to spatial position of patients and noise.

To simplify the problem we focused on detecting changes between two images, namely, reference and test images. As in Chapter 3, the two images are not necessarily

aligned and may contain different noise levels and other acquisitions related artifacts. For each block in the test image, we aim to find the best matching block from the corresponding local dictionary learned from the reference image, instead of seeking a linear approximation of these dictionary blocks as in AEDL algorithm. This idea is inspired by face recognition in [1]. The EigenBlockCD algorithm [23] consists of three stages as illustrated in Fig. 4.4.1:

1. Both reference and test images are divided into many overlapping blocks as shown in Fig. 4.4.1. For each block in the test image, a local dictionary is built by all blocks of the same size existing in a neighborhood of the position (i, j) in the reference image. Blocks from the test image and its corresponding dictionary are projected onto a low dimensional eigen-subspace of the covariance matrix of the dictionary to speed up the process. The algorithm captures local spatial changes between the reference and test images by averaging all the absolute differences of a test image block with its best approximations block found in a local dictionary trained from neighboring blocks in the reference image.

2. Stage One repeats by averaging all the absolute differences of a reference image block with its best approximation block found in a local dictionary trained from the test image.

3. Finally, the two images obtained from Stages One and Two are averaged to form the change image.

The algorithm uses the L_2 norm as similarity measure, i.e., the minimum error of reconstructed change image. We also apply knowledge of principal component analysis to eliminate the redundancy and hence to increase the computational efficiency. In Section 4.3 we discuss theoretical and numerical advantages of using L_2 norm in our EigenBlockCD algorithm. The performance of the algorithm is demonstrated using both synthetic and real images.

In Section 4.2 we discuss PCA-based learning as a way to obtain the best approximation of the dictionary. In Section 4.3 we give the rationale of using L_2 norm in our EigenBlockCD algorithm. In Section 4.4 we discuss the the implementation of the

EigenBlockCD algorithm. In Section 4.5 we present the numerical experiments with synthetic and real images and analyze the results to evaluate the performance of the algorithm. Results of our simulations are shown in Figs. 4.5.1 through 4.5.9.

4.2 PCA-Based Learning

In both the AEDL and the EigenBlockCD algorithms the dictionary $\mathbf{\Phi}$ is of size $m \times n$ with $n = (2\Delta + 1)^2$. The dictionary is composed of reference image atoms and has high level of redundancy because many pixels are included in more than one atom. We thus use PCA to reduce the dimensionality of the dictionary dataset and eliminate such redundancy, therefore decrease the computational time.

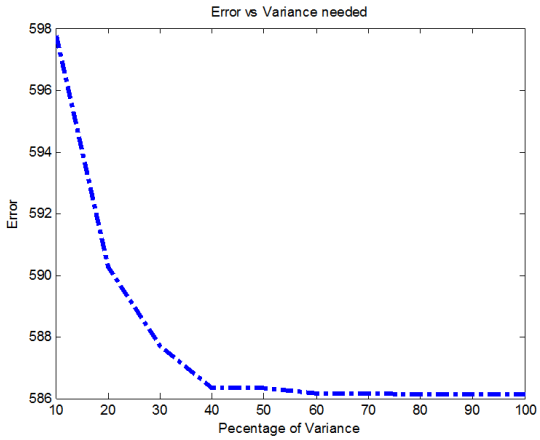

Figure 4.2.1: L_2 error versus the percentage of variance preserved in the image block. Horizontal axis: Percentage of variance from 10 to 100. Vertical axis: error calculated as the mean error of ten different blocks

The purpose of using the PCA in our algorithm is two-fold: 1) it is used as a feature extraction tool by emphasizing most significant features within each local dictionary. 2) it is used as a dimensional reduction tool to reduce the dimensionality of the dictionary and hence increase its computational efficiency [24].

To quantify the runtime reduction by the use of the PCA in our algorithm we compared runtime versus block size curves of the algorithms with and without the use of PCA. The optimal number of eigenvectors selected is determined by the percentage of the variance that minimizes the L_2 norm of the error as shown in Fig.4.2.1. The error is defined as the L_2 norm of the difference between the block of interest and the best matching block found by the algorithm.

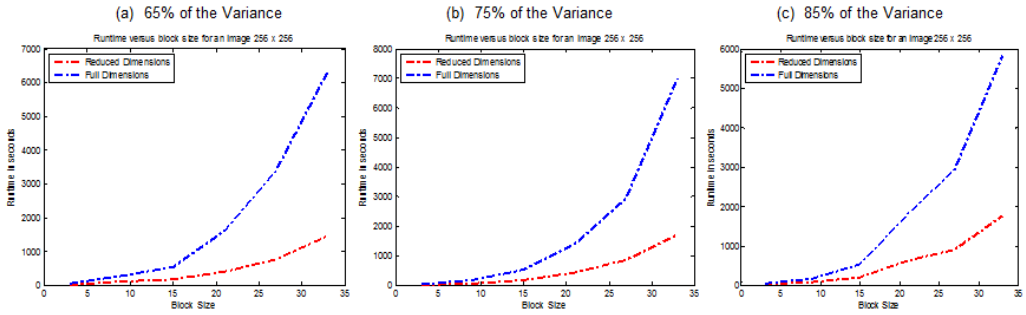

Figure 4.2.2: Comparison of the runtime of the algorithm versus block size with and without the use of PCA. The number of eigenvectors is selected respectively according to (a) 65% (b) 75% and (c) 85% of the total variance.

We run experiments with 23 blocks of interest chosen at different locations in the test image. Blocks are of size 21×21 and their corresponding inquiry blocks are of size 51×51 producing a dictionary of size 441×2601. Fig.4.2.1 shows that, in general, average error for all 23 blocks decreases as the variance increases. Moreover, there is no significant reduction in error when the percentage of the variance reaches above a certain value, e.g., 65%.

To see how the PCA dimensional reduction decreases the runtime we performed three experiments with the dictionary size reduced by choosing 65%, 75% and 85% variances. The size of the block of interest varies from 3×3 to 49×49, while the inquiry block size varies from 15×15 to 73×73. We estimated the runtime for both the reduced and the full size dictionary. As we can observe in Fig.4.2.2, the runtime is reduced by at least 50% when the block size is more than 3×3. The time saving is more significant as the block size increases.

4.3 Effects of L_1 And L_2 Norms In Subspace Learning

The choice of metric is critical in determining which block from the dictionary $\mathbf{\Phi}$ is the "closest" to the block of interest $\mathbf{b}$. Furthermore, the use of PCA requires that the projection of dictionary atoms onto eigen-subspace preserves distances. We exploit the fact that L_2 norm is invariant under an orthonormal transformation of image functions. We show that such a property does not hold for L_1 norm, that is, L_1 norm doesn't

preserve the distance and the the ordering of the distances and image structures under orthonormal transformations.

Here we represent an image as an N-dimensional matrix of pixels where the value of each entry is the grayscale value of the corresponding pixel and N represents the total number of pixels in the image. This is the spatial domain representation of an image and it is one of infinitely many spaces in which the image can be examined. Eigenspace is one of the most interesting spaces which is created by the eigenvectors of the covariance matrix of the training data.

Projecting images into a subspace of such an eigenspace has been widely used in image recognition and particularly in face recognition [1]. First, a subspace is selected to project images and then all the training images and the test image are projected into this subspace. Finally, the projected test image is compared to each projected training image using a distance measure. The training image which is the "closest" in terms of the selected distance measure is labeled as found.

In many applications PCA is used as a linear transformation for reducing dimensionality of a data set. Unlike other linear transforms, its basis vectors depends on the data set, therefore PCA does not have a fixed set of basis vectors. The first principal component is the direction in feature space along which the projections have the largest variance. The second is the direction which maximizes the variance of all directions orthogonal to the first component. Similarly, each k^{th} principal component is the direction which maximizes the variance among all directions orthogonal to the previous k-1 components.

We extend the idea of eigen-subspace learning to find the changes between two MR images. Selecting the appropriate distance measure greatly affects the performance of the algorithm. Therefore, it is useful to examine the performance of two metrics in the algorithm from both theoretical and practical point of view.

In the change detection problem for MR images we are interested in identifying blocks from the reference image that are most similar to the block of interest. Similarity in this case means that they have similar structural features.

Projecting data into eigen-subspace is similar to rotating the original coordinate

system around the origin. When selecting the distance measure, or the similarity measure, we need to ensure that the selected measure preserves distances under an orthonormal change of basis. It can be easily proven that L_2 norm is invariant under an orthonormal change of basis.

The concept of vector space can be generalized to a function space spanned by a set of basis functions. Let us consider the function space of image functions. A two-dimensional image can be viewed as a function of the image intensity $f(x)$ at a pixel or voxel x: $f : \mathbb{R}^2 \to \mathbb{R}$ for gray images and $f : \mathbb{R}^2 \to \mathbb{R}^3$ for colored images, where $x \in X \subset \mathbb{R}^2$. The inner product in this space is defined as:

$$< f(x), g(x) >= \int f(x)\overline{g(x)}dx \tag{4.3.0.1}$$

The following theorem proves that the L_2 norm, as a distance measure, is invariant with respect to orthonormal projections. This means that if a vector is projected into an orthonormal subspace then the length of the projected vector computed via L_2 would not change. It is also true that the covariance between a set of vectors is invariant with respect to an orthonormal projection. In other words, the angles between vectors in one domain are equal to angles between their corresponding projections into an orthonormal subspace.

Theorem 4.3.1: Orthonormal Transformations Preserve L_2 Norm Of An Image Function Let $\mathcal{P} = \{p(x) \mid x \in \mathbb{R}^2\}$ be an orthonormal basis, i.e.,$< p(x), p(y) >= 0$, for $x \neq y$ and, $< p(x), p(y) >= 1$, for $x = y$. Let $span(\mathcal{P})$ be the space spanned by this orthonormal basis. If $g(x)$ is the projected image function of $f(x)$ into this orthonormal basis, i.e., $g(x) =< p(x), f(x) >= \overline{p(x)}\ f(x)$, then

$$||g(x)||_2 = ||f(x)||_2 \tag{4.3.0.2}$$

Note: These are known facts about L_2 norm, we want to include the proof here so that the thesis is self-contained.

Proof: Following the definition of L_2 norm,

$$\begin{aligned} ||f(x)||_2^2 &= \int\limits_{x\in X} |f(x)|^2 dx = \int\limits_{x\in X} f(x)\overline{f(x)}dx, \text{ and} \\ ||g(x)||_2^2 &= \int\limits_{x\in X} |g(x)|^2 dx = \int\limits_{x\in X} g(x)\overline{g(x)}dx, \text{ where } x \in X \subset \mathcal{R}^2 \end{aligned}$$

we have:

$$\begin{aligned} ||g(x)||_2^2 &= \int\limits_{x\in X} |g(x)|^2 dx \\ &= \int\limits_{x\in X} |\overline{p(x)}f(x)|^2 dx \\ &= \int\limits_{x\in X} \Big(\overline{p(x)}f(x)\Big)\Big(\overline{\overline{p(x)}f(x)}\Big)dx \\ &= \int\limits_{x\in X} \Big(\overline{p(x)}f(x)\Big)\Big(\overline{f(x)}p(x)\Big)dx \\ &= \int\limits_{x\in X} \overline{p(x)}\Big(f(x)\overline{f(x)}\Big)p(x)dx \end{aligned}$$

$$\begin{aligned} &= \int\limits_{x\in X} \overline{p(x)}\Big(|f(x)|^2\Big)p(x)dx \\ &= \int\limits_{x\in X} |f(x)|^2\Big(\overline{p(x)}p(x)\Big)dx \\ &= \int\limits_{x\in X} |f(x)|^2\Big(\overline{p(x)\overline{p(x)}}\Big)dx \\ &= \int\limits_{x\in X} |f(x)|^2 dx \\ &= ||f(x)||_2^2 \qquad \blacksquare \end{aligned}$$

Next, we provide another proof using matrices.

Theorem 4.3.2: Orthonormal Transformations Preserve L_2 Norm Of A Matrix Let X and Y be two $m \times n$ matrices and let U be a matrix of size $m \times r$ where $r \ll n$ and columns of U are orthonormal. Then

$$||U^T X - U^T Y||_2 = ||X - Y||_2 \tag{4.3.0.3}$$

Proof: Following the definition of L_2 norm:

$$\begin{aligned} ||U^T X - U^T Y||_2^2 &= (U^T X - U^T Y)^T (U^T X - U^T Y) \\ ||X - Y||_2^2 &= (X - Y)^T (X - Y) \end{aligned}$$

we have:

$$\begin{aligned} ||U^T X - U^T Y||_2^2 &= ||U^T(X-Y)||_2 \\ &= X^T X - X^T Y - Y^T X + Y^T Y \quad \text{from} \quad U\text{-orthonormal} \\ &= (X^T - Y^T)(X-Y) \\ &= (X-Y)^T(X-Y) \\ &= ||X-Y||_2^2 \qquad \blacksquare \end{aligned}$$

This means that the L_2 norm preserves the distance during an orthonormal projection. This result does not hold for L_1 norm which we illustrate below with four counter-examples.

Counter-Example 4.3.1-Graphical Illustration of L_1 not preserving distance in 2D

Rotation is an isometric transform, it preserves the Euclidean distances between two points or two segments and the angle between two segments. As shown in Fig. 4.3.1, when vector **AB** is rotated around origin O obtaining vector $\mathbf{A'B'}$, the L_2 norm is the same but the L_1 norm is different. The only case when the L_1 norm is preserved is when the angle of rotation is a multiple of 90 degrees, i.e., $\theta = k \times \frac{n}{2}$, for $k \in Z$.

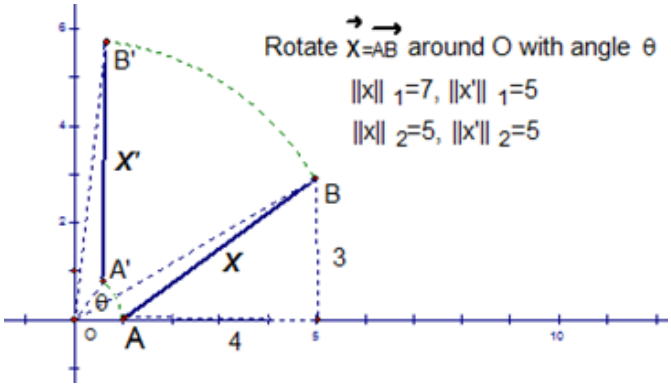

Figure 4.3.1: The L_2 norm is preserved when the Cartesian system has been rotated around O with angle θ. However, L_1 norm is not preserved. Note: θ is chosen such that the transformed vector $\mathbf{A'B'}$ of **AB** under this rotation is vertical vector.

Counter-Example 4.3.2-Specific Counter-Example for L_1-Distance Preservation in 2D :

Let $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{y}$ be three vectors in two dimensional space spanned by the traditional

orthonormal basis $\mathbf{e}_1$ and $\mathbf{e}_2$.

$$\mathbf{x}_1 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \mathbf{y} = \begin{bmatrix} -5 \\ 4 \end{bmatrix}, \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \text{ and } \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The distance between pairs $(\mathbf{x}_1, \mathbf{y})$ and $(\mathbf{x}_2, \mathbf{y})$ using L_1 norms is:

$$||\mathbf{y} - \mathbf{x}_1||_1 = \sum_{i=1}^{2} |y_i - x_{1i}| = |(-5) - (-2)| + |(4) - (0)| = |-3| + |4| = 7$$
$$||\mathbf{y} - \mathbf{x}_2||_1 = \sum_{i=1}^{2} |y_i - x_{2i}| = |(-5) - (-1)| + |(4) - (2)| = |-4| + |2| = 6$$

Similarly, we calculate the L_2 norm of the differences $\mathbf{y} - \mathbf{x}_1$ and $\mathbf{y} - \mathbf{x}_2$.

$$\begin{aligned} ||\mathbf{y} - \mathbf{x}_1||_2 &= (\textstyle\sum_{i=1}^{2} |y_i - x_{1i}|^2)^{\frac{1}{2}} \\ &= \sqrt{(-5) - (-2))^2 + ((4) - (0))^2} \\ &= \sqrt{(-3)^2 + (4)^2} \\ &= \sqrt{25} = 5 \end{aligned}$$

$$\begin{aligned} ||\mathbf{y} - \mathbf{x}_2||_2 &= (\textstyle\sum_{i=1}^{2} |y_i - x_{2i}|^2)^{\frac{1}{2}} \\ &= \sqrt{(-5) - (-1))^2 + ((4) - (2))^2} \\ &= \sqrt{(-4)^2 + (2)^2} \\ &= \sqrt{20} = 2\sqrt{5} \end{aligned}$$

Let $\mathcal{P}$ be a matrix as shown below

$$\mathcal{P} = \frac{1}{\sqrt{13}} \begin{bmatrix} 2 & 3 \\ -3 & 2 \end{bmatrix}.$$

It is easy to see that the columns $\mathbf{p}_1$ and $\mathbf{p}_2$ of $\mathcal{P}$ form an orthonormal basis.

$$< \mathbf{p}_1, \mathbf{p}_2 >= \begin{bmatrix} 2 & -3 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = 0$$

and

$$< \mathbf{p}_i, \mathbf{p}_i >= \frac{1}{13}(2^2 + (-3)^2) = 1, \text{ for } i = 1, 2$$

We project all three vectors $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{y}$ onto this orthonormal basis $\mathcal{P}$ to obtain vectors $\mathbf{x}_1^p$, $\mathbf{x}_2^p$ and $\mathbf{y}^p$ respectively

$$\mathbf{x}_1^p = (\mathcal{P}^T)(x_1) = \frac{1}{\sqrt{13}} \begin{bmatrix} 2 & -3 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} -2 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{13}} \begin{bmatrix} -4 \\ -6 \end{bmatrix}$$

$$\mathbf{x}_2^p = (\mathcal{P}^T)(x_2) = \frac{1}{\sqrt{13}} \begin{bmatrix} 2 & -3 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \frac{1}{\sqrt{13}} \begin{bmatrix} -8 \\ 1 \end{bmatrix}$$

$$\mathbf{y}^p = (\mathcal{P}^T)(y) = \frac{1}{\sqrt{13}} \begin{bmatrix} 2 & -3 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} -5 \\ 4 \end{bmatrix} = \frac{1}{\sqrt{13}} \begin{bmatrix} -22 \\ -7 \end{bmatrix}.$$

We then calculate the differences between vector $\mathbf{y}^p$ and vectors $\mathbf{x}_1^p$ and $\mathbf{x}_2^p$ respectively:

$$\mathbf{y}^p - \mathbf{x}_1^p = \frac{1}{\sqrt{13}} \begin{bmatrix} -18 \\ -1 \end{bmatrix} \text{ and } \mathbf{y}^p - \mathbf{x}_2^p = \frac{1}{\sqrt{13}} \begin{bmatrix} -14 \\ -8 \end{bmatrix}$$

Finally, we calculate the L_1-distances between vector $\mathbf{y}^p$ and vectors $\mathbf{x}_1^p$ and $\mathbf{x}_2^p$

$$||\mathbf{y}^p - \mathbf{x}_1^p||_1 = \sum_{i=1}^{2} |y_i^p - x_{1i}^p| = \frac{1}{\sqrt{13}}(|-18| + |-1|) = \frac{19}{\sqrt{13}}$$

$$||\mathbf{y}^p - \mathbf{x}_2^p||_1 = \sum_{i=1}^{2} |y_i^p - x_{2i}^p| = \frac{1}{\sqrt{13}}(|-14| + |-8|) = \frac{22}{\sqrt{13}}$$

and similarly we compute their L_2-distances

$$\begin{aligned}
||\mathbf{y}^p - \mathbf{x}_1^p||_2 &= (\textstyle\sum_{i=1}^{2} |y_i^p - x_1^p i|^2)^{\frac{1}{2}} \\
&= \tfrac{1}{\sqrt{13}}\sqrt{(-18)^2 + (-1)^2} \\
&= \tfrac{1}{\sqrt{13}}\sqrt{325} \\
&= \sqrt{25} = 5 \\
||\mathbf{y}^p - \mathbf{x}_1^p||_2 &= (\textstyle\sum_{i=1}^{2} |y_i^p - x_1^p i|^2)^{\frac{1}{2}} \\
&= \tfrac{1}{\sqrt{13}}\sqrt{(-14)^2 + (-8)^2} \\
&= \tfrac{1}{\sqrt{13}}\sqrt{260} \\
&= \sqrt{20} = 2\sqrt{5}
\end{aligned}$$

This example shows that for this specific case the L_2-distance between two vectors is invariant under $\mathcal{P}$, the change of orthonormal basis.

$$||\mathbf{y} - \mathbf{x}_1||_2 = 5 = ||\mathbf{y}^p - \mathbf{x}_1^p||_2$$
$$||\mathbf{y} - \mathbf{x}_2||_2 = 2\sqrt{5} = ||\mathbf{y}^p - \mathbf{x}_2^p||_2$$

In fact, according to Theorem 4.3.2, that L_2 norm preserves distances between two vectors projected into any subspace spanned by an orthonormal basis.

However, this is not true for L1 norm. We calculate the L_1 norm of $\mathbf{y}^p - \mathbf{x}_1^p$ and $\mathbf{y}^p - \mathbf{x}_2^p$ respectively:

$$||\mathbf{y} - \mathbf{x}_1||_1 = 7 \neq \frac{19}{\sqrt{13}} = ||\mathbf{y}^p - \mathbf{x}_1^p||_1$$

$$||\mathbf{y} - \mathbf{x}_2||_1 = 6 \neq \frac{22}{\sqrt{13}}||\mathbf{y}^p - \mathbf{x}_2^p||_1$$

Moreover, order of distances is not preserved:

$$||\mathbf{y} - \mathbf{x}_1||_1 = 7 > 6 = ||\mathbf{y} - \mathbf{x}_2||_1, \text{ but}$$
$$||\mathbf{y}^p - \mathbf{x}_1^p||_1 = \frac{19}{\sqrt{13}} < \frac{22}{\sqrt{13}} = ||\mathbf{y}^p - \mathbf{x}_2^p||_1$$

This counter-example not only shows that L_1 norm doesn't preserve distances nor the order of distance under the change of orthonormal basis. But preserving the distance order among elements in a sequence is important in our algorithm because we seek to find the block which is the "closest" to the block of interest.

Counter-Example 4.3.3 - Specific Counter-Example for L_1-Distance Preservation With Real MR Data:

In this example we consider data from real MR images. We select the "block of interest" $\mathbf{b}$ of size 2×2 from the test image:

$$\mathbf{b} = \begin{bmatrix} 116 & 123 \\ 128 & 120 \end{bmatrix}$$

The inquiry block $\mathbf{B}$ in the reference image is a block of pixels containing the blocks that are in a neighborhood of the position of the block of interest in the reference image. For this counter-example we consider the inquiry block of size 4×4 so that we can show our arguments explicitly with all the dictionary atoms inside the inquire block. The inquiry block $\mathbf{B}$ is:

$$\mathbf{B} = \begin{bmatrix} 114 & 125 & 130 & 117 \\ 111 & 115 & 136 & 114 \\ 115 & 130 & 148 & 139 \\ 135 & 133 & 151 & 167 \end{bmatrix}$$

As similarity measures we use both L_1 and L_2 norms. We create a dictionary Φ by extracting $\mathbf{a}_k$ blocks from inquiry block of the same size as the "block of interest", $k = 1, 2, ...16$. Blocks $\mathbf{a}_k$ serve as the training blocks. Both $\mathbf{b}$ and $\mathbf{a}_k$ blocks are stacked into column vectors, denoted as $\mathbf{y}$ and $\mathbf{x}_k$ respectively and are projected into eigen-subspace which is spanned by the eigenvectors of the dictionary atoms, namely the

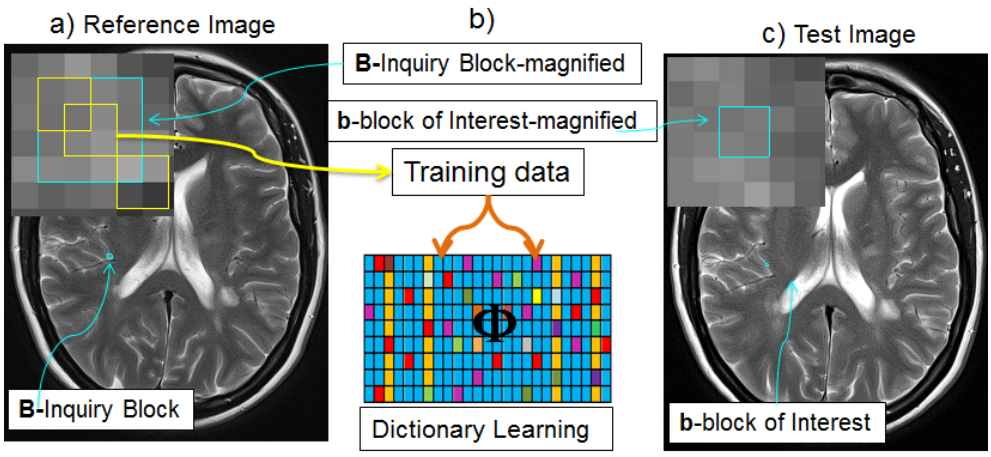

Figure 4.3.2: a) the reference image with the "Inquiry Block" $\mathbf{B}$ in cyan magnified and overlapping training blocks $\mathbf{a}_k$ in yellow. b) dictionary Φ formed by stacking $\mathbf{a}_k$ training blocks into column vectors. c) the test image showing the "block of interest $\mathbf{b}$" enlarged in cyan color.

training blocks shown in Figs. 4.3.2, 4.3.3 .

$$\mathbf{y} = \begin{bmatrix} 116 \\ 128 \\ 123 \\ 120 \end{bmatrix}$$

The dictionary $\mathbf{\Phi}$ is:

$$\mathbf{\Phi} = \begin{bmatrix} 114 & 125 & 130 & 117 & 111 & 115 & 136 & 114 & 115 & 130 & 148 & 139 & 135 & 133 & 151 & 167 \\ 111 & 115 & 136 & 114 & 115 & 130 & 148 & 139 & 135 & 133 & 151 & 167 & 95 & 104 & 113 & 77 \\ 125 & 130 & 117 & 93 & 115 & 136 & 114 & 100 & 130 & 148 & 139 & 133 & 133 & 151 & 167 & 148 \\ 115 & 136 & 114 & 100 & 130 & 148 & 139 & 133 & 133 & 151 & 167 & 148 & 104 & 113 & 77 & 57 \end{bmatrix}$$

We center columns of $\mathbf{\Phi}$ and $\mathbf{y}$ by subtracting the mean μ of the dictionary $\mathbf{\Phi}$. The $\mathbf{\Phi}^c$ is the centered dictionary with zero mean. In this example μ is 126.6719 and $\mathbf{\Phi}^c$ is:

$$\mathbf{\Phi}^c = \begin{bmatrix} -12.67 & -1.67 & 3.33 & -9.67 & -15.67 & -11.67 & 9.33 & -12.67 & -11.67 & 3.33 & 21.33 & 12.33 & 8.33 & 6.33 & 24.33 & 40.33 \\ -15.67 & -11.67 & 9.33 & -12.67 & -11.67 & 3.33 & 21.33 & 12.33 & 8.33 & 6.33 & 24.33 & 40.33 & -31.67 & -22.67 & -13.67 & -49.67 \\ -1.67 & 3.33 & -9.67 & -33.67 & -11.67 & 9.33 & -12.67 & -26.67 & 3.33 & 21.33 & 12.33 & 6.33 & 6.33 & 24.33 & 40.33 & 21.33 \\ -11.67 & 9.33 & -12.67 & -26.67 & 3.33 & 21.33 & 12.33 & 6.33 & 6.33 & 24.33 & 40.33 & 21.33 & -22.67 & -13.67 & -49.67 & -69.67 \end{bmatrix}$$

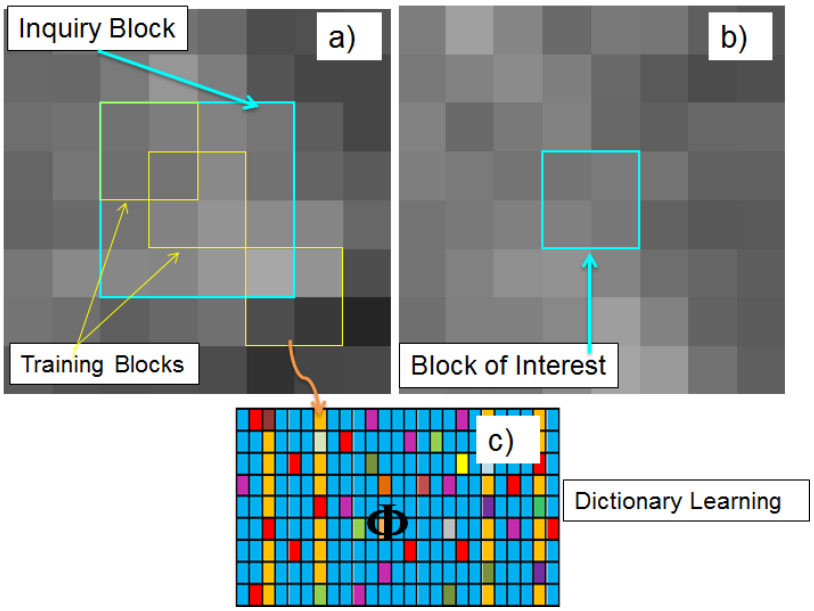

Figure 4.3.3: a) magnified "Inquiry Block" from the reference image $\mathbf{B}$, in cyan color with many overlapping training blocks $\mathbf{a}_k$ in yellow, $k = 1, 2, ..., 16$. b) magnified "block of interest" from the test image $\mathbf{b}$ in cyan color. c) the dictionary $\mathbf{\Phi}$ formed by stacking training blocks $\mathbf{a}_k$ into column vectors.

Then we calculate the covariance matrix $\mathbf{\Omega}$ of $\mathbf{\Phi}^c$ as:

$$\mathbf{\Omega} = \mathbf{\Phi}^c \left(\mathbf{\Phi}^c\right)^T \tag{4.3.0.4}$$

$\mathbf{\Omega}$ for this counter-example is:

$$\mathbf{\Omega} = \begin{bmatrix} 3979 & -1245 & 3024 & -3082 \\ -1245 & 7883 & -1753 & 7883 \\ 3024 & -1753 & 5708 & -1887 \\ -3082 & 7883 & -1887 & 12489 \end{bmatrix}$$

The matrix of the eigenvectors $\mathbf{U}$ and their corresponding eigenvalues λ are also calculated.

$$\mathbf{U} = \begin{bmatrix} -0.2351 & 0.5244 & 0.4981 & 0.6494 \\ 0.5612 & 0.2293 & 0.6396 & -0.4726 \\ -0.2223 & 0.7891 & -0.3945 & -0.4151 \\ 0.7618 & 0.2231 & -0.4326 & 0.4274 \end{bmatrix}$$

We keep only the eigenvectors corresponding to the non-zeros eigenvalues $\lambda = \{19798, 6674, 2663, 924\}$. We compute $\widehat{\mathbf{y}}$ and $\widehat{\mathbf{\Phi}}$, the projections of the zero-mean "block

of interest" $\mathbf{y}^c$ and the dictionary $\mathbf{\Phi}^c$ into the eigen-subspace.

$$\widehat{\mathbf{\Phi}} = \mathbf{U}^T \, \mathbf{\Phi}^c \tag{4.3.0.5}$$

$$\widehat{\mathbf{\Phi}} = \begin{bmatrix} -14.34 & 0.21 & -3.05 & -17.67 & 2.27 & 18.79 & 21.99 & 20.65 & 11.50 & 16.57 & 36.62 & 34.58 & -38.41 & -30.04 & -60.20 & -95.17 \\ -14.16 & 1.15 & -6.58 & -40.50 & -19.36 & 6.76 & 2.53 & -23.45 & -0.17 & 25.45 & 35.49 & 25.46 & -2.96 & 14.27 & 30.36 & 11.04 \\ -10.63 & -13.65 & 16.92 & 11.90 & -12.11 & -16.59 & 17.95 & 9.36 & -4.54 & -13.23 & 3.87 & 20.21 & -8.80 & -15.03 & 8.95 & 10.05 \\ -5.12 & 7.04 & -3.65 & 2.28 & 1.61 & -3.91 & 6.51 & -0.28 & -10.19 & 0.72 & 14.47 & -4.56 & 8.06 & -1.12 & -15.71 & 11.03 \end{bmatrix}$$

$$\widehat{\mathbf{y}} = \begin{bmatrix} -1.0125 \\ -9.6777 \\ -0.1310 \\ -8.8853 \end{bmatrix}$$

Finally, we calculate distances using L_1- and L_2- distances between pairs $\mathbf{y}$ and each column $\mathbf{x}_k$ of the dictionary $\mathbf{\Phi}$. Similarly, we calculate distances between the corresponding projected pairs into the eigen-space, i.e., the distances between $\widehat{\mathbf{y}}$ and each column vector $\widehat{\mathbf{x}}_k$ of the dictionary $\widehat{\Phi}$. We are interested in examining the minimum values and ordering of the L_1- and L_2- distances preserved in physical and eigen-spaces under orthonormal projections. We compute:

$$\min_k ||\mathbf{y} - \mathbf{x}_k||_1, \;\; \min_k ||\widehat{\mathbf{y}} - \widehat{\mathbf{x}}_k||_1, \;\; \min_k ||\mathbf{y} - \mathbf{x}_k||_2, \;\; \min_k ||\widehat{\mathbf{y}} - \widehat{\mathbf{x}}_k||_2$$

for $k = 1, 2, \cdots, 16$.

Table. 4.3.0.1 below shows the result of these distances.

Table 4.3.0.1: Results of L_1 and L_2 norms in both spatial and eigenspace domains

Norm / k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\|\|\mathbf{y} - \mathbf{x}_k\|\|_1$	26.0	45.0	34.0	65.0	36.0	44.0	68.0	49.0	28.0	75.0	118.0	100.0	78.0	76.0	137.0	190.0
$\|\|\widehat{\mathbf{y}} - \widehat{\mathbf{x}}_k\|\|_1$	32.1	41.5	27.4	70.7	35.4	57.7	68.7	53.5	27.7	75.4	110.2	95.4	69.7	75.6	115.1	145.0
$\|\|\mathbf{y} - \mathbf{x}_k\|\|_2$	17.9	23.6	18.2	38.7	18.9	31.0	35.2	28.7	**16.4**	42.5	63.4	54.2	42.5	41.2	72.4	99.0
$\|\|\widehat{\mathbf{y}} - \widehat{\mathbf{x}}_k\|\|_2$	17.9	23.6	18.2	38.7	18.9	31.0	35.2	28.7	16.4	42.5	63.4	54.2	42.5	41.2	72.4	99.0

The minimum for each row in Table 4.3.0.1 is shown in red. The last two rows are identical showing that the distances between the "block of interest" and each training block from the "inquiry block" in the reference images are equal both in image and in the eigenspace domains.

We further observe that in both the image domain and the eigenspace L_2 minimiza-

tion is achieved at the ninth atom $k = 9$ for both original dictionary and its projection and the minimum distance is 16.4. Thus, the L_2 norm produces the same value on a pair of un-projected vectors and on a pair of projected vectors into the eigenspace.

Table 4.3.0.1 also shows that this doesn't hold for L_1 norm. The first two rows of the table show that L_1 norm do not produce the same values for pairs of vectors in image domain and their projections into the eigenspace. Moreover, it is easy to observe that L_1 norm doesn't preserve the order. We consider vector pairs shown in blue colour in Table 4.3.0.1 and compare the results. Note that

$$45.0 = ||\mathbf{y} - \mathbf{x}_2||_1 > ||\mathbf{y} - \mathbf{x}_6||_1 = 44.0$$

but

$$41.5 = ||\tilde{\mathbf{y}} - \tilde{\mathbf{x}}_2||_1 < ||\tilde{\mathbf{y}} - \tilde{\mathbf{x}}_6||_1 = 57.7$$

Similarly, we consider pairs in green colour and compare the results, i.e.,

$$78.0 = ||\mathbf{y} - \mathbf{x}_{13}||_1 > ||\mathbf{y} - \mathbf{x}_{14}||_1 = 76.0$$

but

$$69.7 = ||\tilde{\mathbf{y}} - \tilde{\mathbf{x}}_{13}||_1 < ||\tilde{\mathbf{y}} - \tilde{\mathbf{x}}_{14}||_1 = 75.6$$

We display these results visually in Fig 4.3.4.

The choice of metric is important when determining which block from the dictionary $\mathbf{\Phi}$ is the "closest" to the block of interest $\mathbf{b}$. For instance, in our counterexample the minimum of L_2-distances between block $\mathbf{b}$ and each block $\mathbf{a}_k$ from the dictionary $\mathbf{\Phi}$, $k = 1, 2, ..., 16$, is reached for $k = 9$, in both spatial and in the eigenspace domain. As shown in Fig. 4.3.4, the block outlined in yellow is the "closest" to block $\mathbf{b}$ in spatial domain consistent with the block outlined in red which is the "closest" to block $\mathbf{b}$ in the eigenspace domain.

On the other hand the minimum distance via L_1-distance in spatial domain between each block of the dictionary $\mathbf{\Phi}$ and the block of interest $\mathbf{b}$ is reached for $k = 1$, corresponding to the block with green border. However Fig. 4.3.4 shows that the

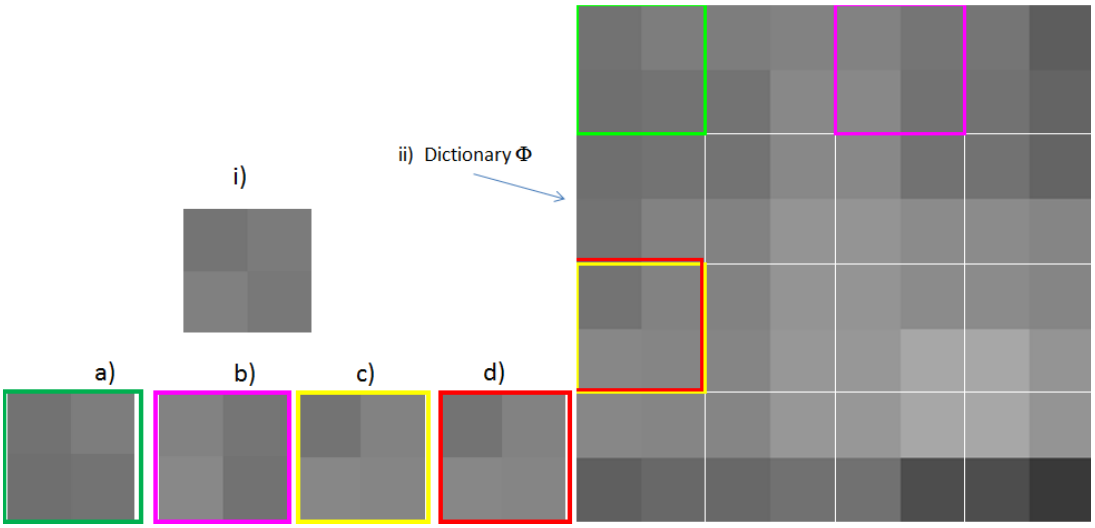

Figure 4.3.4: i) The block of interest $\mathbf{b}$ from the test image. ii) dictionary $\mathbf{\Phi}$ formed with training blocks $\mathbf{a}_k$ chosen from inquiry block in reference image. a) the block from $\mathbf{\Phi}$ which is the "closest" to $\mathbf{b}$ via L_1 norm in spatial domain. b) the block from $\mathbf{\Phi}$ which is the "closest" to $\mathbf{b}$ via L_1 norm in the eigenspace domain. c) and d) blocks from $\mathbf{\Phi}$ which are the "closest" to $\mathbf{b}$ via L_2 norm in both spatial and in the eigenspace domain respectively.

minimum L_1-distance in eigenspace is achieved at $k = 3$ instead of $k = 1$, i.e., the 3^{rd} block of $\mathbf{\Phi}$ outlined in magenta.

Counter-Example.4.3.4 - Real MR Data:

Furthermore, we experiment with blocks of bigger sizes as shown in Fig 4.3.5 in order to observe whether the statements above hold.

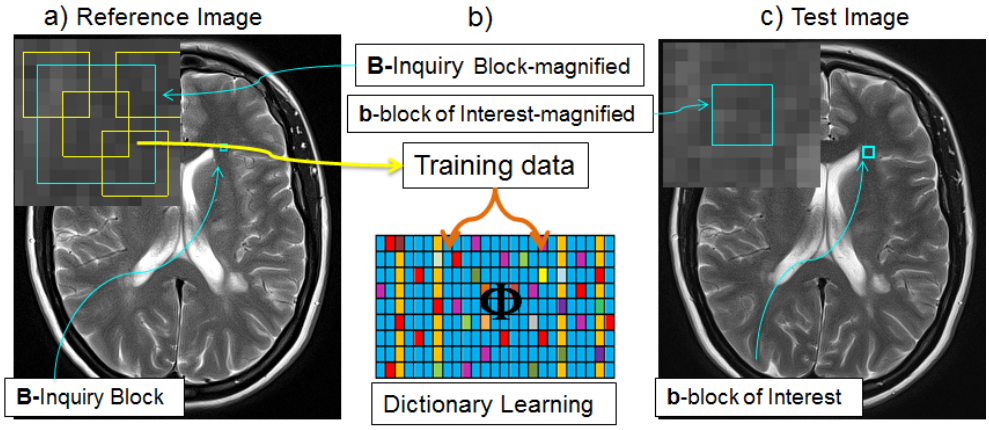

Figure 4.3.5: a) The reference image with the "inquiry block" magnified $\mathbf{B}$ in cyan and overlaping training blocks $\mathbf{a}_k$ in yellow. b) Dictionary $\mathbf{\Phi}$ formed by stacking $\mathbf{a}_k$ training blocks into column vectors. c) Test image showing the enlarged "block of interest" $\mathbf{b}$ highlighted in cyan.

In this counterexample the "block of interest" from the test image $\mathbf{b}$ is of size 5×5. The inquiry block $\mathbf{B}$ from the reference image is of size 9×9. All $\mathbf{a}_k$ blocks are

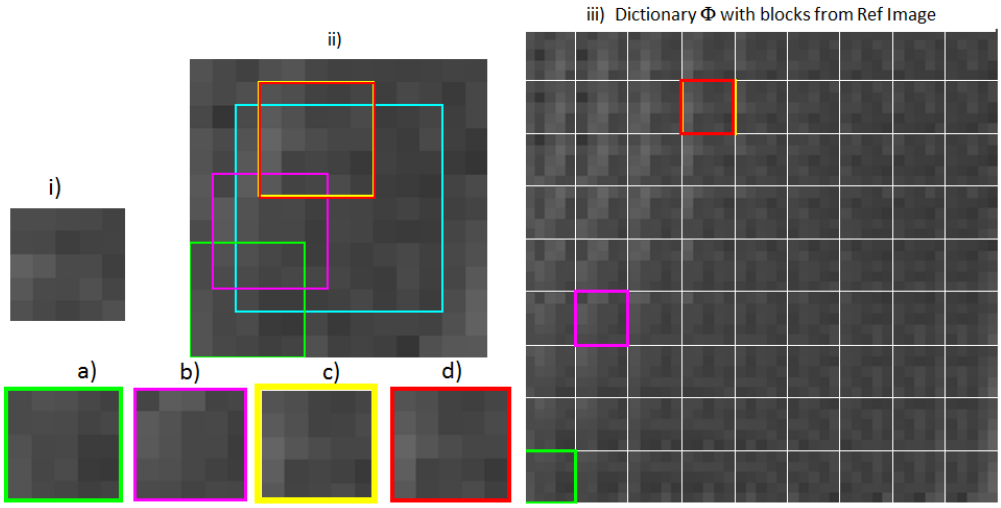

Figure 4.3.6: i) The block of interest **b** from the test image. ii) Inquiry with $\mathbf{a}_k$ blocks determined from L_1 and L_2 minimization. iii) Dictionary $\mathbf{\Phi}$ formed with blocks $\mathbf{a}_k$ chosen from inquiry block in reference image. Blocks in $\mathbf{\Phi}$, bordered in green, magenta, yellow and red correspond to blocks chosen as "closest" to **b** via L_1 and L_2 in spatial and in eigenspace domain respectively. a) the block from $\mathbf{\Phi}$ which is the "closest" to **b** via L_1 norm in spatial domain. b) the block from $\mathbf{\Phi}$ which is the "closest" to **b** via L_1 norm in the eigenspace domain. c) and d) blocks from $\mathbf{\Phi}$ which are the "closest" to **b** via L_2 norm in both spatial and in the eigenspace domain respectively.

extracted from the inquiry block and are of the same size as the "block of interest". The minimum of L_2-distance between block **b** and each block $\mathbf{a}_k$ from the dictionary $\mathbf{\Phi}$, $k = 1, 2, ..., 81$, is reached for $k = 13$ in both spatial and in the eigenspace domain. As it is shown in Fig 4.3.6, the block bordered in yellow is the "closest" to block **b** in spatial domain which is consistent with the block outlined in red as the "closest" to block **b** in eigenspace domain.

On the other hand, the minimum distance via L_1 norm in spatial domain between each block of the dictionary $\mathbf{\Phi}$ and the block of interest **b** is reached for $k = 73$, the block outlined in green. In the eigenspace domain, the minimum via L_1 norm isn't reached for $k = 73$, but for $k = 47$ instead as shown in Fig 4.3.6, which corresponds to the 47^{th} block of $\mathbf{\Phi}$ outlined in magenta. This is not consistent with the best matching block, the 73^{rd}, as identified in spatial domain.

Counter-Example 4.3.5 -Preserving Structure With Real MR Data:

The next logical question to ask is: does the inconsistency of the L_1 minimization in different spaces matter visually? What retains the visual similarity between the best matching blocks in different spaces, the L_1- or L_2-minimization? We experiment with

blocks of larger sizes to observe whether the structure between the "inquiry block" and the block chosen from $\mathbf{\Phi}$ via L_1 and L_2 norms in both spatial and eigensapce domain is preserved as Figs. 4.3.8 and 4.3.9 show.

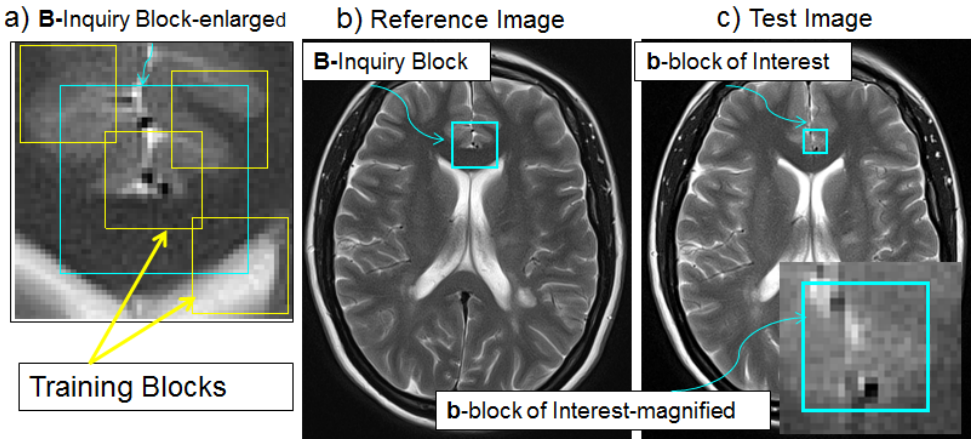

Figure 4.3.7: a) Inquiry block magnified with many overlapping training blocks $\mathbf{a}_k$. b) Reference image with "inquiry block" in cyan. c) Test image with "block of interest" in cyan. A magnified version of the block of interest is placed at the bottom right

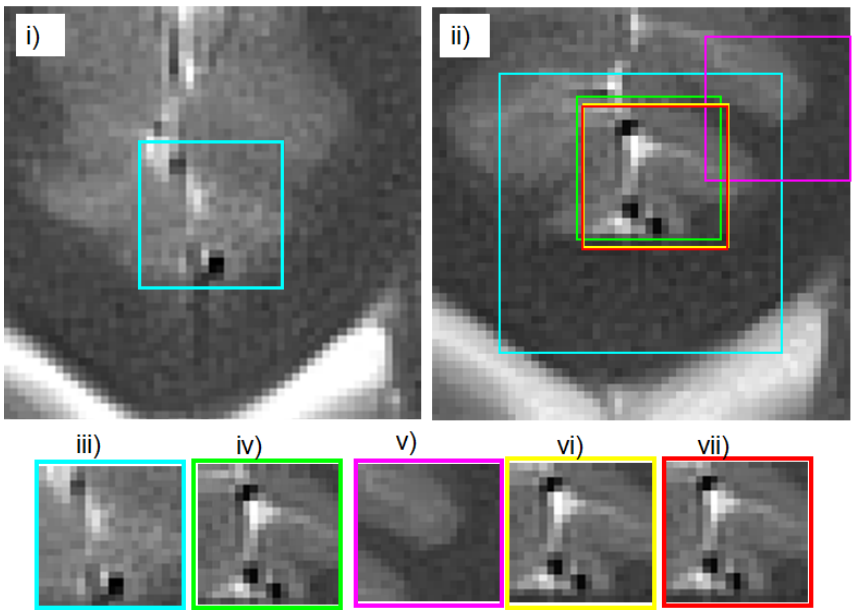

Figure 4.3.8: i) the block of interest $\mathbf{b}$ outlined in cyan from the test image. ii) Inquiry block magnified with the best matching blocks determined from L_1 and L_2 minimization in spatial and eigenspace domains. iii) The block of interest. iv) The block from $\mathbf{\Phi}$ which is the "closest" to $\mathbf{b}$ via L_1 norm in spatial domain. v) The block from $\mathbf{\Phi}$ which is the "closest" to $\mathbf{b}$ via L_1 norm in the eigenspace domain. vi) and vii) The blocks from $\mathbf{\Phi}$ which are the "closest" to $\mathbf{b}$ via L_2 norm in both spatial and eigenspace domains.

In this example the size of "block of interest" and of all the training blocks is 19×19, and the "inquiry block" is of size 37×37. We know that L_2 norm is invariant

from the change of basis. We also know that L_1 norm doesn't preserve the distances and does not preserve the minimum. We get the best matching block in spatial and in eigenspace domain when L_2 norm is used, but may result in two different blocks if L_1 norm is used. The question is: How similar are these two blocks with each other and how similar is each one to the "block of interest"? Figs. 4.3.7, 4.3.8 and 4.3.9 show that the blocks determined via L_2 minimization in spatial and in eigenspace domain have similar structure to the block of interest. The block determined by L_1 minimization norm in eigenspace domain, outlined in magenta, looks very different in structure from the one determined from L_1 minimization in spatial domain, outlined in green. Moreover, the block determined by L_1 minimization norm in eigenspace domain looks different from the two blocks determined by L_2 minimization in both spatial and eigenspace domains, outlined in yellow and red respectively. It also looks very different in structure from the "block of interest" outlined in cyan.

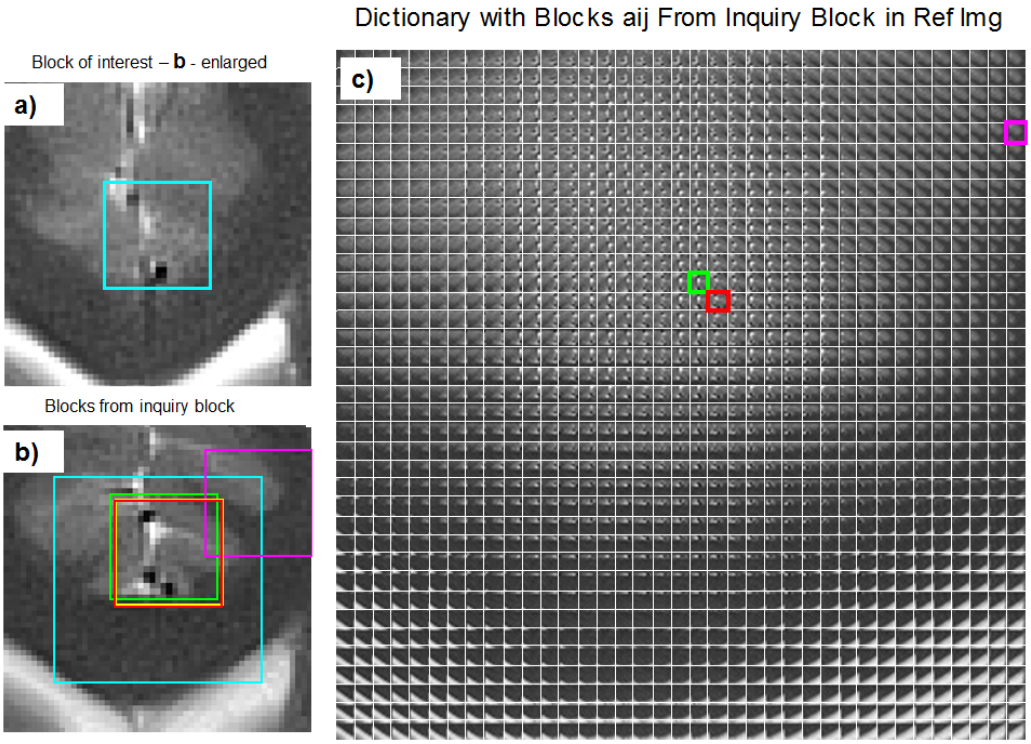

Figure 4.3.9: a) The block of interest **b** enlarged from the test image and outlined in cyan. b) Inquiry block magnified with the best matching blocks determined from L_1 and L_2 minimization in different domains. c) Dictionary $\mathbf{\Phi}$ comprised of blocks $\mathbf{a}_k$ chosen from inquiry block in reference image. Blocks of $\mathbf{\Phi}$ outlined in green, magenta, yellow and red correspond to blocks chosen as "closest" to **b** via L_1 and L_2 minimizations in spatial and in eigenspace domain respectively.

In summary, in this section we discussed the effects of L_1 and L_2 norms in sub-

space learning and showed theoretically and experimentally that L_2 norm preserves the distance and the order under an orthonormal change of bases. We also presented counterexamples to prove that L_1 norm does not satisfy such properties.

4.4 The EigenBlockCD Algorithm with L_2 Minimization

In this section we describe the EigenBlock Change Detection (EigenBlockCD) algorithm for detecting changes between two consecutive fully sampled medical images. We use the same notations and make the same assumptions as in Chapter 3.

As described in Chapter 3, the AEDL algorithm uses the knowledge of sparsity and compressed sensing to reconstruct the background of each block of interest using L_1 minimization algorithms. This makes the AEDL algorithm computationally expensive.

We reiterate here that in the AEDL algorithm a block $\mathbf{b}$ from the test image is sparsely represented as a linear combination of a few $\mathbf{a}_k$ blocks from the reference image over a dictionary $\mathbf{\Phi}$, i.e.,

$$\mathbf{b} = \sum_{k=1}^{(2\Delta+1)^2} \boldsymbol{\gamma}_k \mathbf{a}_k = \sum_{k=1}^{n} \boldsymbol{\gamma}_k \mathbf{a}_k, \quad \mathbf{a}_k \in \mathcal{B} \tag{4.4.0.6}$$

where $k \in \{1, 2, \ldots, (2\Delta + 1)^2\}$

Blocks $\mathbf{a}_k$ and $\mathbf{b}$ are stacked as column vectors $\mathbf{x}_k$ and $\mathbf{y}$ respectively of sizes $m \times 1$, where $m = (2\delta + 1)^2$ and δ is a positive integer representing the radius of these blocks. Then the Eq. (4.4.0.6) can be written as:

$$\mathbf{y} = \sum_{k=1}^{n} \boldsymbol{\gamma}_k \mathbf{x}_k, \quad k \in \{1, 2, \ldots, n\} \tag{4.4.0.7}$$

To account for noise Eq. (4.4.0.7) in matrix format can be written:

$$\mathbf{y} = \mathbf{\Phi}\boldsymbol{\gamma} + \mathbf{n} \tag{4.4.0.8}$$

The AEDL algorithm as described in Section 3 finds the sparsest vector $\boldsymbol{\gamma}$ that

satisfies Eq.(4.4.0.8) by solving the L_1 minimization problem.

$$\boldsymbol{\gamma}^* = \arg\min_{\boldsymbol{\gamma}} ||\gamma||_1 \;\; s.t. \;\; \mathbf{y} = \boldsymbol{\Phi}\gamma + \mathbf{n} \tag{4.4.0.9}$$

By spreading out the mean of block $\mathbf{B}$ we solved L_1 minimization problem:

$$\text{Problem } P_1 : \quad \begin{aligned} &\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} ||\boldsymbol{\alpha}||_1 \\ &\text{subject to:} \quad \mathbf{y} = \mathbf{A}\boldsymbol{\alpha} \end{aligned} \tag{4.4.0.10}$$

where: $\mathbf{A} = [\boldsymbol{\Phi}\ \mathbf{I}]$, $\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\gamma} \\ \mathbf{n} \end{bmatrix}$ and $\boldsymbol{\alpha}^* = \begin{bmatrix} \boldsymbol{\gamma}^* \\ \mathbf{n}^* \end{bmatrix}$

The background $\mathbf{y}^*$ of $\mathbf{y}$ is then computed as $\mathbf{y}^* = \boldsymbol{\Phi}\boldsymbol{\gamma}^*$.

To reduce the running time, the EigenBlockCD algorithm seeks to find the best matching block to the block of interest from the dictionary atoms instead of representing it as a linear combination of the dictionary atoms as in the AEDL algorithm. In fact, it derives from the assumptions that each block $\mathbf{b}$ from the test image its closest block in the reference image will be in some neighbourhood of $\mathbf{a}_k$ named the inquiry block $\mathbf{B}$, i.e., it will be one of the atoms of the dictionary. The block of interest with some structure either is one of the atoms of the dictionary (when there are no changes between the two consecutive times) or close to one of the atoms (when there are changes).

To reduce the redundancy among columns of the dictionary and therefore the dimensionality of the dictionary and to help extract the features, as in the AEDL algorithm, the EigenBlockCD algorithm projects columns of the dictionary onto the eigen-subspace, which explains the choice of the L_2 norm as the similarity measure, i.e., projections onto the eigen-subspace preserves the L_2-distances and the L_2-distance orders.

The EigenBlockCD algorithm consists of three stages as illustrated in Fig. 4.4.1. During Stage 1 we use the reference image to learn the dictionary. The algorithm is a block based dictionary which captures only local disease related changes and ig-

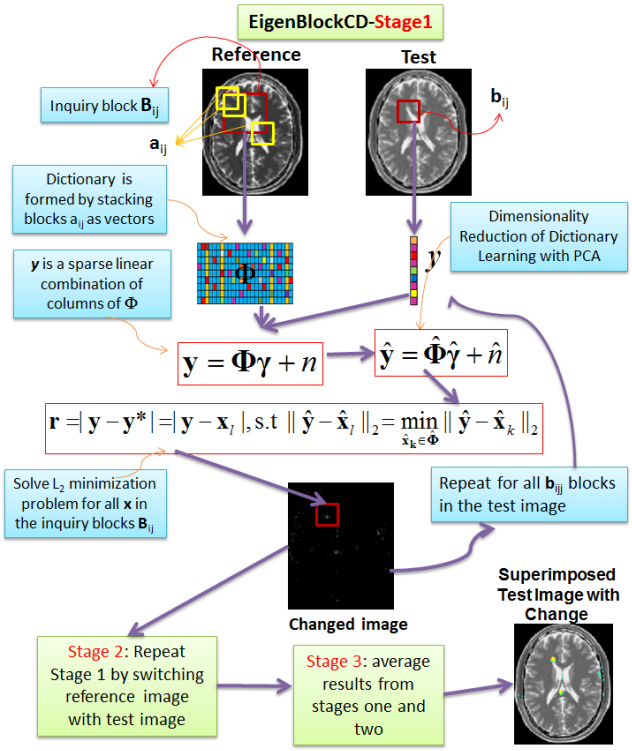

Figure 4.4.1: Three stages of the EigenBlockCD algorithm

nores the changes due to the patient positioning. We use overlapping blocks to reduce image block artifacts. Columns of the dictionary are obtained by projecting blocks from reference image onto the eigen - subspace. In our algorithm for each block $\mathbf{b}$ in the test image we consider an inquiry block $\mathbf{B}$ in the reference image $\mathbf{I}_1$ of size $(2\Delta +1)\times(2\Delta+1)$ and centered at a pixel (i, j), where Δ is a positive integer.

The EigenBlockCD algorithm models the background $\mathbf{y}^*$ of $\mathbf{y}$ by finding the best matching block to the block of interest from all dictionary blocks [24]. This is done by minimizing the L_2 norm of the absolute differences between the block of interest and its best matching block found or by solving the minimization problem:

$$\mathbf{e}_{min} = \min_{x_k \in \mathbf{\Phi}} ||\mathbf{y} - \mathbf{x}_k||_2 \tag{4.4.0.11}$$

Let us consider again the dictionary $\mathbf{\Phi}$:

$$\mathbf{\Phi} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \tag{4.4.0.12}$$

As in Chapter 3, dictionary $\mathbf{\Phi}$ is centered by subtracting the mean and both the

zero mean dictionary $\mathbf{\Phi}^c$ and vector $\mathbf{y}^c$ are projected onto the subspace obtained by the eigenvectors of the covariance matrix $\mathbf{\Omega} = \mathbf{\Phi}^c\left(\mathbf{\Phi}^c\right)^T$.

$$\widehat{\mathbf{\Phi}} = \mathbf{U}_r^T\ \Phi^c \quad \text{and} \quad \widehat{\mathbf{y}} = \mathbf{U}_r^T\ \mathbf{y}^c \tag{4.4.0.13}$$

where $\mathbf{U}_r$ is the matrix of the $\mathbf{r}$ eigenvectors corresponding to the r largest eigenvalues and r.

If the L_2 norm of the residual errors $\mathbf{e}_k$ between $\widehat{\mathbf{y}}$ and $\widehat{\mathbf{x}}_k$ are computed as $\mathbf{e}_k = ||\widehat{\mathbf{y}} - \widehat{\mathbf{x}}_k||_2$, then the minimum of all the errors will be:

$$\mathbf{e}_{min} = \min_{k\in\{1,2,\ldots,n\}}\{\mathbf{e}_k\} = \min_{\widehat{x}_k\in\widehat{\mathbf{\Phi}}}||\widehat{\mathbf{y}} - \widehat{\mathbf{x}}_k||_2 \tag{4.4.0.14}$$

where the $\widehat{x}_k$ are the columns of the projected dictionary $\widehat{\mathbf{\Phi}}$ in the eigen-subspace

$$\widehat{\mathbf{\Phi}} = [\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_k, \ldots, \widehat{\mathbf{x}}_n]$$

The background $\mathbf{y}^*$ of $\mathbf{y}$ is the best approximation to $\mathbf{y}$ in the reference image, that is, a vector $\mathbf{x}_l$ from dictionary $\mathbf{\Phi}$ that minimizes the residual errors, i.e.,

$$\mathbf{y}^* = \mathbf{x}_l \quad \text{such that:} \quad \mathbf{e}_l = ||\widehat{\mathbf{y}} - \widehat{\mathbf{x}}_l||_2 = \min_{\widehat{x}_k\in\widehat{\mathbf{\Phi}}}||\widehat{\mathbf{y}} - \widehat{\mathbf{x}}_k||_2 \tag{4.4.0.15}$$

where $\mathbf{x}_l \in \mathbf{\Phi}$. The residual of $\mathbf{y}$ with significant changes is computed as:

$$\mathbf{r} = |\mathbf{y} - \mathbf{y}^*| \tag{4.4.0.16}$$

Similar with our AEDL method in Chapter 3, the change image $\mathbf{I}_{cd1}$ is created by placing the residual blocks of $\mathbf{b}$ as blocks of size $\sqrt{m}\times\sqrt{m} = (2\delta+1)\times(2\delta+1)$, centered at the pixel (i,j) and divided by m, where m is the number of blocks $\mathbf{b}$ that contain that pixel. We repeat the process in Stage Two by switching the reference image with the test image and, finally, we compute the average of the two change images from

stages one and two. The final change image is:

$$\mathbf{I}_{cd} = \frac{1}{2m} \sum_{j=\delta}^{N_2-\delta} \sum_{i=\delta}^{N_1-\delta} \Big(\mathbf{r}^{[1]}(i,j) + \mathbf{r}^{[2]}(i,j) \Big) \tag{4.4.0.17}$$

where $\mathbf{r}^{[1]}(i,j)$ and $\mathbf{r}^{[2]}(i,j)$ represent the foreground of a block of interest calculated as in Eq. (4.4.0.16).

4.5 Numerical Results

4.5.1 Simulations with Synthetic Images

We first test the performance of the EigenblockCD using the same set of synthetic images that we used to test the AEDL algorithm. We compare the results to the ground truth and simple difference methods. More comprehensive performance analysis of the AEDL and the EigenblockCD will be provided in Chapter 6.

We test our method first by using synthetic images of size 129×128 as shown in Figs. 4.5.1 to 4.5.4. The reference image is created by stacking three strips of size 43×128 each. In each strip we placed three polygons with Gaussian distribution intensities 0.3, 0.2 and 0.4 respectively, from left to right. To test the performance of the algorithm in the presence of mis-alignment caused by shifts, rotations and other acquisition related artifacts caused by noise, the test image is created by including two types of changes to the reference image: significant and insignificant changes. As in Chapter 3, the significant changes are disease related and hence make up the ground truth.

The desired outcome of a change detection algorithm is to detect only the significant changes while rejecting the unimportant ones. The performance of the EigenBlockCD algorithm is compared with the ground truth and the absolute difference method.

In Fig. 4.5.1 strips of 43 rows of reference image are shifted 1-down and 1-left, 3-down and 2-right and 2- down and 3-left respectively which are considered insignificant changes. The EigenBlockCD algorithm ignores the changes related to horizontal and vertical translations of images' strips when the radius of the inquiry block $\mathbf{B}_{ij}$ is not

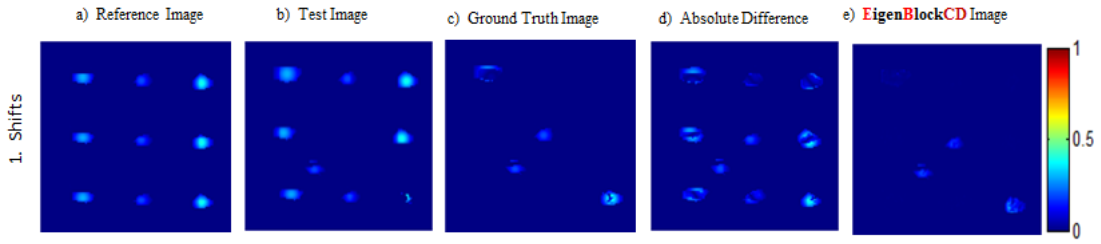

Figure 4.5.1: Testing the algorithm performance in the presence of object shifts. a) Reference image. b) Test image c) Ground truth image showing only the significant changes. d) Absolute difference image between the test and reference images. e) Change image obtained by the EigenBlockCD algorithm shows that the algorithm ignores the changes related to translations that can be captured by local dictionaries, but detects the real and significant changes closer to the ground truth.

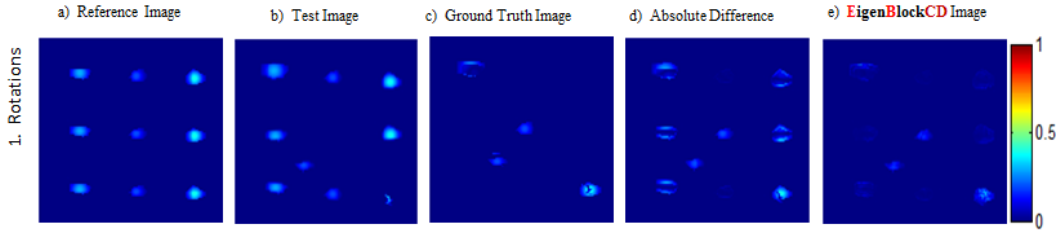

Figure 4.5.2: Testing the algorithm performance in the presence of object rotations. a) Reference image. b) Test image c) Ground truth image showing only the significant changes. d) Absolute difference image between the test and reference images. e) Change image obtained by the EigenBlockCD algorithm shows that our method ignores the changes related to rotations that can be captured by local dictionaries, but detects the significant changes closer to the ground truth.

less than the size of the shift, but detects the real and significant changes closer to the ground truth than the absolute difference method does, with $\delta = 1, c = 3$ and $\Delta = \delta + c = 4$.

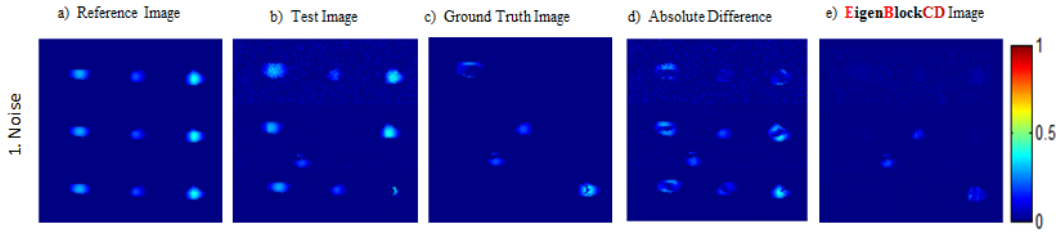

Figure 4.5.3: Testing the algorithm performance in the presence of different noise level. a) Reference image. b) Test image c) Ground truth image showing only the significant changes. d) Absolute difference image between the test and reference images. e) Change image obtained by the EigenBlockCD algorithm is very close to the ground truth and shows that our algorithm method detects the real changes for SNR ratio $>$ 25.

In Fig. 4.5.2 strips of 43 rows of reference image are rotated by 2 degrees clockwise, 1 degree counter-clockwise and the last 42 rows by 3 degrees counter-clockwis, which

are considered insignificant changes. Results show that the EigenBlockCD algorithm ignores the changes related to rotation of strips when the radius of the inquiry block $\mathbf{B}_{ij}$ not less then the angle of rotation and detects the significant changes closer to the ground truth than the absolute difference method does, with $\delta = 2, c = 3$ and $\Delta = \delta + c = 5$.

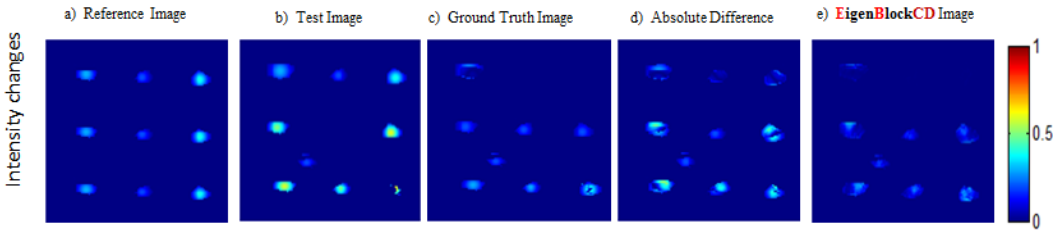

Figure 4.5.4: Testing the algorithm performance in the presence of different intensity level. a) Reference image. b) Test image. c) Ground truth image showing only the significant changes. d) Absolute difference image between the test and reference images. e) shows that the change image obtained by the EigenBlockCD algorithm detects the significant changes closer to the ground truth image.

In Fig. 4.5.3 Gaussian noise is added to the reference image with SNR = 30, 40, and 50 for the first, second and third 43 rows respectively which are considered insignificant changes. The simulations show that the EigenBlockCD algorithm still detects the real changes closer to the ground truth for $SNR \geq 30$ with $\delta = 1, c = 3$ and $\Delta = \delta + c = 4$.

In Fig. 4.5.4 starting from the second strip of 43 rows in the reference image, the intensities of shapes on the second and third strips are increased by 0.2, and 0.3 respectively. We want our algorithm to detect changes due to the intensity changes as they are considered disease related by the radiologists. Shapes at the bottom of the test image have a maximum intensity of 0.7. Therefore, we added these changes to our previous ground truth image which now serves as the ground truth for this experiment. The EigenBlockCD algorithm still detects the significant changes closer to the ground truth, with $\delta = 1, c = 1$ and $\Delta = \delta + c = 1$.

4.5.2 Applications to MR Images

We tested the EigenBlockCD algorithm with real MR images. Our results are confirmed by a radiologist and are shown in Figs. 4.5.5 to 4.5.8.

Fig. 4.5.5 shows the results of two real T2-weighted MR images taken in 2011

and 2013. The change image obtained by EigenBlockCD as shown in Fig. 4.5.5 c) and the superposition of the test image with the colored change image as shown in Fig. 4.5.5 d) demonstrate that the EigenBlockCD algorithm detects significant changes as confirmed by the radiologist, i.e., it detects the interval development of bilateral small hyperintense foci in the periventricular region on the follow-up image.

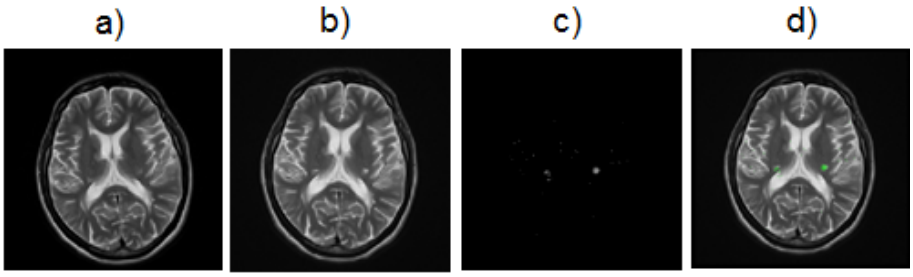

Figure 4.5.5: a) and b) Selected axial T2-weighted images of the brain of a patient who had undergone serial MRIs in 2011 and 2013. Images are at the level of the lateral ventricles and demonstrate interval development of bilateral small hyperintense foci in the periventricular region on the follow-up image. c) and d) Corresponding change image detected by EigenBlockCD algorithm and overlay image highlighting these regions.

Fig. 4.5.6 shows the results of two real T2-weighted MR images. The follow-up image in Fig. 4.5.6 b) demonstrates multiple periventricular foci of increased signal. A few of these foci show interval enlargement on the follow-up image. The change image obtained by the EigenBlockCD algorithm and the superposition of the test image with the colored change image in Fig. 4.5.6 c) and d) show that the EigenBlockCD algorithm detects these significant changes between scans which are confirmed by the radiologist.

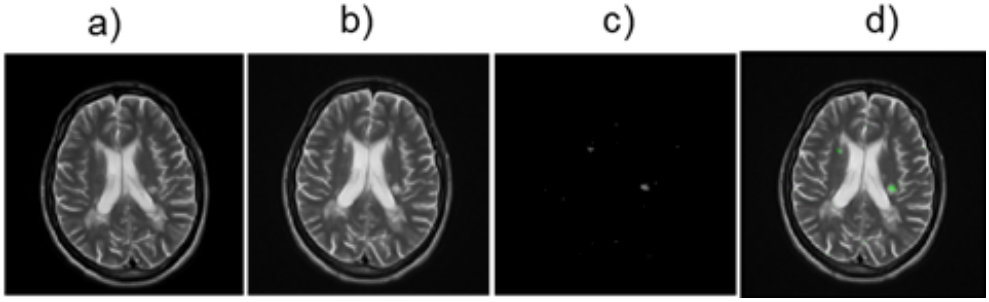

Figure 4.5.6: Axial T2-weighted images of the brain of the same patient at the level of the lateral ventricles more superiorly. Here (a) initial and (b) follow-up images demonstrate multiple periventricular foci of increased signal. A few of these foci show interval enlargement on the follow-up image. c) and d) Corresponding change image and overlay image are highlighting these regions.

Fig. 4.5.7 shows axial T2-weighted image of the brain of a patient at the level of the lateral ventricles further superiorly demonstrating bilateral small hyperintense foci in the periventricular region on both the initial and follow-up images with only one region of interval change. Corresponding change image and overlay image are highlighting this region as significant changes as shown in Fig. 4.5.7 c) and d).

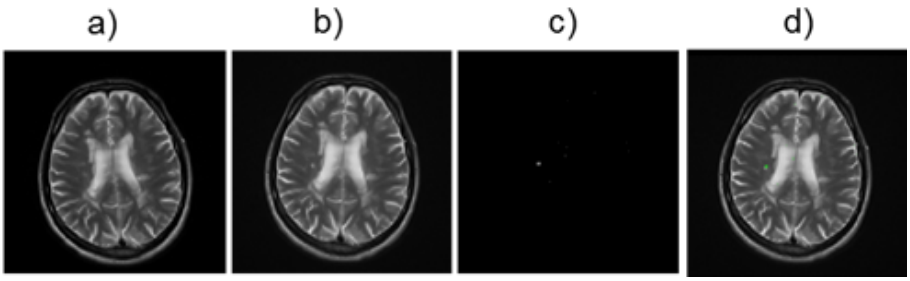

Figure 4.5.7: Axial T2-weighted image of the brain of a patient at the level of the lateral ventricles further superiorly demonstrating bilateral small hyperintense foci in the periventricular region on both a) the initial and (b) follow-up images with only one region of interval change. c) and d) Corresponding change image and overlay image are highlighting this region.

Identifying that there is no change in the follow-up is as tricky as identifying a change. Fig. 4.5.8 shows axial T2-weighted images of the brain of a patient who had undergone serial MRIs in 2010 and 2012. Both images have the same inactive and visible MS lesion which hasn't grown or changed between two exams periods. Images are at the level of the lateral ventricles and demonstrate a large and several smaller hyperintense foci in the periventricular region on both the initial and follow-up images without interval change in size or intensity. The corresponding change image and the overlay image are showing no interval change as shown in Fig. 4.5.8 c) and d).

We also tested our method with images obtained from the simulated brain database BrainWeb [17]. Reference is an image of a normal brain and the test is an image of a brain with moderate MS lesions. Other variables were kept constant: both are T2-weighted MR images with 1mm slice thickness, 3% noise and 0% intensity non-uniformity (RF). Both the change image and the superposition of the test image with the colored change image in Fig. 4.5.9 c) and d) show that our algorithm captures the significant changes related to the new MS lesions formation while rejecting insignificant changes.

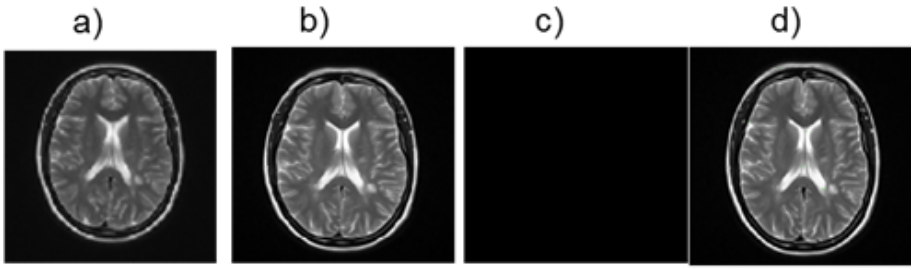

Figure 4.5.8: a) and b) Selected axial T2 images of the brain of a patient who had undergone serial MRIs in 2010 and 2012. Images are at the level of the lateral ventricles and demonstrate a large and several smaller hyperintense foci in the periventricular region on both the initial and follow-up images without interval change in size or intensity evident. c) and d) Corresponding change image and overlay image shows no interval change.

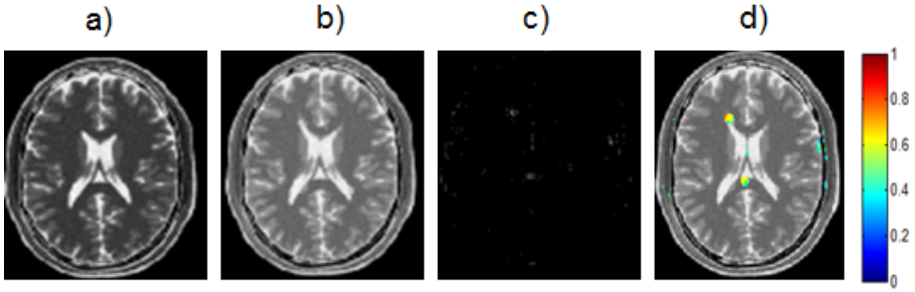

Figure 4.5.9: a) Reference is an image of a normal brain. b) Test image is an image of a brain with moderate MS lesions. c) Results of change detection image via our EigenBlockCD algorithm. d) Superimposed test image with change image obtained by EigenBlockCD. Images are are taken from BrainWeb [17].

4.6 Summary

In this section we presented the EigenBlock Change Detection algorithm (EigenBlockCD) as an application of adaptive dictionary learning techniques. The main differences between the EigenBlockCD algorithm and the AEDL algorithm are:

1. The EigenBlockCD algorithm finds the best matching block from the blocks in the reference image to approximate the background of the block of interest in the test image, while the AEDL algorithm expresses the block of interest in the test image as a sparse linear combination of training blocks in the reference image.

2. The EigenBlockCD algorithm computes the background of a block of interest by minimizing the L_2-distances between the block of interest and all the training blocks in a dictionary, whereas the AEDL algorithm approximates the background by using L_1 minimization techniques.

The algorithm captures local spatial changes between the reference and test images and consists of three stages. In the first stage change image is obtained by averaging all the absolute differences of test image blocks with their best approximation blocks given by a local dictionary trained from the reference image. In Stage Two the process is similar, but the local dictionaries are learned from the test image blocks. In the third stage the change image between the reference and test image is computed as the average of the two outputs from stages one and two. Using L_2 norm as a similarity measure the method finds the closest block from the inquiry block in the reference image to a block from the test image via adaptive dictionary learning technique.

This algorithm automatically detects the significant structural changes in the test image while ignoring unimportant changes related to mis-alignments, noise and acquisition-related artifacts. Experiments on synthetic images illustrate that the EigenBlockCD algorithm detects changes due to appearance or disappearance of objects in the presence of object translations, rotations, intensity changes and moderate noise level. Experiments on MR images show that our algorithm identifies clinically significant changes and rejects clinically insignificant changes within the area of the local dictionary. In addition, we make use of PCA to reduce the size of a local dictionary for each image block, to find the best matching blocks and improve computational efficiency.

The EigenBlockCD algorithm is more computational efficient than the AEDL algorithm. However, for large misalignments between images it requires larger blocks of interests and inquiry blocks and therefore larger dictionary learning, which relatively increases the computational time. Although the EigenBlockCD algorithm performs faster compared to the AEDL algorithm, the computational efficiency is still a limitation. In Chapter 5 we present an improved version of the EigenBlockCD algorithm (EigenBlockCD-2) which is computationally more efficient, accounts for large image shifts, rotations and other misalignments.

In summary, the choice of metric is vital to preserve consistent solutions from the image space to the eigenspace. The L_1 norm fails to preserve the distance ordering and may produce different solutions in image space and the eigenspace. Meanwhile, the L_2 norm is certainly invariant from the change of orthogonal basis and preserves

the distance ordering. Hence the solution given by the L_2 minimization is consistent in both the image space and the eigenspace.

In Section 4.3, we provided theoretical and numerical support that L_2 norm is distance invariant from the change of orthonormal basis. That is, the L_2 norm preserves distances between two vectors projected onto any subspace spanned by an orthonormal basis. Also, L_2 norm is order invariant from the change of orthonormal basis as it preserves the order between two vectors projected onto a subspace spanned by an orthonormal basis. The same claim doesn't hold for L_1 norm.

We provided four counter-examples to show that the L_1 norm isn't distance invariant and isn't order invariant from the change of orthonormal basis. We also showed that L_1 norm doesn't preserve the structure of a block when projected onto an orthonormal basis.

Chapter 5

The Effect of Co-Registration in Change Detection

5.1 An Introduction

In Chapter 3, we explored the possibility of utilizing image sparsity and compressive sensing to automatically detect changes between reference and test images by designing the **A**daptive **E**igenBlock **D**ictionary **L**earning algorithm [23]. However, because of the long processing time of L_1 minimization, we adapted the rational behind the AEDL algorithm and established a simpler but more effective new algorithm, namely EigenBlock Change Detection (EigenBlockCD) [25, 26], as described in Chapter 4.

Both the AEDL and the EigenBlockCD algorithms are based on dictionary learning techniques which are widely used in face and pattern recognition problems, and also with applications in sparse images [13, 15]. The AEDL and the EigenBlockCD use local dictionaries learned from one of the images to detect the changes in the consecutive MRI follow-up scans. The key idea for both the AEDL and the EigenBlockCD algorithm is to perform a local image registration for approximating a common background between two images in order to identify important structural changes, such as appearance, disappearance, growth or shrinkage of a lesion, while automatically rejecting unimportant changes due to spatial position of patients and measurement noise.

The main difference between these two algorithms is that in the AEDL algorithm the block of interest is expressed by a sparse linear combination of training blocks from

the inquiry block and the best matching block to the block of interest is determined by using L_1-minimization. The EigenBlockCD aims to find the best matching block from the local dictionary in the inquiry block using L_2-minimization as the similarity measure. The core of both methods is minimizing a cost function which expresses the dissimilarity map between the images. The minimization of the cost function finds the background information that includes insignificant changes and is used to identify significant changes.

The earlier versions of both the AEDL and the EigenBlockCD algorithms perform well if the size of the inquiry block is large enough [23, 24]. As the size of the inquiry block increases, the run-time increases significantly. Also, as the size of the inquiry block decreases, the error increases leading to the increase of false positives. To overcome these limitations, we add the co-registration step as the key to solving this problem. We design an improved versions for both the AEDL and the EigenBlockCD algorithms for detecting changes between consecutive MR images of the brain, based on dictionary learning techniques. The improved algorithms have no additional preprocessing steps and employ L_1 and L_2 norm respectively as the most effective similarity measure. By bringing the co-registration step the improved algorithms can automatically correct large shifts and rotations and their performance is improved significantly.

In general, a change detection algorithm maps two input images to a third image, namely change image, consisting of only the significant changes between the two input images. In this chapter we explain the improved version of the EigenBlockCD algorithm, named as the EigenBlockCD-2. Note that the conclusions and the numerical results obtained for the EigenBlockCD-2 also apply to the improved version of the AEDL algorithm, named the AEDL-2, but we focus our discussion mostly to the EigenBlockCD-2 algorithm.

The EigenBlockCD-2 algorithm is different from the initial version EigenBlockCD in the following aspects:

1. Includes its own co-registration step, a very important step for any change detection algorithm.

2. Accounts for large rotations and translations.

3. Reduces the computations significantly.

4. Reduces the error and hence improves the detection accuracy significantly.

We describe the EigenBlockCD-2 algorithm and provide a comprehensive comparison of the two algorithms. In both cases, we assume that the differences between the two images may come from disease related changes occurred from time t_1 to time t_2, and/or from misalignment due to subject position shifts and rotations, and other common imaging acquisition artifacts. During this time interval, each block from the reference image has undergone a few disease related changes. Most pixels of the reference image will appear again in the test image either at the same location or nearby. Hence, the insignificant changes in the follow-ups are within a neighborhood.

Experiments with synthetic and real MR images show that the EigenBlockCD-2 algorithm performs better in cases with large shifts and rotations compared to the previous version EigenBlockCD algorithm. The rest of this chapter is organized as follows. In Section 5.2 we discuss the core of the co-registration step followed by the implementation of the EigenBlockCD algorithm in Section 5.3. We compare the two versions, the EigenBlockCD and the EigenBlockCD-2 algorithms in Section 5.4 using synthetic and real human data.

5.2 The EigenBlockCD-2 Algorithm

The EigenBlockCD-2 consists of the following main steps:

1. **Initial global alignment** of images by first determining three parameters followed by linear transformations to align one of the images with the other one. In this step the radii of blocks of interest and the inquiry blocks are selected significantly large to account for large shifts and rotations.

2. **Computing the significant changes** between the two images by running the EigenBlockCD algorithm with fixed small blocks of interest.

5.2.1 Initial Global Co-Registration of Images

We introduce a dictionary-based co-registration step as a very important step for our change detection algorithm. Detecting only relevant and significant changes by decreasing false positives and increasing true positives is one of the benefits of the co-registration. The initial global image registration step is a major improvement which enables the algorithm to identify important structural changes, such as appearance, disappearance, growth or shrinkage of a lesion while automatically rejecting unimportant changes due to spatial changes of patients and common imaging acquisition artifacts.

In general, the registration problem is finding a transformation that best aligns the two images. It involves determining and handling of a transformation space and a cost function that quantifies the quality of the registration. In the registration problem, the transformation space includes six rigid transformations in 3D with six degrees of freedom (three translations and three rotations) and three transformations with three degrees of freedom in 2D (two for translations and one for rotation). In this chapter, we discuss the initial global co-registration problem between two images in two dimensional space. The cost function expresses the dissimilarity map between the images.

Let $\mathcal{T}$ represent a set of transform functions, $\mathcal{P}$ a set of transformations' parameters and $\mathcal{C}$ the cost function to be minimized. The registration problem in two dimensional case is finding three parameters to align the images. That is, if either one or both images go through linear transformations using these parameters, the two transformed images will be aligned to each other. The cost function measures the accuracy of the alignment, that is, smaller values of cost function correspond to higher accuracy of the co-registration. We use the L_2 norm to measure the accuracy of the initial global alignment, that is, the accuracy of the co-registration is maximal when the L_2 norm of the difference between one image and the second image transformed is minimal. Thus, we define the co-registration problem with the following minimization problem:

$$\min_{\mathcal{T},\mathcal{P}} \mathcal{C}\big(\mathcal{T}, \mathbf{I}_1, \mathbf{I}_2, \mathcal{P}\big) \tag{5.2.1.1}$$

where $\mathcal{C}$, $\mathcal{T}$, $\mathcal{P}$, $\mathbf{I}_1$, and $\mathbf{I}_2$ respectively are:

- $\mathcal{C}$ is the cost function in the minimization problem. In our case, the cost function is the L_2 norm, i.e., $\mathcal{C} = ||.||_2$.

- $\mathcal{T}$ is the set of transform functions. In our case, $\mathcal{T}$ is any composition of transform functions from the set $\{\mathbf{T}_h, \mathbf{T}_v, \mathbf{R}_\theta\}$, where $\mathbf{T}_h, \mathbf{T}_v, \mathbf{R}_\theta$ are the horizontal, vertical, and rotational transform functions respectively.

- $\mathcal{P} = \{(h, v, \theta) : \ h \in \mathbf{T}_h, v \in \mathbf{T}_v, \theta \in \mathbf{R}_\theta\}$ is a set of all triples (h, v, θ) representing the three parameters needed to align the two images, i.e., the magnitudes of horizontal shift, vertical shift and rotation angle that one image should be transformed to be aligned to the other image.

- $\mathbf{I}_1$ and $\mathbf{I}_2$ represent the two images to be aligned. It is easy to extend the model to align more than two images in 2D or two sets of 3D image volumes.

Let $I(i, j)$ represent the intensity of the image pixel at location (i, j). Image $\mathbf{I}$ is of size $N_1 \times N_2$. The horizontal and vertical transform functions can be applied to each image pixel as follows:

$$\mathbf{T}_h I(i,j)\big) = I(i, j + h)$$
$$\mathbf{T}_v\big(I(i,j)\big) = I(i + v, j)$$

$$\mathbf{R}_\theta\big(I(i,j)\big) = I\Big((i - \frac{N_1}{2})cos\theta - (i - \frac{N_2}{2})sin\theta, (i - \frac{N_1}{2})sin\theta - (i - \frac{N_2}{2})cos\theta\Big)$$

In our case we consider a composition of all the three transform functions, i.e., $\mathcal{T}_{h,v,\theta} = \mathbf{T}_h \circ \mathbf{T}_v \circ \mathbf{R}_\theta$. In matrix form $\mathcal{T}_{h,v,\theta}$ can be written as:

$$\begin{bmatrix} i' \\ j' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & v \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix}$$

where (i, j) and (i', j') represent the original and the transformed coordinates respectively. The co-registration problem in Eq. (5.2.1.1) can be written as:

$$\min_{\mathcal{T},\mathcal{P}} ||\mathcal{T}(\mathbf{I}_1) - \mathbf{I}_2||_2 \tag{5.2.1.2}$$

The problem seeks to find a triple (h^*, v^*, θ^*) that minimizes Eq. (5.2.1.2):

$$\min_{(h,v,\theta)\in\mathcal{P}} ||\mathcal{T}(\mathbf{I}_1) - \mathbf{I}_2||_2 = \min_{(h,v,\theta)\in\mathcal{P}} ||\mathcal{T}_{h,v,\theta}(\mathbf{I}_1) - \mathbf{I}_2||_2 \tag{5.2.1.3}$$

Eq. (5.2.1.3) can be written as:

$$\begin{aligned} &\Big(\min_{(h,v,\theta)\in\mathcal{P}} ||\mathcal{T}_{h,v,\theta}(\mathbf{I}_1) - \mathbf{I}_2||_2\Big) \Longrightarrow \\ \Longrightarrow &\Big(\exists(h^*, v^*, \theta^*) \in \mathcal{P}, \quad \text{s.t:} \quad \mathcal{T}_{h^*,v^*,\theta*}(\mathbf{I}_1) \approx \mathbf{I}_2\Big) \end{aligned} \tag{5.2.1.4}$$

where triples (h, v, θ) are automatically determined by the co-registration step. The two translation parameters are determined directly without any input and the rotation parameter is selected from a range of given angles with integer values, i.e., $\theta \in [-10^o, 10^o]$. Theoretically there are infinite values $\in [-180^o, 180^o]$ for the parameter θ representing the rotation angles, but practically good enough to correct the rotation with 1 degree accuracy. This is because in the second step, our algorithm ignores changes due to small degrees of rotations. We define a good alignment between the two images if the original image and a transformed image do not differ more than 1^o in terms of rotation and no more to three pixels in terms of translations. Hence there are only a finite number of values for the parameter θ. It derives from the nature of our problem that the two translation parameters h and v consist of only integer values.

Remark: The insignificant changes between two consecutive images due to affine transformations (translations and rotations) are bounded. More specifically, the sizes of the horizontal and vertical shifts cannot be greater than the one half of the maximum image dimension. That is, $|h^*| \leq \frac{1}{2}\max\{N_1, N_2\}$ and $|v^*| \leq \frac{1}{2}\max\{N_1, N_2\}$, where N_1, N_2 are number of rows and columns in the image respectively. Similarly, size of the rotation angle is the range $[-180^o, 180^o]$. This means that there is a solution to the minimization problem in Eq. (5.2.1.4). In other words, there exists a triple (h^*, v^*, θ^*) that minimizes the cost function.

5.2.2 Selection of Blocks during the Co-Registration Step

To align the reference and test images, the EigenBlockCD-2 algorithm automatically selects a few blocks in the test image and their corresponding inquiry blocks from the reference image as shown in Fig.5.2.1.

First, we describe how the blocks of interest are selected. The test image is divided into 25 sub-images. Their sizes should be large enough to contain some anatomical structures. The sub-images shown in Figs.5.2.1 a) to 5.2.3 a) are of size 77×77. The EigenBlockCD-2 algorithm first identifies nine sub-images inside the brain and then for each selected sub-image, it randomly picks a center to form a block of interest. To account for large shifts and rotations, the radii of blocks of interest and corresponding inquiry blocks are chosen relatively large. In our experiments blocks of interest are of size 25×25 and their corresponding inquiry blocks are of size 65×65. The output of these computations is composed of two translations and one rotation needed to globally co-register the images. More specifically, the EigenBlockCD-2 algorithm in the co-registration step:

- Determines the shifts and a rotation parameters needed to globally align 2D images with:

  - Inputs: reference and test images, radius of the block of interest, radius of the inquiry block in the reference image and a range of rotation angles about x-axis.

  - Outputs: two translation parameters and a rotation angle needed to globally co-register images, i.e., vertical and horizontal shift parameters and rotation angle about x-axis.

- Aligns the 2D images by performing a linear transformation to one image to be aligned with its follow-up image. The linear transformation is a composition of two translation transforms and one angular rotation.

The EigenBlockCD-2 algorithm identifies the inquiry blocks in the reference image for each given block of interest in the test image. The dictionary is created with training blocks from the inquiry block. The dictionary is rotated about x-axis with an angle selected from a given range of angles, i.e., $[-10^o, 10^o]$, as shown in Fig.5.4.7 and

Block of Interest	1	2	3	4	5	6	7	8	9	Median
Vertical shift	-24	-22	-22	-21	-20	-20	-13	-9	22	-20
Horizontal Shift	-1	-3	4	3	-1	0	5	-2	-22	-1
Rotation Angle	1	-3	1	-4	-1	0	4	-4	4	-1

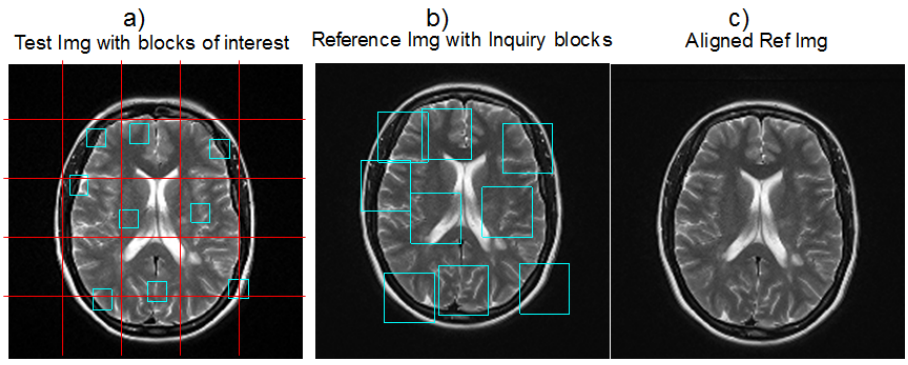

Figure 5.2.1: Top row: three parameters (two for the vertical and horizontal shifts and one for rotation angles) for nine selected blocks from the test image. bottom row: a) MR image in 2012 with nine blocks of interest of size 25×25 selected randomly from nine inner areas; b) An MR image taken in 2010 with corresponding inquiry blocks of sizes 65×65; c) Reference image aligned with test image after two translations and one rotation. In this case the reference image is rotated one degree counter$-$clockwise, shifted 20 pixels down and one pixel to the left.

Block of Interest	1	2	3	4	5	6	7	8	9	Median
Vertical shift	-22	-22	-21	-20	-19	-19	-18	-18	-17	-19
Horizontal Shift	-2	-1	0	-3	-1	-2	2	-3	-2	-1
Rotation Angle	-1	-2	-1	2	-2	-3	-3	0	-1	-2

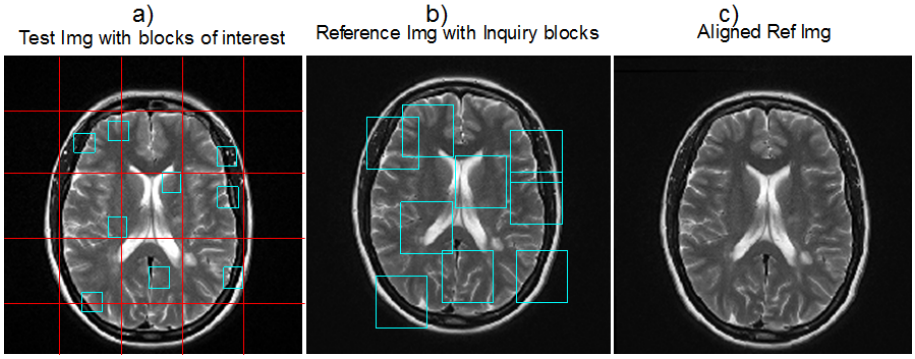

Figure 5.2.2: Top row: three parameters (two for the vertical and horizontal shifts and one for rotation angles) for nine selected blocks from the test image. Bottom row: a) MR image in 2012 with nine blocks of interest of size 25×25 selected randomly from nine inner areas; b) An MR image taken in 2010 with corresponding inquiry blocks of sizes 65×65; c) Reference image aligned with test image after two translations and one rotation. In this case reference image is rotated one degree counter-clockwise, shifted 19 pixels down and two pixels to the left.

Fig.5.4.9. Then, the best matching block for a selected block of interest is computed by minimizing the L_2 differences between the block of interest and all the training blocks projected into the eigen-subspace with only 85% of the total variance.

The process is repeated for all the other angles in that given range and the block

Block of Interest	1	2	3	4	5	6	7	8	9	Median
Vertical shift	-32	-22	-22	-21	-20	-19	-18	-9	3	-20
Horizontal Shift	-6	-7	0	2	-1	-2	0	-15	2	-1
Rotation Angle	0	-3	-2	-1	-2	0	-1	4	0	-2

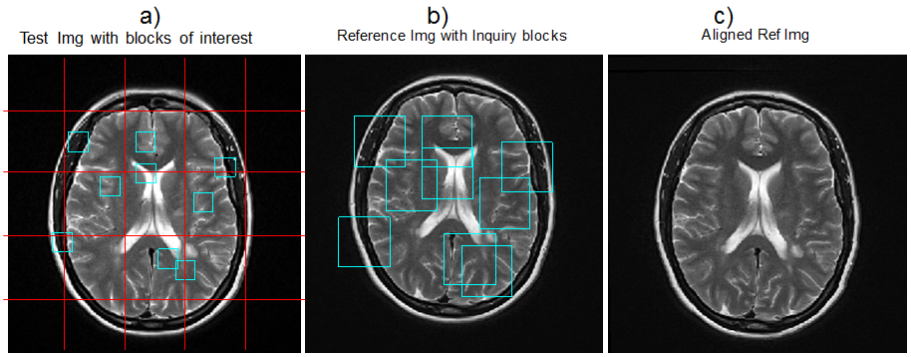

Figure 5.2.3: Top row: three parameters (two for the vertical and horizontal shifts and one for rotation angles) for nine selected blocks from the test image. Bottom row: a) MR image in 2012 with nine blocks of interest of size 25×25 selected randomly from nine inner areas; b) An MR image taken in 2010 with corresponding inquiry blocks of sizes 65×65; c) Reference image aligned with test image after two translations and one rotation. In this case the reference image is rotated 2 degrees counter-clockwise, shifted 20 pixels down and one pixel to the left.

with the minimum L_2 value is selected as the best matching block to the block of interest. The position of this block in the inquiry block relative to the center of the inquiry block determines the vertical and horizontal translations as shown in Figs.5.4.6 and 5.4.8. The angle for which this minimum is reached determines the rotation angle. The algorithm repeats for all other selected blocks of interest. The three parameters needed to align the images are those that correspond to the median of the parameters determined as described above for all the nine blocks of interest as shown in Figs.5.2.1, 5.2.2 and 5.2.3. Finally, rigid transformations with these three computed parameters are performed to one of the images to align it with the other image.

5.2.3 Effect of Choices of Blocks in Co-registration Step

The robustness of the co-registration step guarantees our assumption that pixels in the follow-up scan will be either at the same location or nearby. To demonstrate co-registration accuracy we run two tests with real MR scans of size 384×384 and synthetic MR images.

More specifically, in the first test, we run the co-registration step experiment 30 times using a pair of real MR images. Three parameters are computed for each ex-

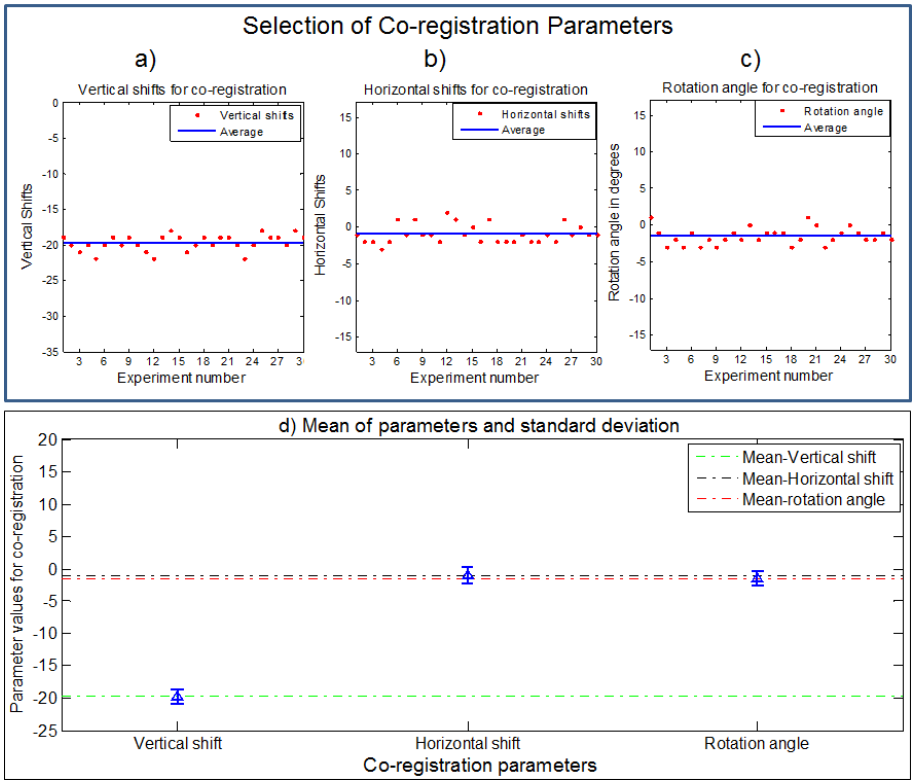

Figure 5.2.4: a) and b) Vertical and horizontal shifts from 30 experiment identified from the co-registration step, c) Rotation angles of 30 experiments identified from the co-registration step, d) Mean of the three parameters and their standard deviation. Co-registration parameters are along x-axis (vertical shift, horizontal shift and rotation angle). Values of these parameters are along the y-axis.

periment, as described above. The mean and standard deviation for each parameter is determined and the results are displayed in Fig.5.2.4. Vertical and horizontal shifts and their averages from 30 experiments computed from the co-registration step are shown in Fig.5.2.4 a) and b) respectively. Rotational angles calculated from the co-registration step of 30 experiments and their averages are shown in Fig.5.2.4 c). The means of the three parameters and their standard deviations are shown in Fig.5.2.4 d). The error-bars in Fig.5.2.4 d) show that all three parameters obtained from 30 trials are closer to their mean which explains the correctness of the co-registration step.

We also tested the co-registration accuracy with synthetic images created from real MR images. To validate the performance of our co-registration step, we created a test image by applying linear transformations on the reference image. The transformation set consists of two types of translations (vertical and horizontal) with values of their parameters chosen randomly in a range [-20 20] and [-16 16] respectively and a rotation with an angle chosen randomly in [-6 6]. Each triple of transformation parameters

obtained in this way forms the ground truth, that is, the true transformations that need to be applied to reference image to be aligned it with the test image. For each pair of synthetic images created in this way, we run the co-registration step which then computes the three co-registration parameters.

To compare the computed and the true parameters, the differences between them are calculated as shown in Fig.5.2.5. We repeat this experiment 20 times. The means and the standard deviations of the differences for the 20 experiments are also computed. The differences of the vertical and the horizontal shifts and their averages from 20 experiments are shown in Fig.5.2.5 a) and b) respectively. The differences of rotation angles calculated from the co-registration step of 20 experiments and their average are shown in Fig.5.2.5 c). Fig.5.2.5 d) shows the mean and standard deviation of the differences. The error-bars in Fig.5.2.5 d) show that all three parameters obtained from 20 experiments are close to their mean. It can be easily seen that the absolute values of the differences for all three parameters are less than 2. This guarantees that after co-registration step, pixels in the follow-ups will be either at the same location or close enough for any inquiry block of size ≥ 5 to contain the best matching block for a selected block of interest.

A visual comparison of Experiment 1 and 5 are shown in Fig.5.2.6. Images obtained after co-registration step shown in Fig.5.2.6 Column c) are aligned with their follow-ups images as shown in Fig.5.2.6 Column b).

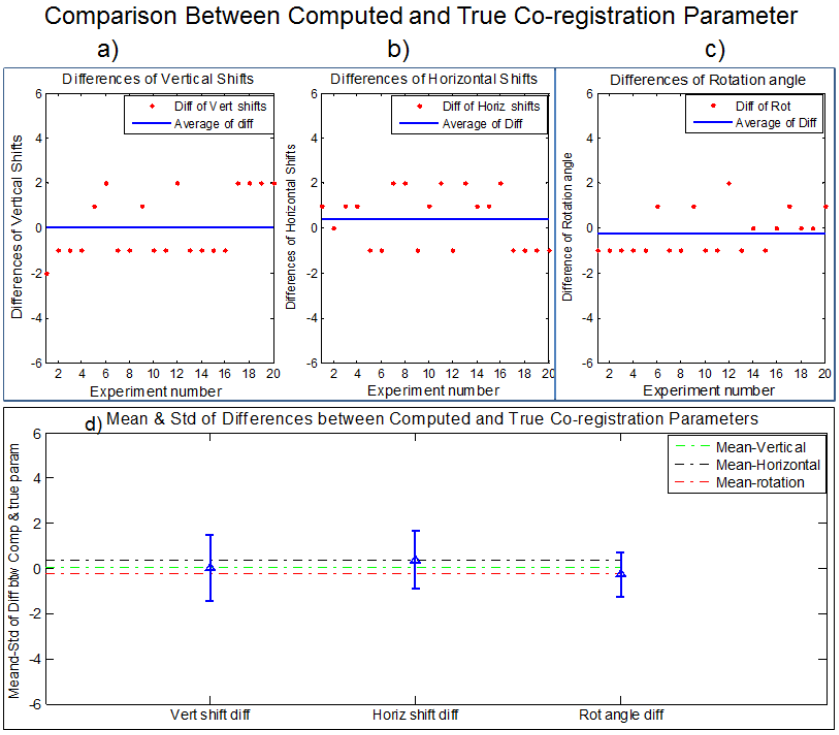

Figure 5.2.5: a) and b) Differences between true and computed vertical and horizontal shifts from 20 experiments, c) Differences between true and computed rotation angles of 20 experiments, d) Means and standard deviations of the three differences. Co-registration parameters are along x-axis, vertical shift, horizontal shift, and rotation angle. Values of the differences of these parameters are along the y-axis.

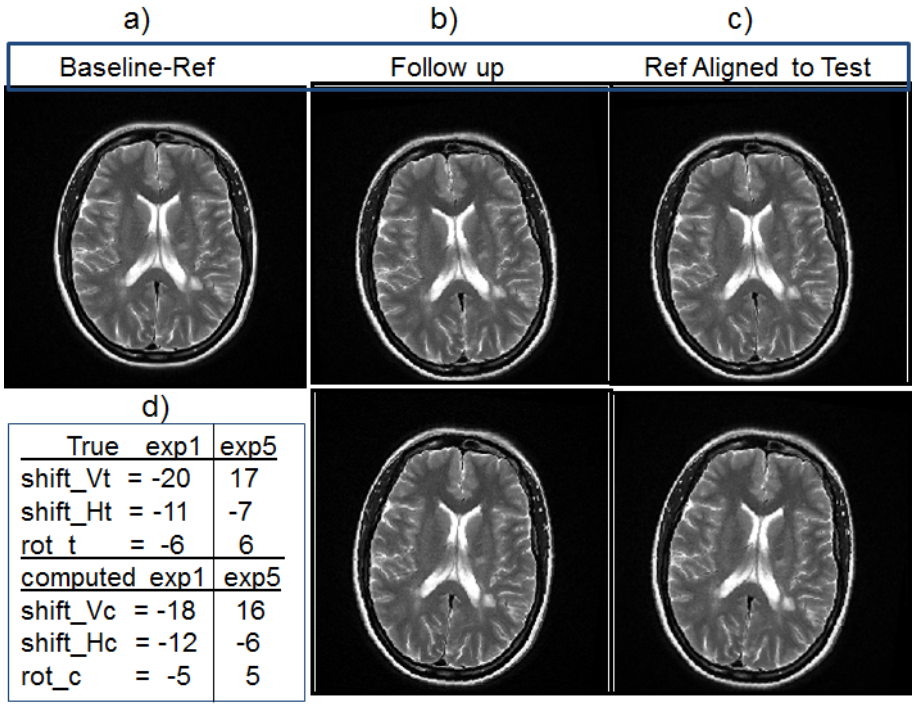

Figure 5.2.6: a) Baseline-reference image, Column b) Follow-up images from the first and fifth experiment with synthetic data, Column c) Corresponding reference aligned with its follow-up image, d) Values of the ground truth parameters and its corresponding computed ones by the co-registration step of the EigenBlockCD-2 algorithm.

5.3 Using EigenBlockCD in EigenBlockCD-2

In Section 5.2, we described the co-registration step of the EigenBlockCD-2 algorithm and its effects in our change detection algorithm. In this section we briefly discuss the second step of the EigenBlockCD-2 algorithm, determining the significant changes. The detailed explanation can be found in in Chapter 4 and also in Nika et al [23, 24].

We use the same notations of Chapters 3 and 4: $\mathbf{I}_1$ - reference image, $\mathbf{I}_2$ - test image, $\mathbf{b}$ - block of interest, $\mathbf{B}$ - inquiry block, $\mathbf{a}_k$ - training blocks from inquiry block, δ - diameter of the block of interest and also the diameter of each training block, Δ - diameter of inquiry block, and the dictionary $\mathbf{\Phi}$ formed by stacking training blocks as columns vectors:

$$\mathbf{\Phi} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \tag{5.3.0.1}$$

The EigenBlockCD algorithm computes the background $\mathbf{y}^*$ of $\mathbf{y}$ by seeking to find the best approximation to $\mathbf{y}$ in the reference image, a vector $\mathbf{x}_l$ from dictionary $\mathbf{\Phi}$ that minimizes the residual error. That is:

$$\mathbf{y}^* = \mathbf{x}_l \quad \text{such that:} \quad \mathbf{e}_l = ||\widehat{\mathbf{y}} - \widehat{\mathbf{x}}_l||_2 = \min_{\widehat{x}_k \in \widehat{\mathbf{\Phi}}} ||\widehat{\mathbf{y}} - \widehat{\mathbf{x}}_k||_2 \tag{5.3.0.2}$$

where $\mathbf{x}_l \in \mathbf{\Phi}$, $\widehat{\mathbf{x}}_l \in \widehat{\mathbf{\Phi}}$, and $\widehat{\mathbf{\Phi}}$ is the projected dictionary $\mathbf{\Phi}$ onto the eigen-subspace. Then, we compute the significant changes in the image domain:

$$\mathbf{r} = |\mathbf{y} - \mathbf{y}^*| \tag{5.3.0.3}$$

The change image block for a block of interest $\mathbf{b}$, in other words the foreground of $\mathbf{b}$ is:

$$\mathbf{F} = \frac{1}{2}\left(\frac{1}{m}\sum_{k=j-\delta}^{j+\delta}\sum_{l=i-\delta}^{i+\delta}\mathbf{r}_{k,l}^{[1]} + \frac{1}{m}\sum_{k=j-\delta}^{j+\delta}\sum_{l=i-\delta}^{i+\delta}\mathbf{r}_{k,l}^{[2]}\right) \tag{5.3.0.4}$$

where $\mathbf{r}_{k,l}$ computed as in Eq. (5.3.0.3), is centered at pixel (k,l). Then, the change image is:

$$\mathbf{I}_{cd} = \frac{1}{2}\left(\frac{1}{m}\sum_{j=\delta}^{N_2-\delta}\sum_{i=\delta}^{N_1-\delta}\mathbf{r}^{[1]}(i,j) + \frac{1}{m}\sum_{j=\delta}^{N_2-\delta}\sum_{i=\delta}^{N_1-\delta}\mathbf{r}^{[2]}(i,j)\right) \tag{5.3.0.5}$$

which can be simplified as:

$$\mathbf{I}_{cd} = \frac{1}{2m} \sum_{j=\delta}^{N_2-\delta} \sum_{i=\delta}^{N_1-\delta} \left(\mathbf{r}^{[1]}(i,j) + \mathbf{r}^{[2]}(i,j) \right) \tag{5.3.0.6}$$

5.4 Comparison of EigenBlockCD-2 and EigenBlockCD

In order for the initial EigenBlockCD algorithm to account for large shifts and rotations, due to patient position, blocks of interest and their corresponding inquiry blocks need to be large. A small block of interest might not have enough structure that allows L_2 norm to identify the best matching block from the reference image. Furthermore, if the inquiry block is not large enough it may not contain the best matching block when the patient's position has significantly changed in the follow-up scan, that is, shifts and rotations are large. Lesions that initially appear in the follow-up scan might be missed and lesions that disappear in the next scan might also be missed.

As the size of the inquiry block increases, the run-time of the previous EigenBlockCD algorithm increases significantly. To make it computationally efficient the size of blocks of interest and their corresponding inquiry blocks should be small. The best matching block for a selected block of interest is found by computing the L_2 minimization of the differences between the block of interest and all the training blocks in the projected local dictionary. These differences contain clinical related changes (true positives) and insignificant changes due to the patient's position in the follow-up scans (false positives).

The error of the algorithm consists of these insignificant changes and depends on the size of the inquiry block. As the radius of the inquiry block decreases, the L_2 norm of the differences between the block of interest and all the training blocks increases, and hence, the error increases leading to the increase of false positives.

To overcome these limitations, we added the co-registration step to the EigenBlockCD-2 algorithm. This key step significantly improves the performance of the algorithm in the following aspects:

1. Automatically corrects large rotations and translations.

2. Significantly reduces the sizes of blocks of interest and their corresponding inquiry blocks which results in decreasing the computations and improving the detection accuracy significantly.

To compare the two versions of the algorithms we run three tests with real MR scans of size 384×384.

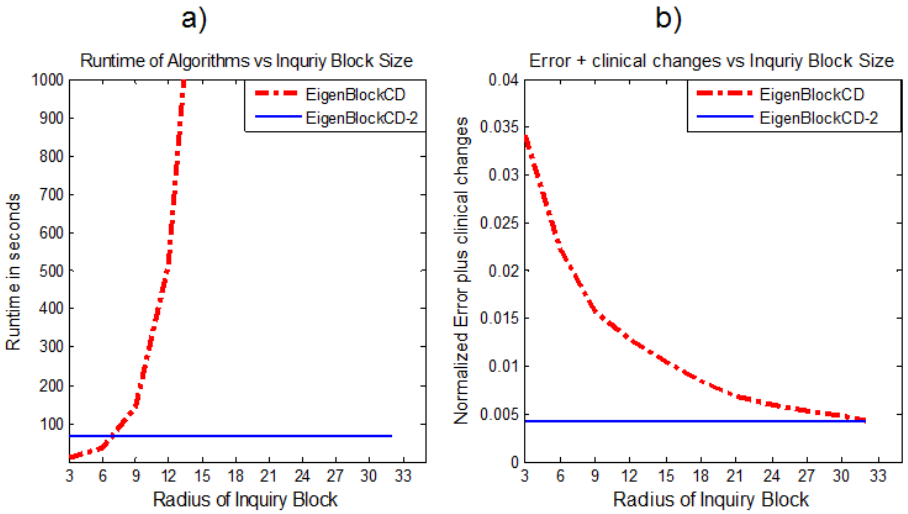

Figure 5.4.1: a) Runtime of both algorithms versus the radius of inquiry block, b) The L_2 norm of the error between each block of interest and its best matching block versus the radius of inquiry block.

In the first test, we compare the computational efficiency of the algorithms by running the experiment 10 times. Fig.5.4.1 a) shows the averages of run-times versus the radius of inquiry blocks for all 10 runs. Fig.5.4.1 a) shows that the EigenBlockCD-2 algorithm computes a change image close to the ground truth in about 60 seconds. This is confirmed by the radiologist. In such time the EigenBlockCD algorithm detects both significant and insignificant changes due to shifts, rotations and other acquisition related artifacts, and hence it produces more false positives as shown in Figs.5.4.2, 5.4.3 and 5.4.4.

In the second test we show the error for each algorithm. This is done by computing the L_2 norm of the change image for different sizes of inquiry blocks as Fig.5.4.1 b) shows. For this pair of MR images used, it is clinically confirmed by the radiologist that the disease related changes are very small. Therefore, we expect the L_2 norm of the desired change images to be small, hence the L_2 norm of the error to be small.

The results of Test 2 is also consistent with the results in the third test. In

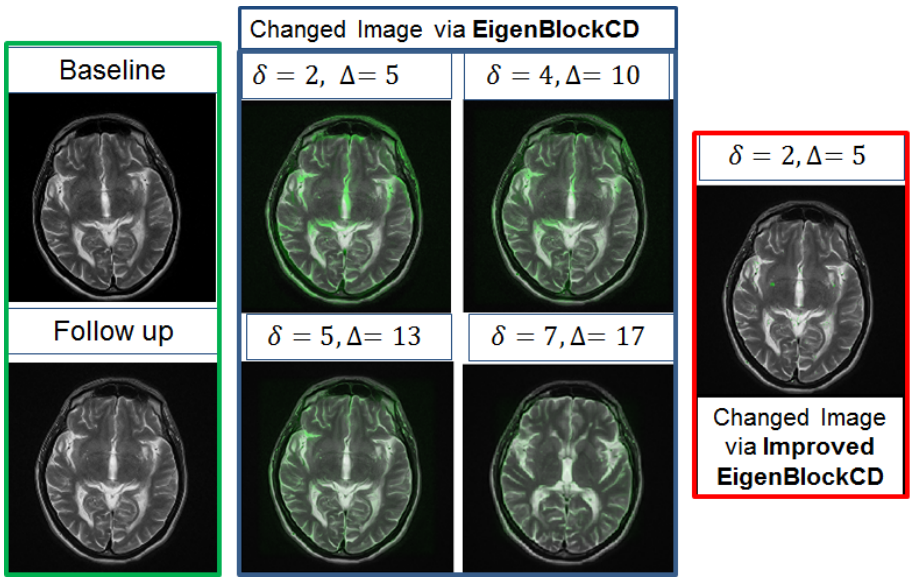

Figure 5.4.2: Left: Two real T2-weighted axial MR images of the brain of a patient at the lateral level. Baseline is a MR image taken in 2011 and follow up scan in 2013. Middle: Four change images via the EigenBlockCD algorithm with different block sizes. The radii of blocks of interest are $\delta = 2, 4, 5,$ and 7 and the corresponding radii of inquiry blocks $\Delta = 5, 10, 13, 17$. Right: change image obtained by the improved EigenBlockCD-2 with $\delta = 2, and \Delta = 5$.

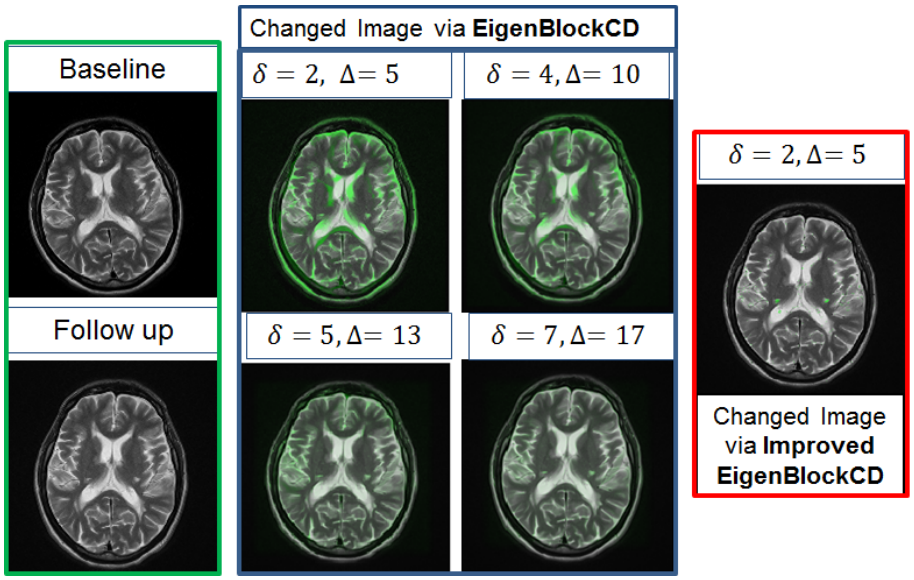

Figure 5.4.3: Left: Two real T2-weighted axial MR images of the brain of a patient at the lateral level, baseline scan in upper left corner and its follow-up scan in down left corner. Middle: four change images via the EigenBlockCD algorithm. The radii of blocks of interest are $\delta = 2, 4, 5,$ and 7 and the corresponding radii of inquiry blocks are $\Delta = 5, 10, 13, 17$. Right: change image obtained by the improved EigenBlockCD-2 with $\delta = 2,$ and $\Delta = 5$.

this test we run the algorithms for three different pairs of real MR images, baselines and their corresponding follow-ups. The radius of blocks of interest and their corresponding inquiry blocks for the EigenBlockCD algorithm vary, i.e., $\delta = 2, 4, 5, 7$ and

$\Delta = 5, 10, 13, 17$. For the EigenBlockCD-2 algorithm we use two fixed pairs of δ and Δ. In the co-registration step, $\delta = 12$ and $\Delta = 20$, and then in the second step both radii are chosen small, i.e., $\delta = 2$ and $\Delta = 5$. By visually comparing the images in Figs.5.4.2, 5.4.3 and 5.4.4, one can observe that the performance of the EigenBlockCD algorithm gets closer to that of the EigenBlockCD-2 algorithm as the size of inquiry block increases and eventually converges to the performance of the EigenBlockCD-2 algorithm for very large blocks sizes.

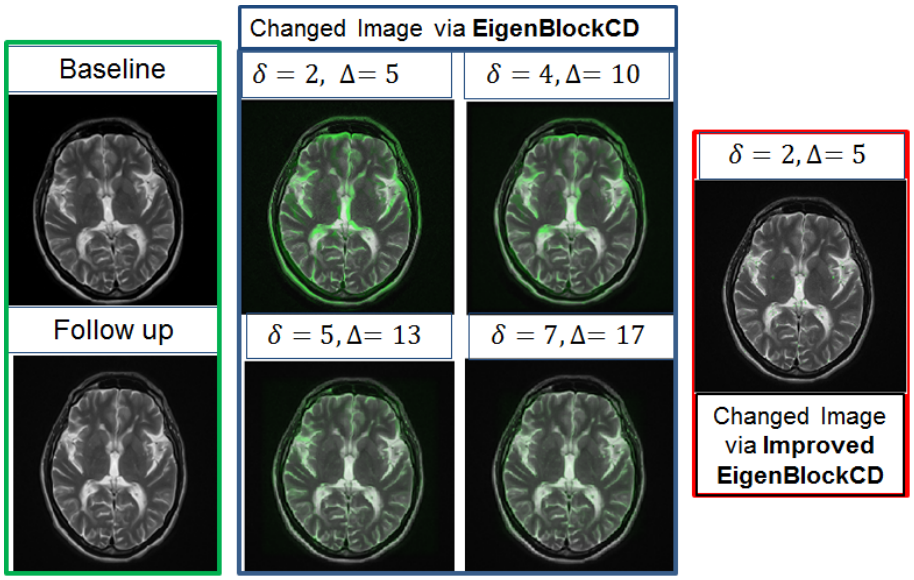

Figure 5.4.4: Left: Two real T2-weighted axial MR images of the brain of a patient at the lateral level, baseline and follow-up scans, Middle: Four change images via the EigenBlockCD. The radii of blocks of interest are $\delta = 2, 4, 5$ and 7 and the corresponding radii of inquiry blocks are $\Delta = 5, 10, 13, 17$. Right: Change image obtained by the improved EigenBlockCD-2 with $\delta = 2$ and $\Delta = 5$.

The error between the block of interest from the test image and its best matching block from the reference image depends on the size of the inquiry block. As the radius of the inquiry block decreases, the error increases and vice versa. The co-registration step solves this problem. Fig.5.4.1 b) shows that L_2 minimum value of this error for the EigenBlockCD algorithm is reached for a radius of inquiry block greater than the size of the shifts and rotations, i.e., $\Delta \geq 30$. Whereas the same minimum is reached by the EigenBlockCD-2 algorithm for $\Delta \geq 5$.

The results for change image via the EigenBlockCD-2 with real MR data are shown in Fig.5.4.5. The first and second rows in Columns a) to c) show selected axial T2-weighted images of the brain of a patient at the level of the lateral ventricles who had

undergone serial MRIs in 2011 and 2013. The third row shows changes obtained by the EigenBlockCD-2 algorithm, overlayed onto the MR 2013 image. Column d) shows selected axial T2 image of the brain of another patient who had undergone serial MRI in 2010 and 2012.

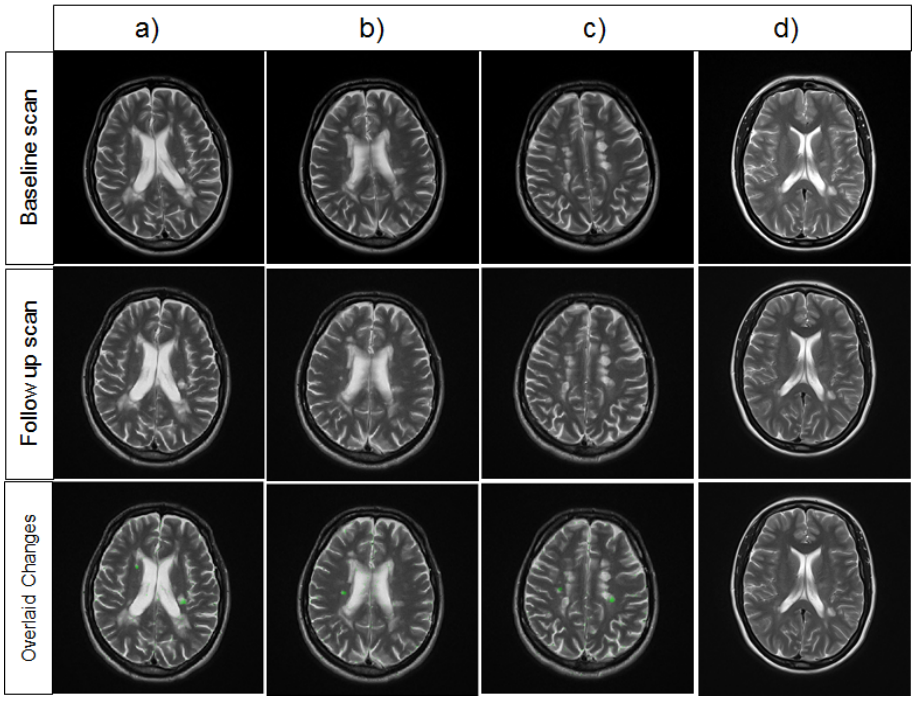

Figure 5.4.5: First row: Axial T2-weighted images of the brain of two patients at the lateral level. Second row: Corresponding follow-up images of the same patients. Third row: Overlay change images computed via the EigenBlockCD-2 algorithm onto the follow-up images.

We also run experiments with another pair of real MR scans. The initial global co-registration step of two MR images taken in 2010 and 2012 namely reference and test images are shown in Fig.5.2.1, Fig.5.2.2 and Fig.5.2.3. Both images are of size 384×384. Blocks of interest selected randomly from nine inner areas from the test image are of sizes 25×25 and their corresponding inquiry blocks in the reference image are of sizes 65×65. The maximum values of vertical and horizontal translations are half sizes of each dimensions in the inquiry block $\mathbf{B}$, i.e., translation parameters of Fig.5.2.1, Fig.5.2.2, Fig.5.2.3 are in the range [-32, 32] pixels and rotation angles is randomly selected from $[-10^o, 10^o]$.

For each degree from $[-10^o, 10^o]$, the best matching block is chosen among 4225 dictionary blocks. In total, it results in 21 blocks selected for the 21 angles. In Fig.5.2.1 the L_2 minimum was reached for vertical shift = -20, horizontal shift 1 and rotation

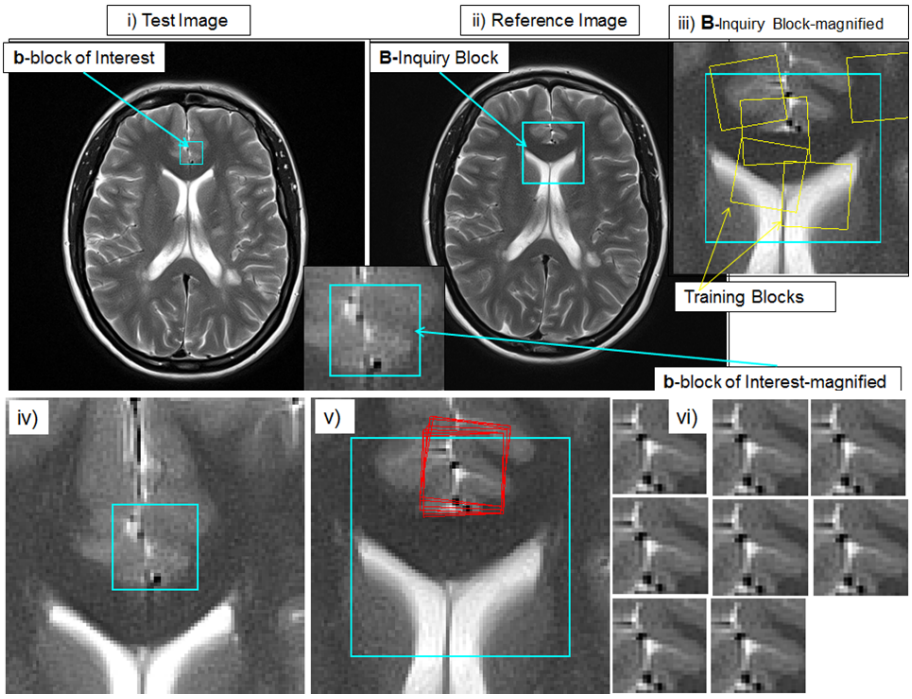

Figure 5.4.6: i) and ii) Baseline and its follow up MR images respectively. A block of interest of size 25×25 centered at pixels (198,120) in the test image, its corresponding inquiry block of size 65×65 in the reference image, iii) Enlarged inquiry block with overlapping training blocks to form the dictionary. iv) Block of interest **b**, v) Inquiry block **B** with the first best 8 matching blocks in red from 21 blocks determined by the algorithm for each angle of rotation, vi) The best 8 blocks closest to the block of interest.

Parameters	1	2	3	4	5	6	7	8	9	10	11
L_2 min	699	701	712	715	721	722	727	729	729	730	730
sh_v	-22	-24	-22	-22	-22	-24	-24	-22	-24	-24	-22
sh_h	-1	-1	-1	-1	-1	-8	-9	-1	-6	-6	-1
$\propto$ (degrees)	-3	-2	-1	0	1	2	3	-6	-5	-4	5
Parameters	12	13	14	15	16	17	18	19	20	21	
L_2 min-	731	735	736	741	745	750	750	753	757	759	
sh_v	-22	-22	-22	-24	-22	-22	-22	-22	-22	-24	
sh_h	-1	-1	-1	-11	-1	-1	-1	-1	-1	-13	
$\propto$ (degrees)	6	7	9	-9	-7	10	8	-8	4	-10	

L_2-min $\Rightarrow \min_{\propto, sh_v sh_h} \|\hat{y} - \hat{x}(\propto, sh_v, sh_h)\|_2$

sh_v, sh_h => Vertical and horizontal shifts with respect to x-y axis

$\propto$ => Rotation angles about x- axis with which the best matching block is rotated

Figure 5.4.7: Three parameters identified by the algorithm in ascending order of their L_2 minimums.

angles about x-axis of -1^o. Hence, the reference image is rotated 1 degree clock-wise, shifted 20 pixels down and one to the left to be aligned with test image as Fig.5.2.1 Column c) shows.

Figs.5.4.6 and 5.4.8 show that the L_2 minimum for one block of interest is reached

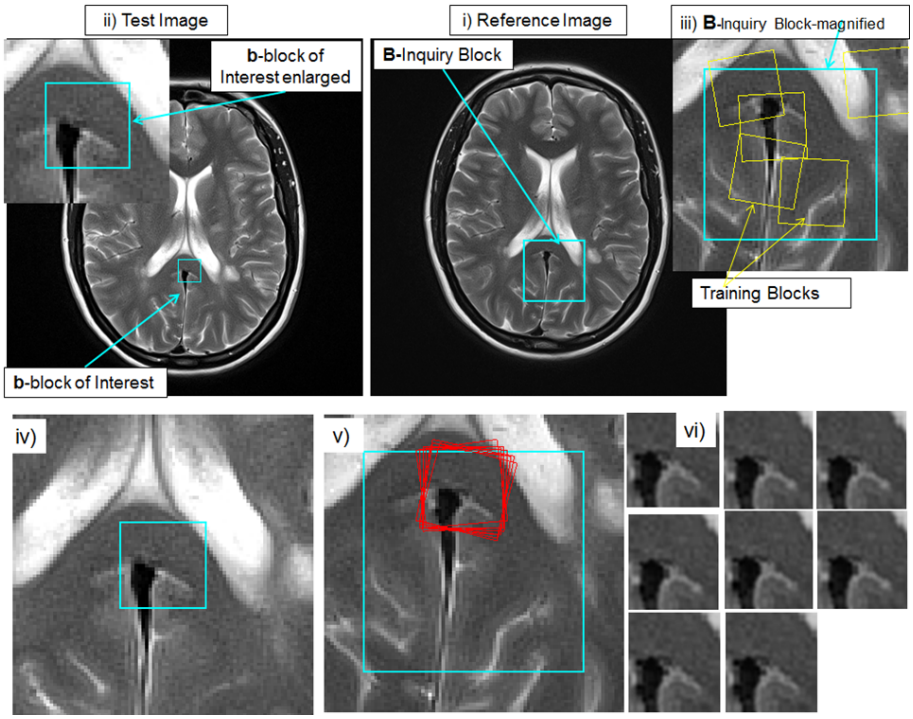

Figure 5.4.8: i) and ii) Baseline and its follow-up MR images respectively. A block of interest of size 25×25 centered at pixels (198, 250) in the test image, its corresponding inquiry block of size 65×65 in the reference image, iii) Enlarged inquiry block with overlapping training blocks to form the dictionary. iv) Block of interest **b**, v) Inquiry block **B** with the first best 8 matching blocks in red from 21 blocks determined by the algorithm for each angle of rotation, vi) The best 8 blocks closest to the block of interest.

Parameters	1	2	3	4	5	6	7	8	9	10	11
L_2 min-	414	431	447	458	476	506	514	534	552	587	592
sh_v	-22	-24	-22	-22	-22	-24	-24	-22	-24	-24	-22
sh_h	-1	-1	-1	-1	-1	-8	-9	-1	-6	-6	-1
$\propto$(in degrees)	-3	-2	-1	0	1	2	3	-6	-5	-4	5
Parameters	12	13	14	15	16	17	18	19	20	21	
L_2 min-	608	625	643	663	686	729	778	784	793	799	
sh_v	-22	-22	-22	-24	-22	-22	-22	-22	-22	-24	
sh_h	-1	-1	-1	-11	-1	-1	-1	-1	-1	-13	
$\propto$ (in degrees)	6	7	9	-9	-7	10	8	-8	4	-10	

L_2-min $\Rightarrow \min_{\propto, sh_v sh_h} \|\hat{y} - \hat{x}(\propto, sh_v, sh_h)\|_2$

sh_v, sh_h => Vertical and horizontal shifts with respect to x-y axis

$\propto$ => Rotation angles about x- axis with which the best matching block is rotated

Figure 5.4.9: Three parameters identified by the algorithm in ascending order of their L_2 minimums.

for a rotation angle with x-axis of -3 degrees. Blocks of interest of size 25×25 and centered at pixels (198,120) and at (198, 250) and their corresponding inquiry blocks **B** in cyan color of sizes 65×65 from the reference image are selected. Column v) in the second rows of Figs. 5.4.6 and 5.4.8 show that the best 8 matching blocks from the

inquiry block (in red) are positioned at the top middle location of the inquiry block, that is 22 pixels above the centre of the inquiry block and one pixel to the right from it. In other words, this means that the reference image needs to be rotated first, shifted 22 pixels down and one pixel to the left to be aligned with the test image.

5.5 Summary

In this chapter we presented an improvement of the EigenBlockCD algorithm. The improved version, namely the EigenBlockCD-2, uses a co-registration step to achieve global alignment of images. It allows the algorithms to account for large shifts and rotations. We defined the co-registration of two medical images as an optimization problem. The solution finds three non-rigid transformation parameters, two for translation and one for rotation. The algorithm then performs a linear transformation to align the first image with the second one. In the second step, the EigenBlockCD-2 algorithm computes the significant changes between medical images by employing the EigenBlockCD algorithm.

We showed the consistent performance of the co-registration step and demonstrated the results of the EigenBlockCD-2 algorithm with real MR images. The results show that the EigenBlockCD-2 performs significantly better than the earlier version the EigenBlockCD algorithm in the following aspects: It accounts for large shifts and rotations, it is computationally more efficient, and it increases the accuracy of change detection. The same conclusions can be applied to the AEDL-2 algorithm compared to the AEDL algorithm.

Chapter 6

Numerical Validation

Performance analysis of change detection methods is essential to the design of automated change detection systems for MR images. Such an analysis should take into account all possible non-significant changes. We provide a complete mathematical characterization on the performance of the EigenBlockCD-2 algorithm in the presence of such changes. We also thoroughly discuss the results of several performance measures as applied to the EigenBlockCD and the AEDL algorithms and their improved versions the EigenBlockCD-2 and the AEDL-2 algorithms [23, 24, 25]. To validate the performance of our algorithms, the obtained performance prediction, namely the ground truth, is compared to the results obtained by the algorithms with synthetic MR Images for which the ground truth is known.

In this dissertation, we have used both quantitative and qualitative performance measures utilized in previous works in [7, 8, 9, 27, 29, 42, 43]. We calculated the following performance measures: **GT/IP** (ratio of number of ground truth pixels and the total number of image pixels), **PCC** (Percentage correct classification), **Jaccard**, **Yule**, **Sensitivity**, **Specificity**, **SI** (Similarity Index and **SSIM** (Structure Similarity). We briefly describe these performance measures in this section followed by the simulations and results.

6.1 Performance Measures

Performance measures tell us how well an algorithm performs when its output is compared with an output from a well established algorithm which sometimes is called the gold standard or the ground truth. We have designed the ground truth image for validating the results. These measures are based on the unions, intersections and complements of the ground truth images and the computed change images obtained by the algorithms. The first six measures, the **PCC**, **Jaccard**, **Yule**, **Sensitivity**, **Specificity** and the **SI** are binary measures where 1 suggest correct and 0 suggest incorrect detection of change or absence of change. Binary measures are computed based on four sets of data:

We denote with **IP** the total number of image pixels, i.e., $N_1 \times N_2$ where N_1 and N_2 are the dimensions of an image, e.g., 256×256. **GT** stands for the ground truth image which contains only significant changes, i.e., the total number of true changed pixels. Computed change (**CC**) image represents the change image obtained by an algorithm, i.e., the total number of changed pixels detected by an algorithm. False positive (**FP**) represents the number of non-changed pixels incorrectly detected as change, that is, the difference between **CC** and **TP**. False negative (**FN**) is the number of changed pixels incorrectly detected as no-change, that is, the difference between **GT** and **TP**. True positive (**TP**) represents the number of pixels correctly detected as change by the algorithms, that is, the intersection of **GT** with **CC**. True negative (**TN**) represents the number of non-changed pixels correctly detected.

An illustration of these four sets is shown in Fig 6.1.1, i.e., **FP** in yellow, **FN** in magenta, **TP** in red and **TN** in white: the equations are given below:

$$\mathbf{TP} = \mathbf{GT} \cap \mathbf{CC}$$

$$\mathbf{FP} = \mathbf{CC} - \mathbf{TP}$$

$$\mathbf{FN} = \mathbf{GT} - \mathbf{TP}$$

$$\mathbf{TN} = (\mathbf{GT} \cup \mathbf{CC})^C$$

$$= (\mathbf{GT})^C \cap (\mathbf{CC})^C$$

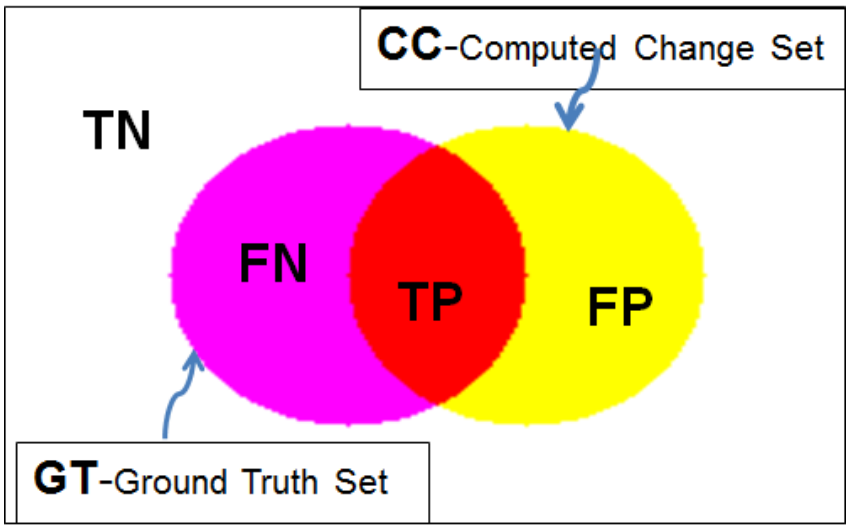

Figure 6.1.1: Venn diagram of pixel based comparison between the ground truth set (**GT**) and the computed change (**CC**).

Based on these sets we describe six quantitative measures which we apply to validate our algorithms. Percentage correct classification measure (**PCC**) is the most commonly used performance measure with values in the range $[0, 1]$. It represents the ability of the algorithm to correctly detect true changes (**TP**) and non-changes (**TN**). **PCC** values close to 1 indicate higher performance. The **PCC** ratio approaches 1 if the ground truth image contains very few changes, i.e., the ratio $\frac{\mathbf{GT}}{\mathbf{IP}} \leq 0.043\text{r}$ [29]. Therefore it is not a sufficient measure for images that have undergone little changes.

$$\mathbf{PCC} = \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{FP} + \mathbf{TN} + \mathbf{FN}} \tag{6.1.0.1}$$

The **Jaccard** measure avoids this limitation by excluding **TN** and using only the detection of changed pixels:

$$\mathbf{Jaccard} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP} + \mathbf{FN}} \tag{6.1.0.2}$$

The **Jaccard** measure gives values in $[0, 1]$. It is easy to see from Eq. (6.1.0.2) that higher values of this measure indicate higher performance of an algorithm. The **Yule** coefficient is another quantitative performance measure in $[-1, 1]$ and is given by:

$$\mathbf{Yule} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}} + \frac{\mathbf{TN}}{\mathbf{TN} + \mathbf{FP}} - 1 \tag{6.1.0.3}$$

Sensitivity measures the probability that an algorithm correctly detects true changes. The Eq. (6.1.0.4) is used to calculate the sensitivity value:

$$\textbf{Sensitivity} = \frac{\textbf{GT} \cap \textbf{CC}}{\textbf{GT}} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FN}} \tag{6.1.0.4}$$

Specificity measures the probability that an algorithm correctly detects non-changes and it is given by Eq. (6.1.0.5).

$$\textbf{Specificity} = \frac{\textbf{TN}}{\textbf{TN} + \textbf{FP}} = \frac{(\textbf{IP} - \textbf{GT}) \cap (\textbf{IP} - \textbf{CC})}{\textbf{IP} - \textbf{CC}} \tag{6.1.0.5}$$

The similarity index (**SI**) with values in [0,1] is defined as the ratio of twice the common area to the sum of individual areas:

$$\textbf{SI} = 2 \times \frac{\textbf{GT} \cap \textbf{CC}}{\textbf{GT} \cup \textbf{CC}} = 2 \times \frac{\textbf{TP}}{\textbf{TP} + \textbf{FP} + \textbf{FN}} \tag{6.1.0.6}$$

Quantitative measures described above do not take into account the structural similarities between two images. The **S**tructural **SIM**ilarity index (**SSIM**) [45] measures the similarity between two images by introducing structural information, with one of the images being compared, provided the other image is regarded as of perfect quality:

$$\textbf{SSIM}(I_1, I_2) = \frac{(2\mu_{I_1}\mu_{I_2} + C_1)(2\sigma_{I_1 I_2} + C_2)}{(\mu_{I_1}^2 + \mu_{I_2}^2 + C_1)(\sigma_{I_1}^2 + \sigma_{I_2}^2 + C_2)} \tag{6.1.0.7}$$

where $\mu_{\mathbf{I}_1}$, $\mu_{\mathbf{I}_2}$, $\sigma_{\mathbf{I}_1}$ and $\sigma_{\mathbf{I}_1}$ are the mean and the standard deviation of images $\mathbf{I}_1$ and $\mathbf{I}_2$ and $\sigma_{\mathbf{I}_1\mathbf{I}_2}$ is the covariance between $\mathbf{I}_1$ and $\mathbf{I}_2$, i.e.,

$$\mu_{\mathbf{I}_s} = \frac{1}{N_1 + N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} I_s(i,j), \text{ for } s \in \{1, 2\}$$

$$\sigma_{I_s} = \left(\frac{1}{N_1 + N_2 - 1} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (I_s(i,j) - \mu_{\mathbf{I}_s})^2 \right)^{\frac{1}{2}}, \text{ for } s \in \{1, 2\}$$

$$\sigma_{\mathbf{I_1 I_2}} = \frac{1}{N_1 + N_2 - 1} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (I_1(i,j) - \mu_{\mathbf{I_1}})(I_2(i,j) - \mu_{\mathbf{I_2}})$$

where $I_1(i,j)$ and $I_2(i,j)$ represent pixel intensity of the baseline image and its follow-up at location (i,j). $C_1 = (k_1 L)^2$, $C_2 = (k_2 L)^2$, $k_1 = 0.01$, $k_2 = 0.03$, and $L = 2^{(\#\ of\ bits\ per\ pixel)} - 1$, (i.e., $L = 255$ for gray-scale images).

6.2 Simulations and Results

In this section we describe the details of the experiments and results. First, we design the reference image. The test image is created by applying two types of changes to the reference image, significant and insignificant. We are able to control both types of changes as shown in Fig. 6.2.1. Significant changes regarded as the ground truth include enlargement, shrinkage, disappearance, intensity changes of an existing lesion, and appearance of new lesions as shown in Fig. 6.2.1 third row.

6.2.1 Simulations and Results With Synthetic Images

The reference image, as shown in Fig. 6.2.1 Column a), is created by stacking three sub-images (image strips) of size $\frac{L}{3} \times L$, where $L = 129$. Number L is chosen such that it is divisible by 3, so that L/3 is a positive integer. The reference image is of size $L \times L$. For each image strip:

- We place three polygons with vertices that are created at random within a given range.

- Each polygon is filled with intensities whose distribution is random Gaussian, using maximum intensity chosen at random in $[.45, .55]$, $[.35, .45]$ and $[.25, .35]$, respectively, from left to right.

- Four corresponding test images, as shown Fig.6.2.1 Column b), are obtained from each reference image by applying two kinds of changes:

- **Insignificant changes**: shift, rotation, and noise. These changes will mimic the changes due to patient position, contrast changes, and other acquisition related artifacts.

- horizontal and vertical shift sizes are chosen at random in the range [-3,3]. Negative values represent shifts in negative directions (i.e., up or left respectively). Positive values represent shifts in positive directions (i.e., down or right respectively).
- Rotation angles are chosen at random in the range [-3, 3]. Negative values represent clockwise direction of rotation. Positive values represent counter-clockwise direction of rotation.
- White Gaussian noise with signal-to-noise ratio (SNR) chosen at random in range [25, 55].
- Changes in intensity due to changes in field inhomogeneity.

- **Significant changes** forming the ground truth, as shown in Fig.6.2.1 Column c), include clinical changes to an existing lesions and the appearance of new lesions.

  a) We considered the following changes for an existing lesion:

  - Enlargement which mimics whether a lesion has grown between time t_1 and its follow-up at time t_2.
  - Shrinkage of a lesion showing whether a lesion has shrunk between the two examination times.
  - Disappearance of a lesion present in the previous scan but it has disappeared in the follow-up scan.
  - intensity changes of an existing shape which has clinical significance for a patient.

  b) Appearance of a new shape which mimics the appearance of a new lesion.

The ground truth image contains only the significant changes, such as enlargement, shrinkage, disappearance, and intensity changes of an existing lesion. It also contains new shapes or lesions as shown in Fig 6.2.1. We create four ground truth images corresponding to each pair (Reference, Test) images.

The change image is computed via the AEDL-2, EigenBlockCD-2, and the simple

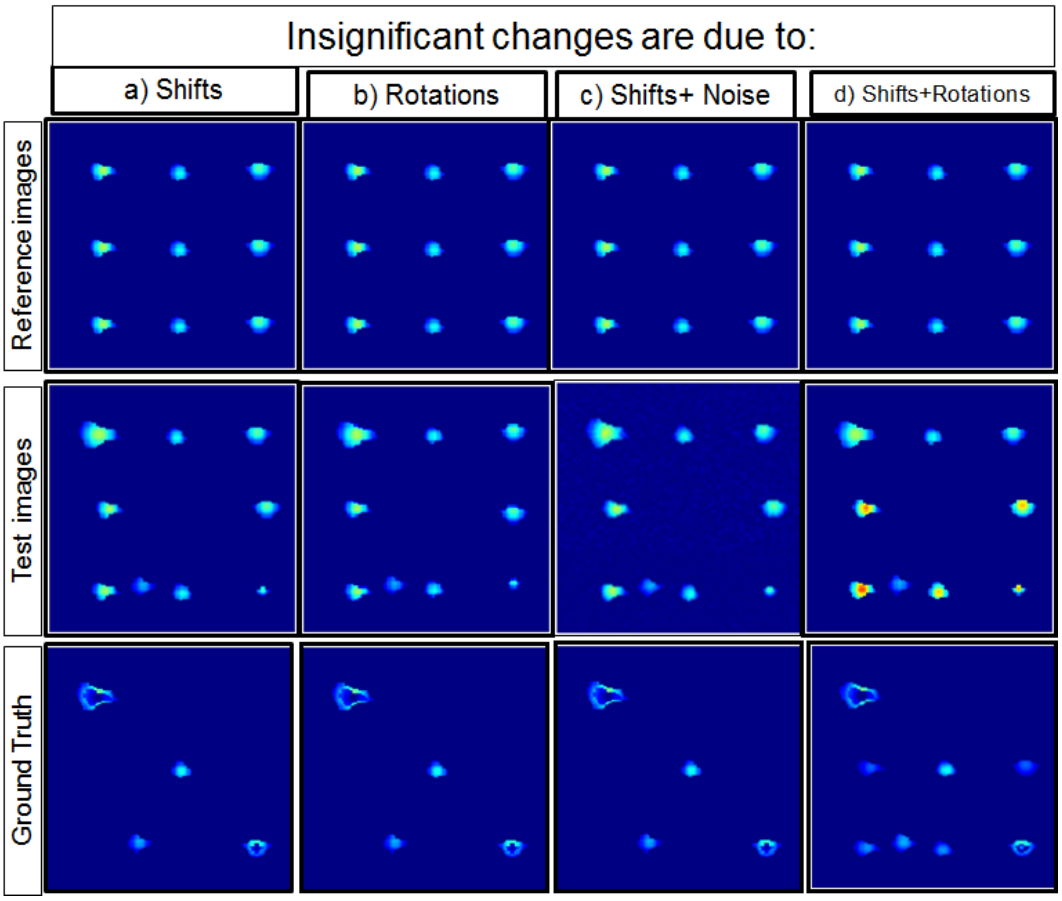

Figure 6.2.1: First row: reference images. Second row: Corresponding test images with same significant and insignificant changes. Third row: ground truth images which contains only significant changes for each pair of reference and test images. Insignificant changes are due to: Column a) shifts, Column b) rotations, Column c) shifts plus rotations, and Column d) shifts and intensity changes.

differencing algorithms for each pair of reference and test images. Performance measures are calculated by comparing the ground truth with the change image obtained from the algorithm. We calculate the following performance measures: **GT/IP** (ratio of number of ground truth pixels and the total number of image pixels), **PCC** (Percentage correct classification), **Jaccard**, **Yule**, **Sensitivity**, **Specificity**, and **SSIM** (Structure Similarity).

We ran four sets of experiments, where each experiment ran 100 times to test the performance of the AEDL-2 and the EigenBlockCD-2 algorithms in the presence of insignificant changes due to shifts, rotations, shifts + noise, and shifts + rotations changes. The change image is computed via simple differencing, the AEDL-2, and the EigenBlockCD-2 algorithms. The computed change image is compared to the ground truth based on the performance measures discussed earlier, and the results are shown

in Figs. 6.2.3, 6.2.5, 6.2.7 6.2.9. A comparison between the EigenBlockCD-2 and the simple differencing algorithms for all the performance measures described in this section is shown in Fig. 6.2.2.

The reference and test image pairs, created as described above, are different for each experiment, because of the randomness of the vertices of the polygons, the randomness of their intensities, noise level, rotation angles, translation sizes, and the level of intensity changes.

Statistical Analysis: Mean of 100 Experiments - Insignificant Changes are due to changes in:

	Shifts		Rotations		Shifts & Noise		Shifts & Rotation	
Measures	Simple Difference	EigenBlockCD-2	Simple Difference	EigenBlockCD -2	Simple Difference	EigenBlockCD -2	Simple Difference	EigenBlockCD-2
GT/IP	0.0231	0.0231	0.0295	0.0295	0.0334	0.0334	0.0405	0.0405
PCC	0.9690	0.9934	0.9857	0.9953	0.8246	0.9826	0.9810	0.9903
Jaccard	**0.3102**	**0.5809**	**0.5427**	**0.6747**	**0.3351**	**0.7377**	**0.5746**	**0.6957**
Yule	**0.3103**	**0.7383**	**0.6073**	**0.8676**	**0.1522**	**0.7342**	**0.6101**	**0.7713**
Sensitivity	**0.3381**	**0.7415**	0.6181	0.8693	**0.3412**	**0.7514**	**0.6260**	**0.7778**
Specificity	0.9961	0.9965	0.9962	0.9969	0.9940	0.9992	0.9962	0.9965
SSIM	0.9998	0.9999	0.9999	1.0000	0.9997	0.9999	0.9998	0.9999

Figure 6.2.2: Mean of the performance measures of 100 experiments. The significant changes include enlargement, shrinkage, noise, intensity changes, appearance, and disappearance of shapes. The insignificant changes are due to shifts, rotations, shifts and noise, and shifts and rotations changes.

It is difficult to provide a fair comparison among automatic change detection algorithms due to the fact that there is no gold standard and implementation of state-of-the art methods are not publicly available [11]. Work in [42] evaluated the performance of their methods on simulated lesion images and reported their SI value around 0.75.

Authors in [43] tested their lesion detection method based on segmentation to lesions generated from MRI [102] and reported their SI values on the simulated target slices with 20%, 40% and 60% intensity reduction as 0.867, 0.879 and 0.724 respectively.

The method in [42] obtained an average value of 0.879 for SI. As an overall measure of performance, they were able to achieve sensitivity = 0.877, specificity = 0.998 and similarity index (SI) = 0.879. They claim that it performs comparably with the method used by [43]and outperforms the method used by [42]. Work in [29] used three measures: the PCC, the Jaccard, and the Yule coefficients.

Authors in [7],[8] also used the Sensitivity and the Specificity measures to validate

their algorithm. They obtained values of SI in range [0.54-0.94] with P-value = 0.004 and SI values in range [0.71 - 0.96] for P-value = 0.008. The specificity values they obtained are in the range [0.87 - 1] with P-value = 0.035.

In the first 100 experiments, insignificant changes applied to test images consist of horizontal and vertical translations, which are chosen randomly in range [-3, 3]. Fig 6.2.3 shows the mean and standard deviation of these results. The mean values for the **PCC** measure are 0.9690, 0.9947 and 0.9934. These values are very high due to the low ratio of **GT/IP**. The mean values of the **Jaccard** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are respectively 0.6042 and 0.5809, which are higher than the mean value for the **Jaccard** measure 0.3102 obtained by simple differencing, and also much higher than 0.2202 value reported in [29] for the image rationing.

Statistical Analysis of 100 Experiments-
Insignificant Changes to the Test Image are due to Shifts

	Simple Differencing-mean and std of 100 trials						
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.0231	0.9690	0.3102	0.3103	0.3381	0.9961	0.9998
Std.	0.0032	0.0056	0.0501	0.0551	0.0519	0.0008	0.0001

	AEDL-2 - Algorithm-mean and std of 100 trials						
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.0231	0.9947	0.6042	0.7223	0.7254	0.9977	0.9999
Std.	0.0032	0.0012	0.0815	0.0722	0.0716	0.0006	0.0000

	EigenBlockCD-2 - Algorithm--mean and std of 100 trials						
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.0231	0.9934	0.5809	0.7383	0.7415	0.9965	0.9999
Std.	0.0032	0.0013	0.0806	0.0633	0.0629	0.0008	0.0000

Figure 6.2.3: Mean and standard deviation for the performance measures of the first 100 experiments.The significant changes include enlargement, shrinkage, noise, intensity changes, appearance and disappearance of lesions. The insignificant changes are due to shifts.

The mean values of the **Yule** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are respectively 0.7223 and 0.7383, which are higher than the mean value for the **Yule** measure 0.3103 obtained by simple differencing and much higher than 0.4783 value reported in [29] for their own Fuzzy XOR algorithm. Mean values of the **Sensitivity** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are respectively 0.7254 and 0.7415, which are higher than the mean value for the **Sensitivity** measure 0.3381 obtained by simple differencing. Our values are comparable with value

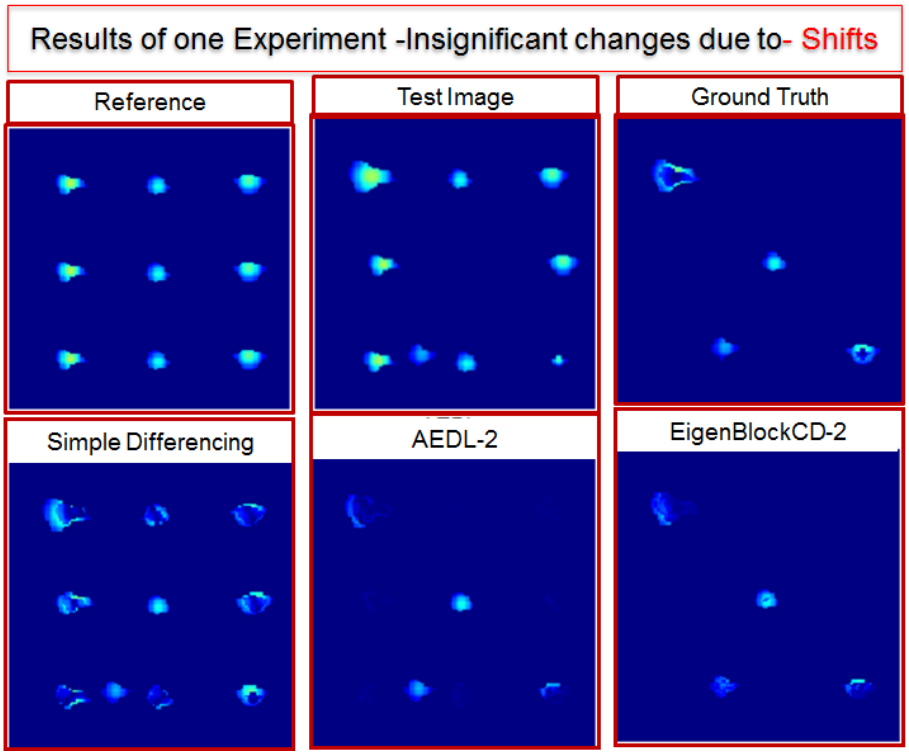

Figure 6.2.4: Computed change images from an experiment where the insignificant changes are due to shifts. First row: From left to right- the reference, test, and ground truth images. Second row: from left to right the computed change images via simple differencing, the AEDL-2, and the EigenBlockCD-2 algorithms.

0.877 reported in [11] on their algorithm.

The mean values of the **Specificity** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are respectively 0.9977 and 0.9965, which are comparable with 0.998 value reported in [11]. Our values are also comparable with values in range [0.87 - 1] reported in [7],[8] with P-value = 0.035. Finally, the mean value of the **SSIM** index is almost one for both the AEDL-2 and the EigenBlockCD-2 algorithms.

This shows that the computed change images from both the AEDL-2 and the EigenBlockCD-2 algorithms have similar structure with the ground truth as Fig 6.2.4 shows. In other words, this means that both algorithms detect significant clinical changes from one scan to another while ignoring the insignificant changes caused by shifts.

In the second 100 experiments, insignificant changes applied to test images consist of rotations with rotation angles chosen randomly in the range [-3, 3]. Fig 6.2.5 shows the mean and standard deviation of these results. The mean values for the **PCC** measure are 0.9857, 0.9981 and 0.9953. These high values are due to the low ratio of

GT/IP which is 0.0295. The mean values of the **Jaccard** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.8102 and 0.6747, respectively, which are higher than the mean value for the **Jaccard** measure 0.5427 obtained by simple differencing and much higher then 0.2202 value reported in [29] for the image rationing.

Statistical Analysis of 100 Experiments- Insignificant Changes to the Test Image are due to Rotations							
Simple Differencing-mean and std of 100 trials							
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.0295	0.9857	0.5427	0.6073	0.6181	0.9962	0.9999
Std.	0.0035	0.0039	0.0756	0.0882	0.0851	0.0007	0.0000
AEDL-2 - Algorithm-mean and std of 100 trials							
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.0295	0.9981	0.8102	0.9773	0.9775	0.9983	1.0000
Std.	0.0035	0.0005	0.0355	0.0346	0.0342	0.0003	0.0000
EigenBlockCD-2 - Algorithm--mean and std of 100 trials							
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.0295	0.9953	0.6747	0.8676	0.8693	0.9969	1.0000
Std.	0.0035	0.0014	0.0577	0.0788	0.0775	0.0005	0.0000

Figure 6.2.5: Mean and standard deviation for the performance measures of the second 100 experiments.The significant changes include enlargement, shrinkage, noise, intensity changes, appearance and disappearance of shapes. The insignificant changes are due to rotations.

The mean values of the **Yule** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.9773 and 0.8676 respectively which are higher than the mean value for the **Yule** measure 0.6073 obtained by simple differencing and much higher than 0.4783 value reported in [29] for their own Fuzy XOR algorithm. Mean values of the **Sensitivity** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.9775 and 0.8693 respectively which are higher than the mean value for the **Sensitivity** measure 0.6181 obtained by simple differencing and are also higher than with 0.877 value reported in [11].

The mean values of the **Specificity** measure for the AEDL-2 and the EigenBlockCD-2 algorithms, are 0.9983 and 0.9969 respectively which are comparable with 0.998 value reported in [11] and with values in range [0.87 - 1] reported in [7],[8] with P-value = 0.035. Finally, the mean value of the **SSIM** index is one for both the AEDL-2 and the EigenBlockCD-2 algorithms. This shows that the probability of having similar structure between the computed change images from both algorithms and the ground truth

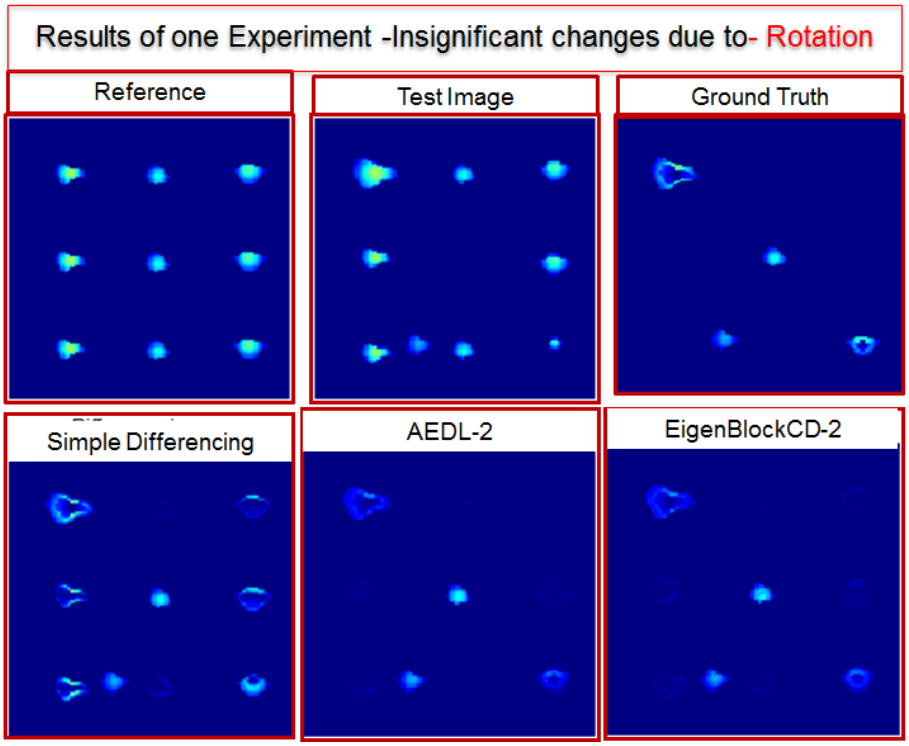

Figure 6.2.6: Computed change images from an experiment with rotations as the insignificant changes. First row: From left to right- the reference, test and the ground truth images. Second row: from left to right the computed change images via simple differencing, the AEDL-2 and the EigenBlockCD-2 algorithms.

is almost one. In other words, the computed change and the ground truth images look very similar as Fig 6.2.6 shows.

In the third 100 experiments, insignificant changes applied to test images consist of horizontal and vertical shifts and also additive white Gaussian noise with signal to noise ratio in the range [25, 55]. Fig 6.2.7 shows the mean and standard deviation of these results. The mean values for the **PCC** measure are very high, 0.9555 and 0.9826, for both algorithms due to the low ratio of 0.0231 of **GT/IP**. The mean values of the **Jaccard** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.6360 and 0.7377 respectively which are higher than the mean value for the **Jaccard** measure 0.351 obtained by simple differencing and much higher then value 0.2202 reported in [29] for the image ratioing.

The mean values of the **Yule** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.6753 and 0.7342 respectively which are higher than the mean value 0.15522 for the **Yule** measure obtained by simple differencing and much higher than 0.4783 value reported in [29] for their own Fuzy XOR algorithm. Mean values of

Statistical Analysis of 100 Experiments- Insignificant Changes to the Test Image are due to shifts + noise							
Simple Differencing-mean and std of 100 trials							
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.0334	0.8246	0.3351	0.1522	0.3412	0.9940	0.9997
Std.	0.0032	0.0436	0.0527	0.0879	0.0533	0.0022	0.0001
AEDL-2 - Algorithm-mean and std of 100 trials							
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.0334	0.9555	0.6360	0.6753	0.7181	0.9969	0.9999
Std.	0.0032	0.0787	0.2719	0.3887	0.3108	0.0011	0.0000
EigenBlockCD-2 - Algorithm--mean and std of 100 trials							
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.0334	0.9826	0.7377	0.7342	0.7514	0.9992	0.9999
Std.	0.0032	0.0468	0.1133	0.1558	0.1126	0.0009	0.0000

Figure 6.2.7: Mean and standard deviation for the performance measures of the third set of 100 experiments. The significant changes include enlargement, shrinkage, noise, intensity changes, appearance and disappearance of shapes. The insignificant changes are due to shifts and noise.

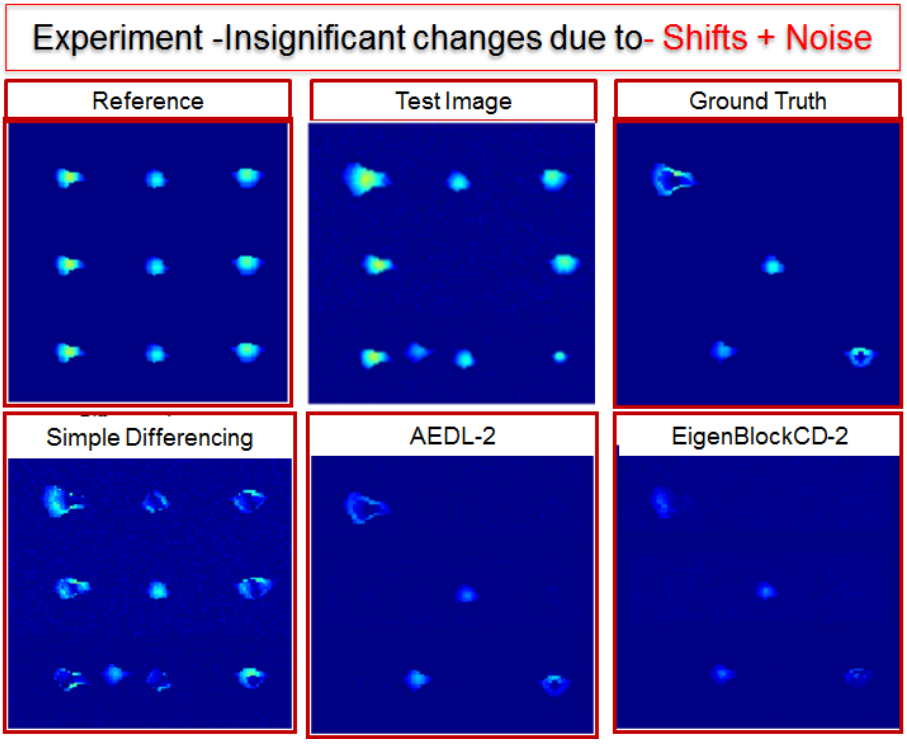

Figure 6.2.8: Computed change images from an experiment where the insignificant changes are due to shifts and noise. First row: From left to right- the reference, test and the ground truth images. Second row: from left to right the corresponding computed change images via simple differencing, the AEDL-2 and the EigenBlockCD-2 algorithms.

the **Sensitivity** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.7181 and 0.7514 respectively which are higher than the mean value 0.6181 for the **Sensitivity** measure obtained by simple differencing and they are comparable with

0.877 value reported in [11].

The mean values of the **Specificity** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.9969 and 0.9992 respectively which are comparable with 0.998 value reported in [11] and with values in range [0.87 - 1] presented in [7],[8] with P-value = 0.035. Finally, the mean value of the **SSIM** index is 0.9999 for both the AEDL-2 and the EigenBlockCD-2 algorithms. This shows that change images computed are very close to the ground truth image in terms of their structure as Fig 6.2.8 shows.

In the fourth 100 experiments, insignificant changes applied to test images consist of horizontal and vertical shifts and also rotations. Intensity of some shapes is increased or decreased randomly chosen in [0.20, 0.50]. We consider increases in intensity of a lesion as significant changes because they have clinical significance. Fig 6.2.9 shows the mean and standard deviation of these results. The mean values for the **PCC** measure are 0.9810, 0.9946 and 0.9903. These very high values are due to the low **GT/IP** ratio of 0.0405. The mean values of the **Jaccard** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.6092 and 0.7957 respectively which are higher than the mean value 0.5746 for the **Jaccard** measure obtained by simple differencing and much higher then 0.2202 value reported in [29] for the image ratioing.

Statistical Analysis of 100 Experiments-
Insignificant Changes to the Test Image are due to shifts + rotations

	Simple Differencing-mean and std of 100 trials						
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.0405	0.9810	0.5746	0.6101	0.6260	0.9962	0.9998
Std.	0.0057	0.0036	0.0533	0.0580	0.0554	0.0008	0.0001

	AEDL-2 - Algorithm-mean and std of 100 trials						
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.0405	0.9946	0.6092	0.7284	0.7314	0.9977	0.9999
Std.	0.0057	0.0013	0.0814	0.0704	0.0698	0.0006	0.0000

	EigenBlockCD-2 - Algorithm--mean and std of 100 trials						
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.0405	0.9903	0.6957	0.7713	0.7778	0.9965	0.9999
Std.	0.0057	0.0019	0.0506	0.0468	0.0457	0.0007	0.0000

Figure 6.2.9: Mean and standard deviation for the performance measures of the fourth 100 experiments.The significant changes include enlargement, shrinkage, noise, intensity changes, appearance and disappearance of shapes. The insignificant changes are due to shifts and rotations.

The mean values of the **Yule** measure for the AEDL-2 and the EigenBlockCD-2

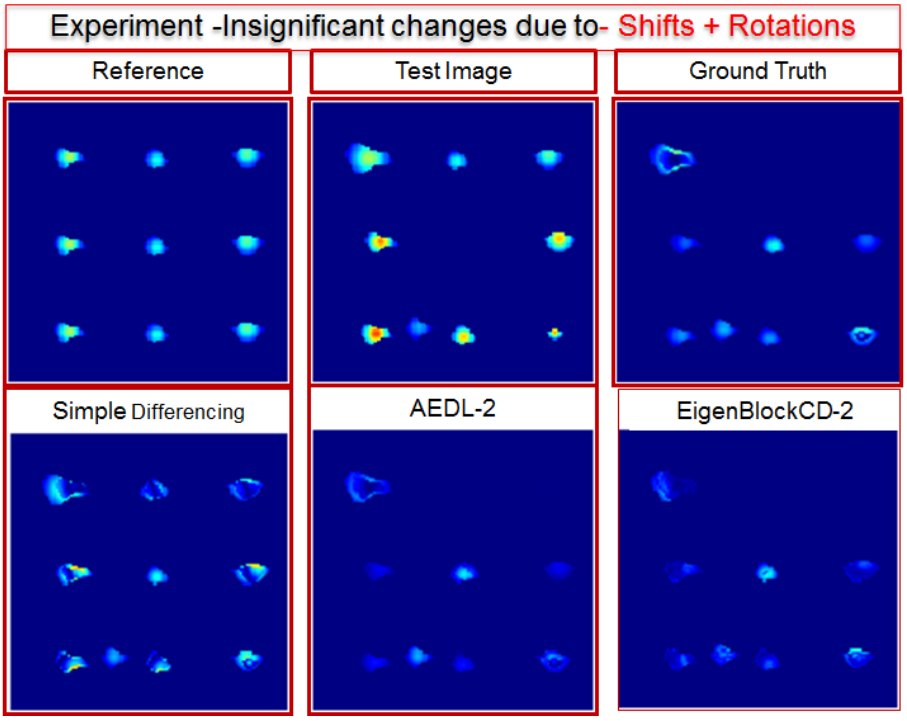

Figure 6.2.10: Computed change images from an experiment where the insignificant changes are due to shifts and rotations. First row: From left to right- reference, test, and ground truth image. Second row: from left to right the computed change image via simple differencing, the AEDL-2 and the EigenBlockCD-2 algorithms.

algorithms are 0.7284 and 0.7713 respectively which are higher than the mean value 0.6101 for the **Yule** measure obtained by simple differencing and much higher than 0.4783 value reported in [29] for their own Fuzy XOR algorithm. The mean values of the **Sensitivity** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.7314 and 0.7778 respectively which are higher than the mean value 0.6260 for the **Sensitivity** measure obtained by simple differencing and they are comparable with 0.877 value reported in [11].

The mean values of the **Specificity** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.9977 and 0.9965 respectively which are comparable with 0.998 value reported in [11] and with values in range [0.87 - 1] obtained in [7],[8] with P-value = 0.035. Finally, the mean value of the **SSIM** index is 0.9999 for both the AEDL-2 and the EigenBlockCD-2 algorithms. This means that our methods computes change images very close to the ground truth image in terms of their structure as Fig 6.2.10 shows.

6.2.2 Simulations and Results With Serial MR Images

We created a series of images from real MR images. In order to validate our change detection methods quantitatively and qualitatively, we simulated lesions and applied them to real MR images. Because the exact sizes and locations of each lesion are known, we regard them as the ground truths. Fig 6.2.11 shows an example of a serial MR images created from a T2-weighted real MR scan, namely the reference image.

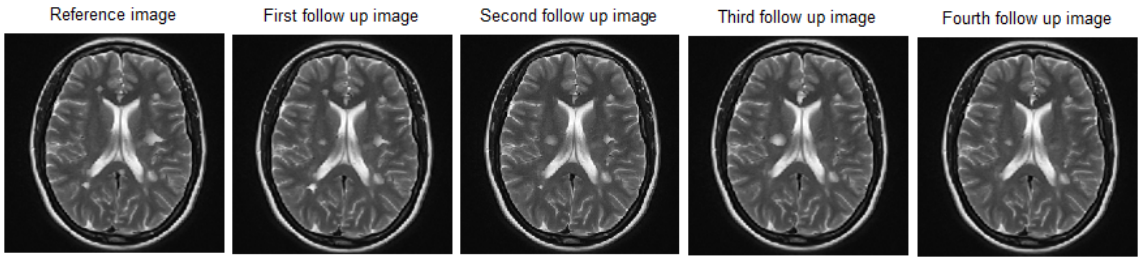

Figure 6.2.11: From left to right: Reference, first, second, third and fourth follow-up images, respectively.

Taking into the consideration what a radiologist might be looking into a follow-up MR scan, we simulated seven lesions by applying two kinds of changes: insignificant and significant changes. Significant changes, which make up the ground truth, include enlargement, shrinkage, disappearance, intensity changes of an existing lesion, and also the appearance of a new lesion. Four ground-truth images corresponding to pairs (reference, first follow-up), (first follow-up, second follow-up), (second follow-up, third follow-up) and (third follow-up, fourth follow-up) are designed as shown in Fig.6.2.11.

Insignificant changes include horizontal and vertical shifts, clockwise and counterclockwise rotations, changes in contrast and different noise level. Horizontal and vertical shift sizes are chosen at random in the range [-5,5], rotation angles are chosen at random in the range [-5, 5], and white Gaussian noise is chosen at random with signal-to-noise ratio in range [25, 55]. Intensity value are chosen randomly in [0.2, 0.5] and coordinates of the vertices for each polygon are generated randomly.

From the 7 simulated lesions, six lesions are inserted into the reference image, labeled with numbers from 1 to 6. Lesion 7 will appear in the first follow-up. Some of these shapes act as inactive lesions and the other remaining shapes mimic active lesions. Fig 6.2.12 and table 6.2.2.1 show in more details changes between each pair of two consecutive serial MR images. Lesion 1 is inactive in the first follow-up, disap-

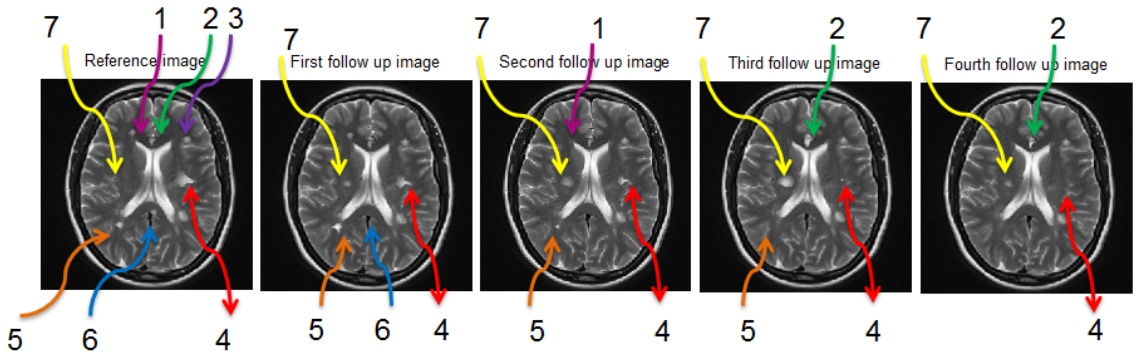

Figure 6.2.12: From left to right: The reference, the first, the second, the third and the fourth follow up images respectively.

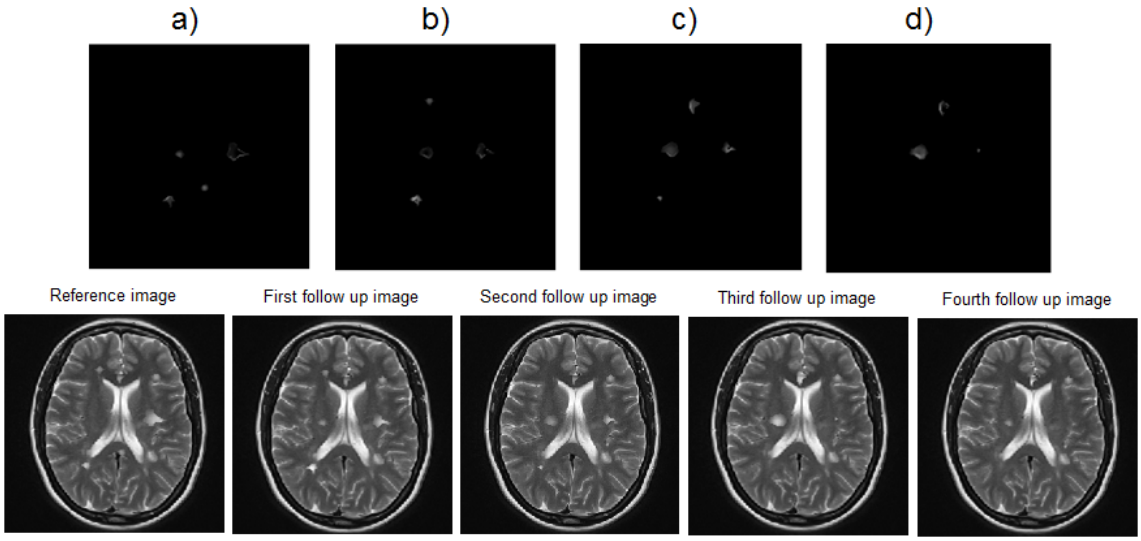

Figure 6.2.13: Top row shows the ground truth images with significant changes between two consecutive images: a) the reference and the first follow-up images, b) the first and the second follow- ups. c) the second and the third follow-ups. d) the third and the fourth follow-ups, Bottom row from left to right: the reference, the first, the second, the third, and the fourth follow-up images, respectively.

pears in the second and it remains gone. Lesion 2 stays inactive in the first and second follow-ups, then it grows for the next two follow-ups. Lesion 3 is inactive and stays inactive during all five scans. Lesion 4 shrinks in the next three scans until it disappears in the last scan. Lesion 5 grows in the first follow-up and then it shrinks until it disappears in the last scan. Lesion 6 is present in the reference image, disappears in the next follow-up and remains gone in the rest of the scans. Lesion 7 appears as a new lesion in the first follow-up, grows in the next two scans and then shrinks in the last scan.

We performed 48 experiments with both significant and insignificant changes applied to test images consisting of horizontal and vertical shifts, rotations, intensity

Table 6.2.2.1: Description of simulated lesions in a series of MR images

Images/ lesion	Reference image	First follow up	Second follow up	Third follow up	Fourth follow up
Significant changes		four changes	four changes	four changes	three changes
lesion 1	first inactive then disappears	inactive	disappears	stays disappeared	stays disappeared
lesion 2	inactive which grows	inactive	inactive	grows with increased intensity	grows more with increased intensity
lesion 3	inactive lesions	inactive	inactive	inactive	inactive
lesion 4	shrinks until it disappears	shrinks	shrinks further	shrinks more	disappears
lesion 5	grows shrinks disappears	grows with increased intensity	shrinks	disappears	stays disappeared
lesion 6	disappears and stays disappeared	disappears	stays disappeared	stays disappeared	stays disappeared
lesion 7	appears grows then shrinks	appears as a new lesion	grows with increased intensity	grows more	shrinks

changes, enlargements of lesions, shrinkage, appearance of a new lesion, and disappearance of an existing one. Fig 6.2.14 shows the mean and standard deviation of these results. We compared our results with results of several algorithms as presented by their authors [7, 8, 25, 29] shown in Table 6.2.2.2. The mean values for the **PCC** measure are 0.9989 and 0.99883 for the AEDL-2 and the EigenBlockCD-2 algorithms which are higher than 0.35952 obtained by the simple differencing. The values of the **PCC** measures are very high due to the low **GT/IP** ratio of 0.00822. The mean values of the **Jaccard** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.7653 and 0.7558 respectively which are much higher than the mean value 0.03267 for the **Jaccard** measure obtained by simple differencing and much higher then 0.2202 value reported in [29] for the image rationing.

The mean values for the **Yule** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.75467 and 0.76532 respectively which is higher than the mean value -0.62180 for the **Yule** measure obtained by simple differencing and much higher than 0.4783 value reported in [29] for their own Fuzy XOR algorithm. The mean values of **Sensitivity** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.7735

Statistical Analysis: Mean and Standard Deviation of 48 Experiments with Serial MRI Images							
	Simple Differencing-mean and std of 48 trials						
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.00822	0.35952	0.03267	-0.62180	0.03267	1.00000	0.99832
Std.	0.00175	0.02584	0.01454	0.03302	0.01454	0.00000	0.00056
	AEDL-2 - Algorithm-mean and std of 48 trials						
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.0822	0.9989	0.7653	, 0.76532	0.7735	0.9999	1.0000
Std.	0.0018	0.0004	0.0893	0.0953	0.0932	0.0000	0.0000
	EigenBlockCD-2 - Algorithm-mean and std of 48 trials						
	GT/IP	PCC	Jaccard	Yule	Sensitivity	Specificity	SSIM
mean	0.00822	0.99883	0.75584	0.75467	0.75584	1.00000	0.99999
Std.	0.00175	0.00049	0.09695	0.09733	0.09695	0.00000	0.00000

Figure 6.2.14: The mean and the standard deviation for the performance measures of 48 experiments. The significant changes include enlargement, shrinkage, noise, intensity changes, appearance and disappearance of shapes. The insignificant changes are due to shifts, rotations and noise level.

and 0.75584 respectively which are higher than 0.03267 the mean value for **Sensitivity** measure obtained by simple differencing and they are comparable with 0.877 value reported in [11].

The mean values of the **Specificity** measure for the AEDL-2 and the EigenBlockCD-2 algorithms are 0.9999 and 1.0000 respectively which are comparable with 0.998 value reported in [11] and with values obtained in [7, 8] which are in range [0.87 - 1] with P-value = 0.035. Finally, the mean value of the **SSIM** index are 1.000 and 0.9999 for the AEDL-2 and the EigenBlockCD-2 algorithms. This means that our methods compute change images very close to the ground truth images in terms of their structure as Fig 6.2.15 shows.

6.3 Summary

In this chapter we presented a thorough performance analysis of the AEDL-2 and EigenBlockCD-2 algorithms. Our analysis was both qualitative and quantitative. We used well known performance measures such as the PCC, the Jackard, the Yule, the Sensitivity, the Specificity, and the SSIM coefficients. We ran a significant number of experiments and compared our results to those published in the related literature. The simulations showed that our both algorithms, the AEDL-2 and the EigenBlockCD-2

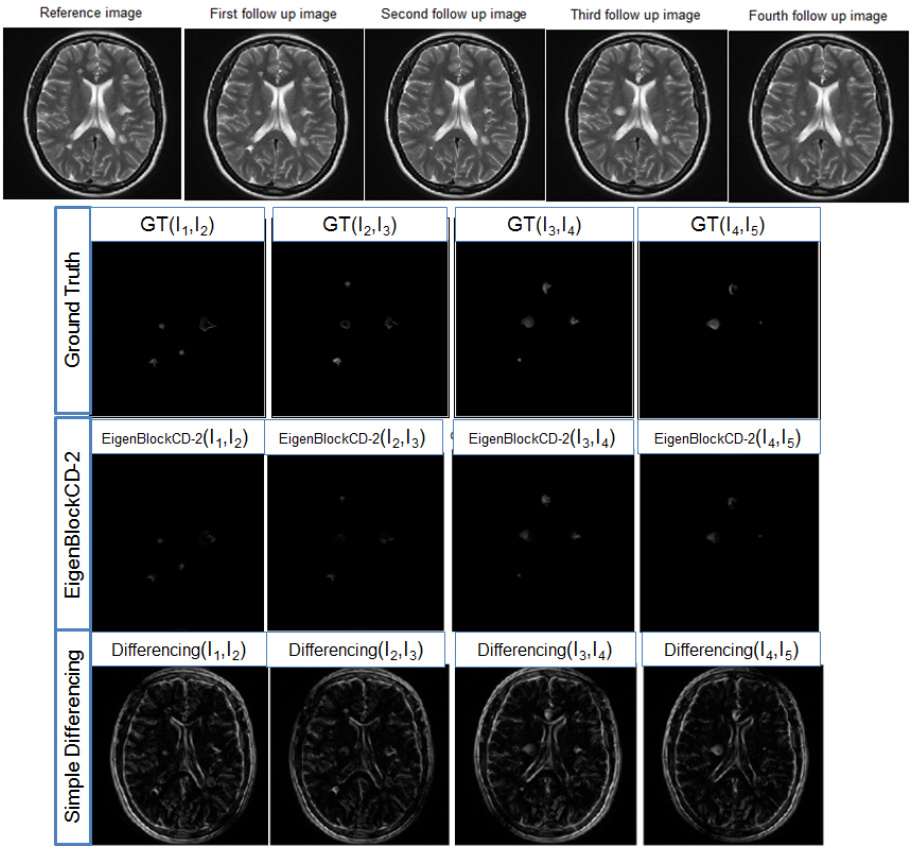

Figure 6.2.15: Serial MRI results. First row: the reference image and its four follow-ups. Second row: the corresponding ground truth images for each change image computed. Third row: the computed change images in serial MRI via the EigenBlockCD-2 algorithm. Fourth row: the computed change images in serial MRI via simple differencing method.

algorithms, perform better than the previous algorithms cited in the previous sections. We included both synthetic and real MR images in our simulations. We also created a series of MR images from real MR scans.

Table 6.2.2.2: Performance measures comparing algorithms presented in [29, 7, 8, 25, 26].

Algorithms	**PCC**	**Jaccard**	**Yule**	**Sensitivity**	**Specificity**	**SSIM**
Image differencing [29]	0.8082	0.1888	0.4451			
Image rationing [29]	0.8054	0.2202	0.4072			
Image regression [29]	0.8024	0.1861	0.3971			
CVA [29]	0.8081	0.1809	0.4512			
Fuzzy XOR [29]	0.8023	0.1044	0.4783			
Background Sub [29]	0.7945	0.2125	0.4285			
CD-Change Detector [7, 8]				[.68, .94]	[.91, 1.0]	
CS-Classify Subtract [7, 8]				[.55, .65]	[.80, .98]	
AEDL-2 [25, 26]	0.9989	0.7653	0.7653	0.755	1.0	1.0
EigenBlockCD-2 [25, 26]	0.9988	0.7558	0.7547	0.7735	0.999	0.9999

Chapter 7

Extensions to Volumetric Imaging Data

In this section we extend the EigenBlockCD-2 and AEDL-2 algorithms to detect the changes of 3D volumetric data. This is motivated by the fact that MR images are originally volumes in 3D as shown in Fig. 7.1.1. Similar to the 2D case, we consider the 3D volumes of the same anatomical objects taken at two different times. The obtained image slices are generally misaligned due to patient position and other acquisition related artifacts. We define the problem of change detection in 3D and present the EigenBlockCD-2 algorithm for volumetric data. The algorithms include two steps: 1) the interpolation step for which a transformation model is created to describe the alignment process and 2) the detection step of 3D change detection algorithm. The algorithms are tested with real MR images.

7.1 Problem Formulation And Notation

MR imaging provides a very detailed anatomical view of soft tissues such as human brain and is the best among all structural imaging modalities. As such, MRI can be used to detect brain tumors, multiple sclerosis, stroke etc. An original MR image produced by MRI machine software depicts the anatomic variations of signal intensity in thin slices as shown in Fig. 7.1.1. As a result, a set of images, each describing a slice of the object (brain, etc), is obtained and stored by MRI software in a special format.

Geometrically speaking, a 3D image obtained from MRI scans is a volume of voxels. A voxel or a volume element is the 3D corresponding term for a pixel in 2D image. An automated change detection system compares two 3D volumes from a patient taken at two different times.

The extension of the EigenBlockCD-2 and the AEDL-2 algorithms to 3D case takes into consideration a new transformation model, a interpolation step, a similarity measure, and an optimization method which are the usual steps for any medical registration software package [97, 98, 99, 100].

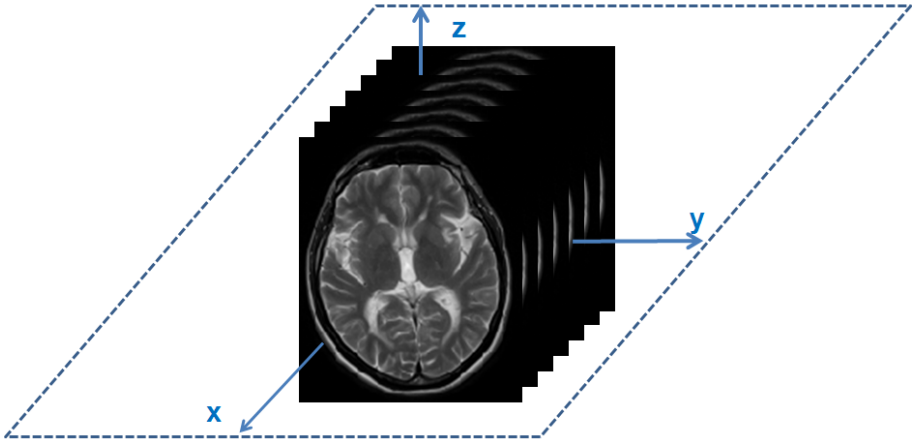

Figure 7.1.1: An illustration of a 3D MR volume of a brain

We extend the notations we used in the 2D case to the 3D case:

- $\mathbf{V}_1$ and $\mathbf{V}_2$ represent volumes of the same anatomical structure corresponding to times t_1 and t_2 respectively. $\mathbf{V}_1$ and $\mathbf{V}_2$ are referred to as the reference and the test volume.

- The block $\mathbf{b}_{ijk}$ from the test volume is centered at voxel (i, j, k) and it is referred to as the **Block of Interest**. For simplicity we denote it by $\mathbf{b}$.

- The Δ neighbourhood of (i, j, k) voxel, denoted by $\mathbf{B}$, in reference volume is called the "**Inquiry Block**".

$$\mathbf{B} = \left\{ \mathbf{V}_1(\tilde{i}, \tilde{j}, \tilde{k}) \colon \tilde{i} \in [i - \Delta, i + \Delta], \tilde{j} \in [j - \Delta, j + \Delta], \tilde{k} \in [k - \Delta, k + \Delta] \right\}$$

- $\mathcal{D}$ is the set of all blocks $\mathbf{a}_l$ of the same size as the block of interest $\mathbf{b}$ from the

inquiry block $\mathbf{B}$.

$$\mathcal{D} = \left\{ \mathbf{a}_l \colon l = 1, 2, ...(2\Delta + 1)^3 \right\}$$

- Blocks $\mathbf{a}_l$ from the inquiry block are referred to as the "**Training Blocks**". Blocks $\mathbf{a}_l$ are of sizes $(2\Delta + 1) \times (2\Delta + 1) \times (2\Delta + 1)$ and centered at (p, q, s), where $l = (s - 1) \times (2\Delta + 1)^2 + (q - 1) \times (2\Delta + 1) + p$.

- Let $\Phi = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k, \ldots, \mathbf{x}_n]$ be the dictionary formed by stacking all the training blocks $\mathbf{a}_j$ to column vectors $\mathbf{x}_j$. Fig. 7.1.2 shows two 3D volumes, a block of interest $\mathbf{b}$ stacked as a column vector $\mathbf{y}$ from the test volume, and the training blocks $\mathbf{a}_l$ stacked as column vectors $\mathbf{x}_j$. Training blocks $\mathbf{a}_l$ are extracted from the inquiry block $\mathbf{B}$.

- Let $m = (2\delta + 1)^3$ and $n = (2\Delta + 1)^3$. Then, the block of interest $\mathbf{b}$ and each training block $\mathbf{a}_l$ are of sizes $m^{\frac{1}{3}} \times m^{\frac{1}{3}} \times m^{\frac{1}{3}}$. The inquiry block $\mathbf{B}$ is of size $n^{\frac{1}{3}} \times n^{\frac{1}{3}} \times n^{\frac{1}{3}}$.

To achieve a more precise alignment in the 3D case we need to generate more slices between the existing ones. The 3D anatomical data sets we have used in our simulations are of size $384 \times 384 \times 20$, i.e., 20 slices of size 384×384 each and with a slice thickness of 5 mm. Between every two slices we insert four slices by applying cubic spline interpolation, thus, obtaining a volume of size $384 \times 384 \times 96$. In the co-registration step, our algorithms aligns the baseline volume with its follow-up first. After the two volumes are aligned, the algorithm extracts the 20 original slices from the volume that hasn't been transformed and the corresponding 20 aligned slices from the transformed volume.

We make similar assumptions as in the 2D case. That is, two consecutive volumes differ from one another due to disease related changes, shifts, rotations and noise. We assume that the background of each block from the test volume can be learned from the reference volume blocks. Each block of interest can be sparsely approximated using a linear combination of a few blocks from the reference volume over a dictionary Φ. Therefore we express $\mathbf{b}$ as a linear combination of blocks $\mathbf{a}_l$ by the following equation:

$$\mathbf{b} = \sum_{l=1}^{(2\Delta+1)^3} \boldsymbol{\gamma}_l \mathbf{a}_l = \sum_{l=1}^{n} \boldsymbol{\gamma}_l \mathbf{a}_l, \quad \mathbf{a}_l \in \mathcal{D} \tag{7.1.0.1}$$

where $l \in \{1, 2, \ldots, (2\Delta + 1)^3\}$

Blocks $\mathbf{a}_l$ and $\mathbf{b}$ are stacked as $m \times 1$ column vectors $\mathbf{x}_l$ and $\mathbf{y}$, respectively, where $m = (2\delta + 1)^3$ and δ is a positive integer representing the radius of these blocks. Eq. (7.1.0.1) can be written as:

$$\mathbf{y} = \sum_{l=1}^{n} \boldsymbol{\gamma}_l \mathbf{x}_l, \quad l \in \{1, 2, \ldots, n\} \tag{7.1.0.2}$$

Eq. (7.1.0.2) in matrix form can be written as:

$$\mathbf{y} = \Phi\boldsymbol{\gamma} + \mathbf{n} \tag{7.1.0.3}$$

where $\mathbf{n}$ accounts for noise.

The reference and test volumes are not necessarily aligned. Fig. 7.1.4 a) and b) show that the tenth slices from the reference and the test volumes are not aligned and do not represent the same anatomical structure, meaning that they do not represent pictures of the same scene taken at different times. To accurately detect the relevant clinical changes, we first perform the global initial alignment between the reference and the test volumes. After the co-registration step, the tenth slice from the test volume and its corresponding tenth transformed slice from the reference volume are aligned as shown in Fig. 7.1.4 b) and c), respectively. The algorithms used in 2D can be easily extended for 3D volumes which we present below. Next, we discuss the co-registration step followed by change detection step. In the end we provide a summary of the EigenBlockCD-2 algorithm for volumetric data.

7.1.1 Initial Global Alignment of 3D Volumes

As in Chapter 6, the co-registration problem finds the transformation parameters that best aligns the two volumes by determining a transformation space and a cost function that quantifies the quality of alignment. In our 3D registration problem, the transfor-

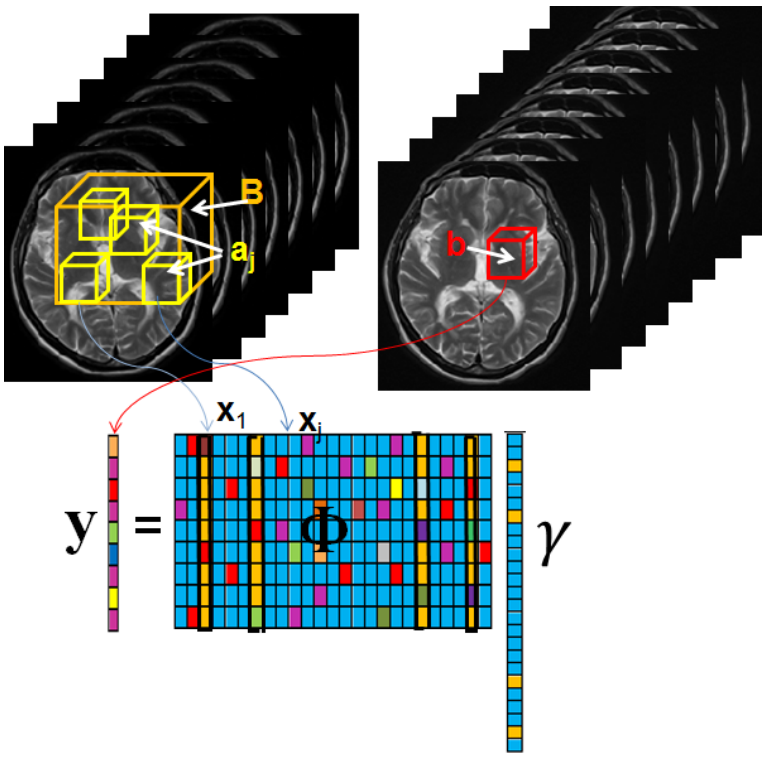

Figure 7.1.2: An illustration of two 3-dimensional volumes, a block of interest $\mathbf{b}$ from the test volume, $\mathbf{b}$ is stacked as a column vector $\mathbf{y}$, its inquiry block $\mathbf{B}$ from the reference volume, $\mathbf{B}$ contains overlapping training blocks $\mathbf{a}_j$ which are then stacked as column vectors $\mathbf{x}_j$ of the dictionary Φ.

mation space includes six rigid transformations with six degrees of freedom, i.e., three parameters for translations and three for rotations. Fig. 7.1.3 illustrates the three rotation parameters to be determined by the co-registration step of our algorithm. As with the EigenBlockCD-2 algorithm, we use the L_2 norm as the cost function. The minimization of the cost function finds the optimal parameters for which the changes between the two consecutive volumes are clinically relevant and disease related.

As in Chapter. 6, we let $\mathcal{T}$ to be a set of transform functions, $\mathcal{P}$ a set of transformation's parameters and $\mathcal{C}$ the cost function to be minimized. For any two 3D volumes of the same anatomical part, the registration problem finds six parameters such that the two volumes are to be geometrically aligned after transforming one or both volumes using these parameters. The cost function checks the accuracy of the alignment, that is, maximizing the accuracy of the co-registration by minimizing the cost function, minimizing the norm of the difference between one volume and the second volume transformed.

Suppose that the reference volume has to be rotated by angles α, β and γ to be

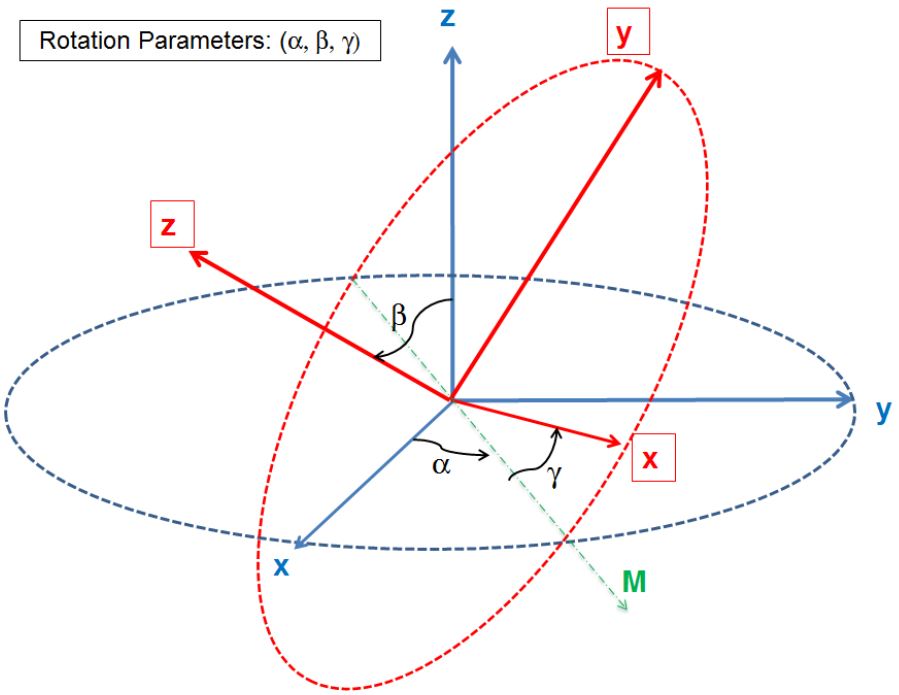

Figure 7.1.3: An illustration of the three rotation parameters (α, β, γ) to be determined by the extended EigenBlockCD-2 algorithm for 3D volumes during the co-registration step.

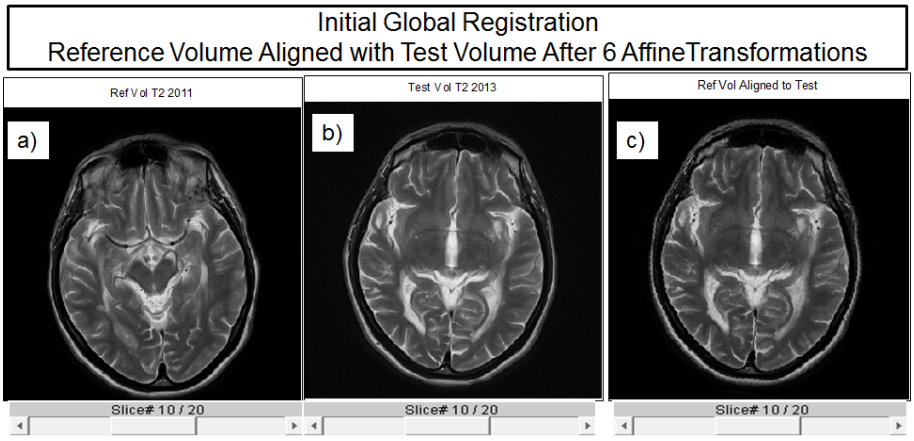

Figure 7.1.4: a) and b) The reference and the test volumes of real T2-weighted volumes of a brain taken in 2011 and in 2013, respectively. c) The reference volume aligned with the test volume after six affine transformations.

aligned with the test volume. This is equivalent of rotating x, y and z-axis by angles α, β, and γ. In Fig. 7.1.3, if line M (in green) represents the intersection of the xy-plane (in blue) and its rotated xy-plane (in red), then the angle α is the angle between the x-axis and the line M, which can be interpreted as rotation around z-axis. Angle β is the angle between the two z-axis, (in blue and red), and γ is the angle between the new x-axis and line M. The co-registration problem can be written as a minimization

problem:

$$\min_{\mathcal{T},\mathcal{P}} \mathcal{C}\big(\mathcal{T}, \mathbf{V}_1, \mathbf{V}_2, \mathcal{P}\big) \tag{7.1.1.1}$$

where:

- $\mathbf{V}_1, \mathbf{V}_2$ represent the two volumes to be aligned.
- $\mathcal{C}$ is the cost function in the minimization problem, i.e., the L_2 norm given by $\mathcal{C} = ||.||_2$.
- $\mathcal{T}$ is the set of transform functions; in our case $\mathcal{T}$ is any composition of transform functions from the set $\{\mathbf{T}(t_x, t_y, t_z), \mathbf{R}(\alpha, \beta, \gamma\}$, where $\mathbf{T}$ are translation transforms and $\mathbf{R}$ are rotational transform functions.
- $\mathcal{P} = \{(t_x, t_y, t_z, \alpha, \beta, \gamma)\} : \ t_x \in \mathbf{T}, t_y \in \mathbf{T}, t_z \in \mathbf{T}, \alpha \in \mathbf{R}, \beta \in \mathbf{R}$ and $\gamma \in \mathbf{R}$, is a set of 6-tuples $(t_x, t_y, t_z, \alpha, \beta, \gamma)$ representing the six parameters needed to align the two volumes, i.e., the sizes of the three shifts in x, y and z directions and the sizes of the three rotation angles, α, β and γ.

Let $\mathbf{V}(i, j, k)$ represent the intensity of a volume voxel at location (i, j, k). Image $\mathbf{V}$ is of size $N_1 \times N_2 \times N_3$. The translation transform functions can be applied to each volume voxel as follows:

$$\mathbf{T}_{x,y,z}\big(\mathbf{V}(i, j, k)\big) = \mathbf{V}(i + t_x, j + t_y, k + t_z)$$

$$\mathbf{R}_{\alpha}\big(\mathbf{V}(i, j, k)\big) = \mathbf{V}\Big((i - \frac{N_1}{2})\cos\alpha - (j - \frac{N_2}{2})\sin\alpha, (i - \frac{N_1}{2})\sin\alpha - (j - \frac{N_2}{2})\cos\alpha, k\Big)$$

The formulas for $\mathbf{R}_{\beta}\big(\mathbf{V}(i, j, k)\big)$ and $\mathbf{R}_{\gamma}\big(\mathbf{V}(i, j, k)\big)$ are obtained in similar fashion. In our initial co-registration problem we consider a composition of all the six transform functions: $\mathcal{T}_{t_x,t_y,t_z,\alpha,\beta,\gamma} = \mathbf{T} \circ \mathbf{R}$, where $\mathbf{T} = \mathbf{T}_{t_x,t_y,t_z}$ and $R = R_{\alpha,\beta,\gamma}$. In matrix form

$\mathcal{T}_{t_x,t_y,t_z,\alpha,\beta,\gamma}$ can be written as:

$$\begin{bmatrix} i' \\ j' \\ k' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha & 0 \\ 0 & sin\alpha & cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\beta & 0 & sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -sin\beta & 0 & cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\gamma & -sin\gamma & 0 & 0 \\ sin\gamma & cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ k \\ 1 \end{bmatrix}$$

where (i, j, k) and (i', j', k') represent the original and the transformed coordinates, respectively. Then, the co-registration problem in Eq. (7.1.1.1) can be written as:

$$\min_{\mathcal{T},\mathcal{P}} ||\mathcal{T}(\mathbf{V}_1) - \mathbf{V}_2||_2 \tag{7.1.1.2}$$

The problem seeks to find a 6-tuple $(t_x^*, t_y^*, t_z^*, \alpha^*, \beta^*, \gamma^*)$ that minimizes Eq. (7.1.1.2):

$$\min_{(t_x,t_y,t_z,\alpha,\beta,\gamma)\in\mathcal{P}} ||\mathcal{T}(\mathbf{V}_1) - \mathbf{V}_2||_2 \tag{7.1.1.3}$$

where $\mathcal{T} = T \circ R$, $T = T_{t_x} \circ T_{t_y} \circ T_{t_z}$ and $R = R_\alpha \circ R_\beta \circ R_\gamma$.
Eq. (7.1.1.3) can be written as:

$$\begin{aligned} &\Big(\min_{(t_x,t_y,t_z,\alpha,\beta,\gamma)\in\mathcal{P}} ||\mathcal{T}(\mathbf{V}_1) - \mathbf{V}_2||_2\Big) \Longrightarrow \\ \Longrightarrow &\Big(\exists(t_x^*, t_y^*, t_z^*, \alpha^*, \beta^*, \gamma^*) \in \mathcal{P}, \quad \text{s.t:} \quad \mathcal{T}(\mathbf{V}_1) \approx \mathbf{V}_2\Big) \end{aligned} \tag{7.1.1.4}$$

The 6-tuples $(t_x^*, t_y^*, t_z^*, \alpha^*, \beta^*, \gamma^*)$ are automatically determined by the co-registration step. The three translation parameters have integer values and are determined directly without any input. On the other hand, the rotation parameters are selected from a range of given angles with integer values, i.e., $\theta \in [-10^o, 10^o]$. The co-registration step is sufficient if the difference between the original test and the transformed reference volumes is no larger than 2^o angle about each axis. It derives that there are only finite values $\in [-180^o, 180^o]$ for the rotation parameters.

Remark: For two consecutive volumes, the insignificant changes due to affine transformation, i.e., translations and rotations, are bounded. More specifically, the sizes of the shifts cannot be greater than half of the maximum of each edge of the vol-

ume. That is, $|t_x^*|, |t_y^*|$ and $|t_z^*| \leq \frac{1}{2}\max\{N_1, N_2, N_3\}$, where N_1, N_2 and N_3 are number of rows, columns and the number of slices in a volume. Therefore the sets of the three translation parameters are finite. Similarly, the sets of the rotation angles as integers from $[-180^o, 180^o]$ are finite. This means that the minimum in Eq. (7.1.1.4) is always reached.

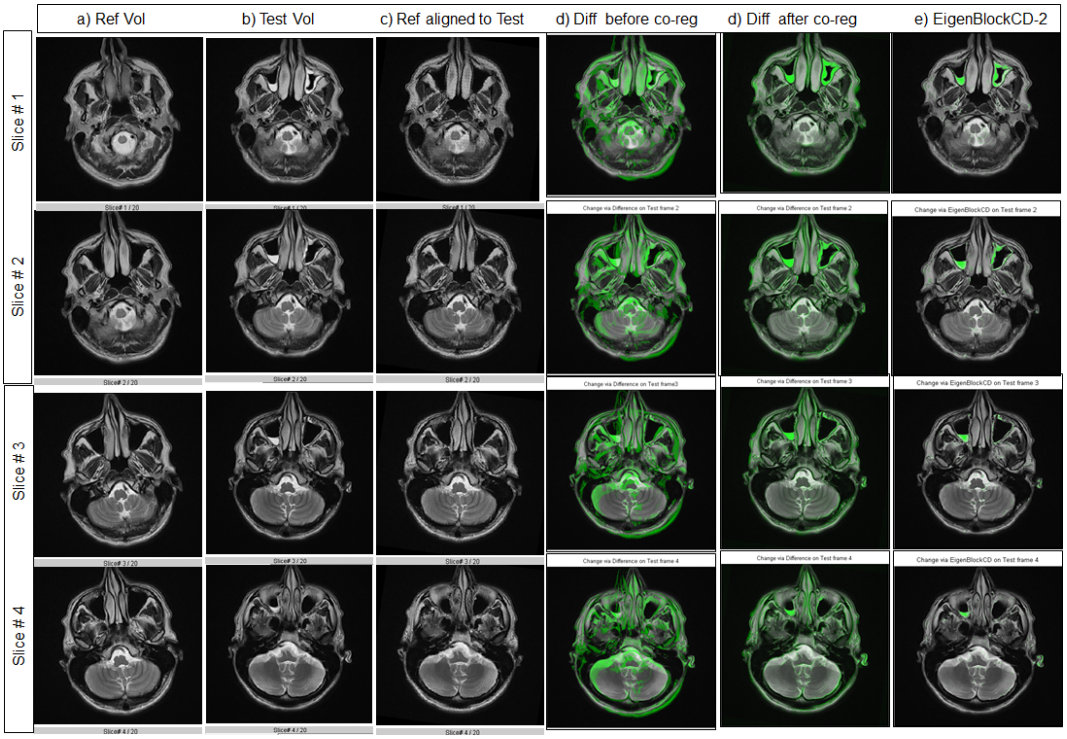

Figure 7.1.5: Rows 1 - 4 represent real T2-weighted MR slices of a brain of the reference volume taken in 2011 and its follow-up in 2013. Columns a) and b) Reference volume and Test volume, c) Reference aligned to test after six affine transformations, Columns d) and e) change volume via differencing before and after the co-registration step respectively done by our algorithm, Column f) change volume via the extension of EigenBlockCD-2 for 3D volumes.

We tested the EigenBlockCD-2 for 3D volumes algorithm with real MR volumes and the results of the initial co-registration step are displayed in Figs. 7.1.5, 7.1.6, 7.1.7, 7.1.8 and 7.1.9 Columns a), b) and c). Columns a) represent real T2-weighted MR slices of a brain in 2011 named as the reference volume, Columns b) show corresponding follow-ups slices of the T2-weighted MR volume in 2013 named as test volume and Columns c) show the reference volumes aligned to test volumes after six affine transformations, three shifts and three rotations. The six parameters in this case are: $t_x^* = 9, t_y^* = -3, t_z^* = -5, \alpha^* = -9^o, \beta^* = 2^o$ and $\gamma^* = -3^o$.

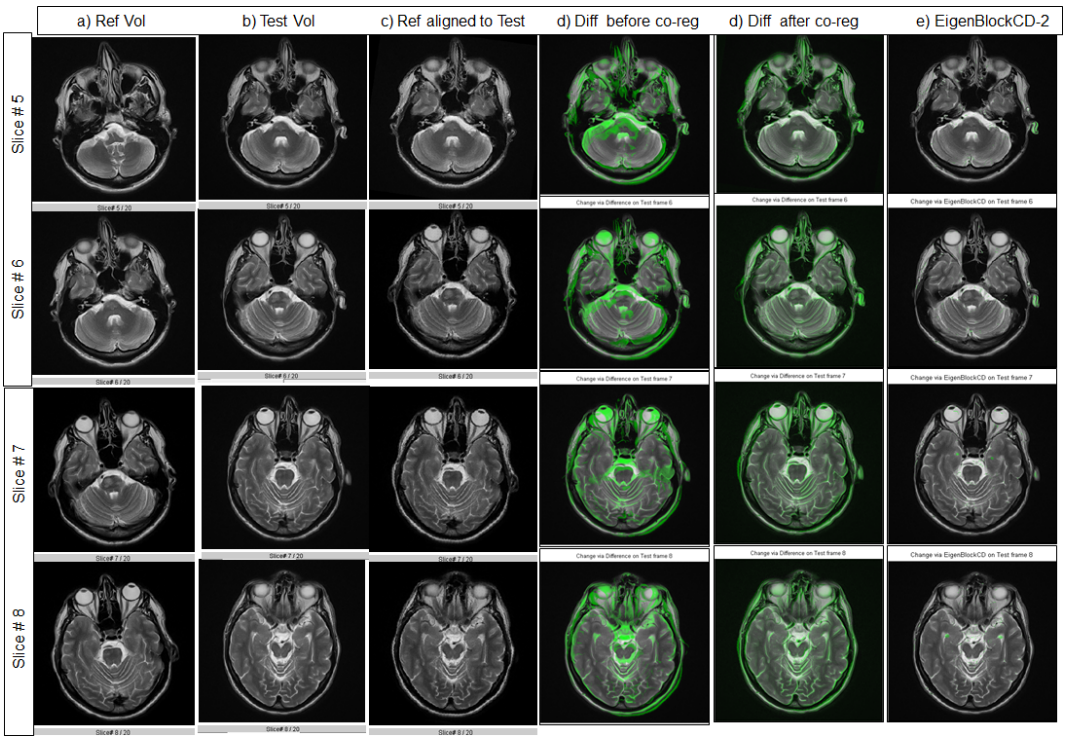

Figure 7.1.6: Rows 1 - 4 represent real T2-weighted MR slices of a brain of the reference volume taken in 2011 and its follow-up in 2013. Columns a) and b) Reference volume and Test volume, c) Reference aligned to test after six affine transformations, Columns d) and e) change volume via differencing before and after the co-registration step respectively done by our algorithm, Column f) change volume via the extension of EigenBlockCD-2 for 3D volumes.

7.1.2 Detecting Clinical Changes

In this step, the problem becomes a 2D optimization problem. As in Chapter 6, the dictionary $\mathbf{\Phi}$ is formed by stacking training blocks as column of $\mathbf{\Phi}$:

$$\mathbf{\Phi} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \tag{7.1.2.1}$$

In case of using the AEDL algorithm, the background of the block of interest is modeled by solving the minimization problem (P_1):

$$\text{Problem } P_1: \quad \begin{aligned} \boldsymbol{\gamma}^* &= \arg\min_{\gamma} ||\boldsymbol{\gamma}||_1 \\ \text{subject to:} \quad \mathbf{y} &= \Phi\boldsymbol{\gamma} + \mathbf{n} \end{aligned} \tag{7.1.2.2}$$

The residual errors $\mathbf{r}$ of the approximation algorithm given by Eq. (7.1.2.2) is computed as:

$$\mathbf{r} = |\mathbf{y} - \mathbf{y}^*| \tag{7.1.2.3}$$

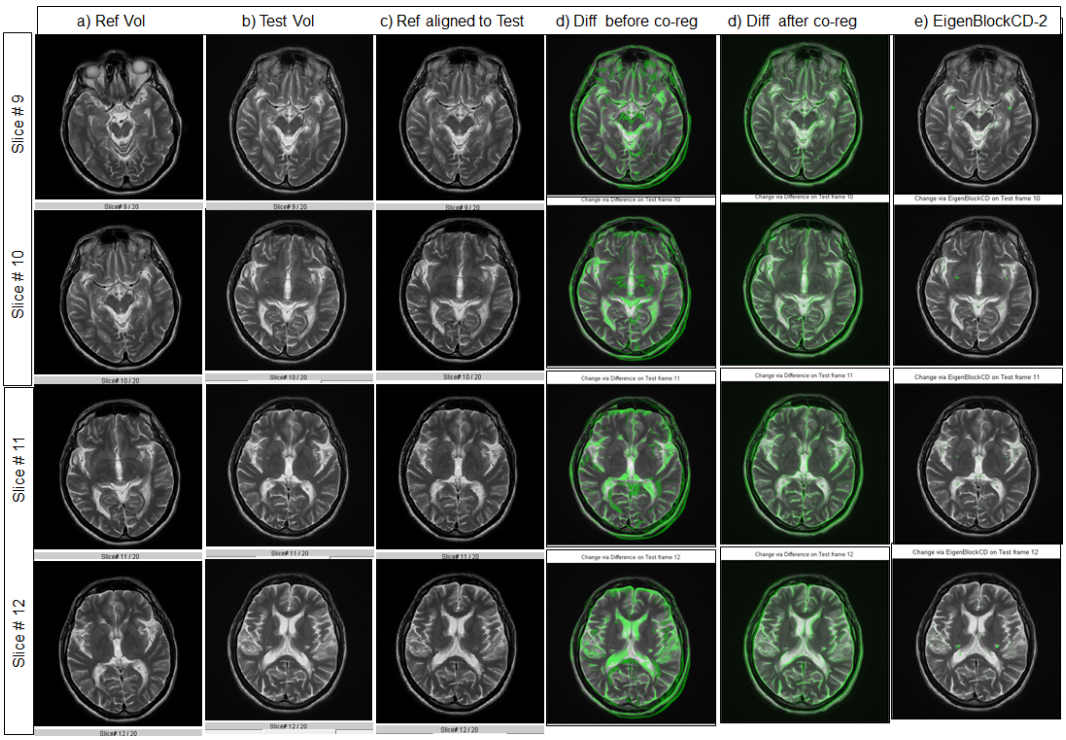

Figure 7.1.7: Rows 1 - 4 represent real T2-weighted MR slices of a brain of the reference volume taken in 2011 and its follow-up in 2013. Columns a) and b) Reference volume and Test volume, c) Reference aligned to test after six affine transformations, Columns d) and e) change volume via differencing before and after the co-registration step respectively done by our algorithm, Column f) change volume via the extension of EigenBlockCD-2 for 3D volumes.

The EigenBlockCD algorithm computes the background $\mathbf{y}^*$ of $\mathbf{y}$ by finding the best approximation to $\mathbf{y}$ in the reference volume, that is, a vector $\mathbf{x}_l$ from dictionary $\mathbf{\Phi}$ that minimizes the residual error.

$$\mathbf{y}^* = \mathbf{x}_l \quad \text{such that:} \quad \mathbf{e}_l = ||\widehat{\mathbf{y}} - \widehat{\mathbf{x}}_l||_2 = \min_{\widehat{x}_k \in \widehat{\mathbf{\Phi}}} ||\widehat{\mathbf{y}} - \widehat{\mathbf{x}}_k||_2 \tag{7.1.2.4}$$

where $\mathbf{x}_l \in \mathbf{\Phi}$, $\widehat{\mathbf{x}}_l \in \widehat{\mathbf{\Phi}}$ $\widehat{\mathbf{\Phi}}$ is the projected dictionary $\mathbf{\Phi}$ onto the eigen-subspace. The residual error block with changes is computed as:

$$\mathbf{r} = |\mathbf{y} - \mathbf{y}^*| \tag{7.1.2.5}$$

The foreground block $\mathbf{F}$ containing only the significant changes is computed as:

$$\mathbf{F} = \frac{1}{2m} \left(\sum_{a=k-\delta}^{k+\delta} \sum_{c=j-\delta}^{j+\delta} \sum_{h=i-\delta}^{i+\delta} \left(\mathbf{r}_{a,c,h}^{[1]} + \mathbf{r}_{a,c,h}^{[2]} \right) \right) \tag{7.1.2.6}$$

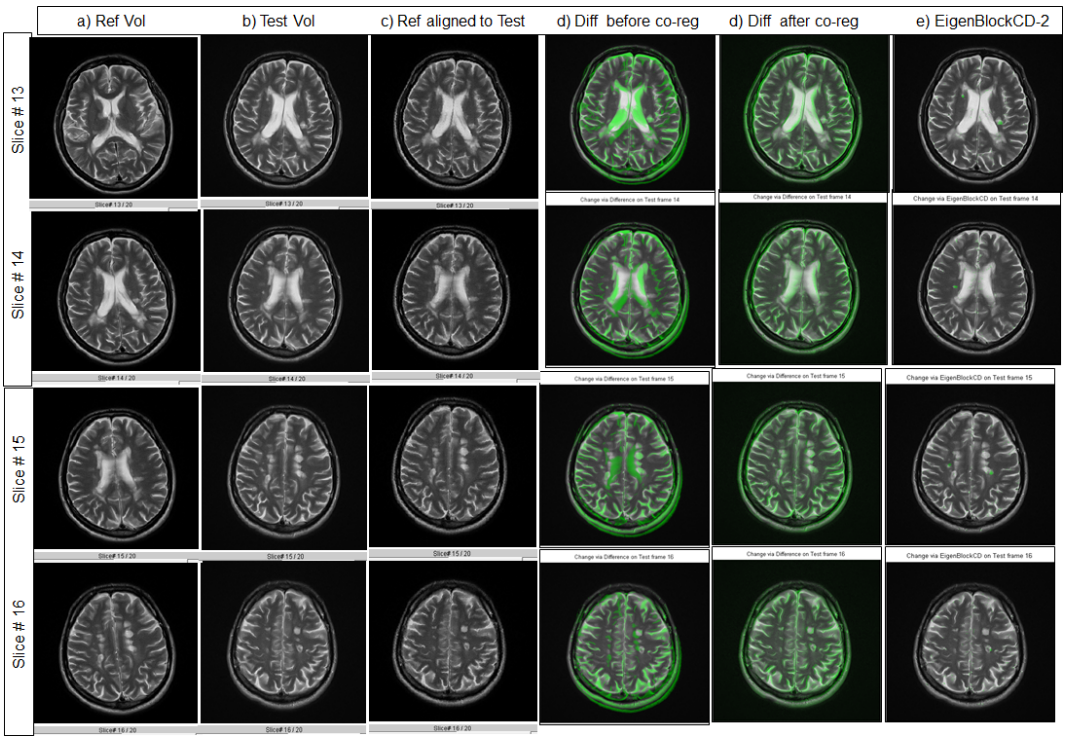

Figure 7.1.8: Rows 1 - 4 represent real T2-weighted MR slices of a brain of the reference volume taken in 2011 and its follow-up in 2013. Columns a) and b) Reference volume and Test volume, c) Reference aligned to test after six affine transformations, Columns d) and e) change volume via differencing before and after the co-registration step respectively done by our algorithm, Column f) change volume via the extension of EigenBlockCD-2 for 3D volumes.

where, $\mathbf{r}^{[1]}_{a,c,h}$ is the block $\mathbf{r}$ centered at voxel (a, c, h) when the background of the block of interest in the test image is learned from the training blocks in the reference image, and similarly, $\mathbf{r}^{[2]}_{a,c,h}$ is the block $\mathbf{r}$ centered at voxel (a, c, h) when the background of the block of interest in the reference image is learned from the training blocks in the test image. The change volume between the baseline and its follow-up is:

$$\mathbf{V}_{cd} = \frac{1}{2m}\left(\sum_{k=\delta}^{N_3-\delta}\sum_{j=\delta}^{N_2-\delta}\sum_{i=\delta}^{N_1-\delta}\mathbf{r}^{[1]}(i,j,k) + \mathbf{r}^{[2]}(i,j,k)\right) \tag{7.1.2.7}$$

The detection step results of our simulations with real MR volumes are displayed in Figs. 7.1.5, 7.1.6, 7.1.7, 7.1.8 and 7.1.9 Columns d) to f). We compared the change volume obtained by the extension of the EigenBlockCD-2 algorithm for 3D volumes with one obtained by simple differencing.

First, simple differencing is performed between the original reference and test volumes before the co-registration step. These results for 20 slices are shown in Figs. 7.1.5,

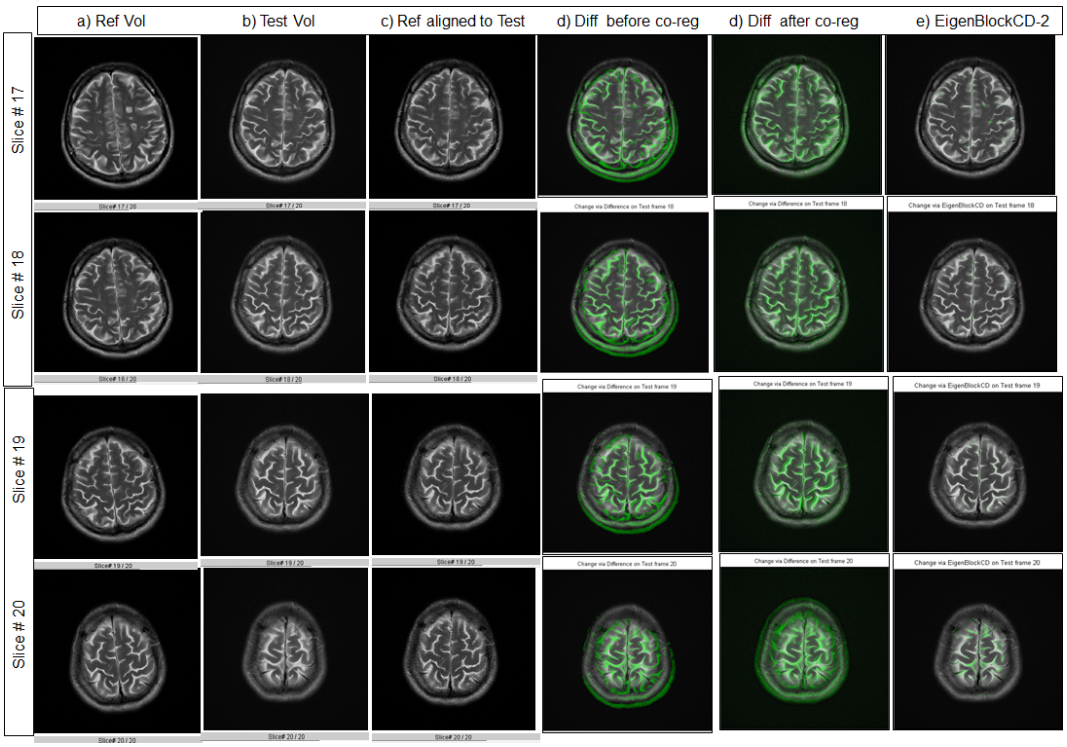

Figure 7.1.9: Rows 1 - 4 represent real T2-weighted MR slices of a brain of the reference volume taken in 2011 and its follow-up in 2013. Columns a) and b) Reference volume and Test volume, c) Reference aligned to test after six affine transformations, Columns d) and e) change volume via differencing before and after the co-registration step respectively done by our algorithm, Column f) change volume via the extension of EigenBlockCD-2 for 3D volumes.

7.1.6, 7.1.7, 7.1.8 and 7.1.9 Columns d).

Second, simple differencing is performed after the co-registration step by the extension of the EigenBlockCD-2 co-registration step algorithm, i.e., after the initial alignment of the two volumes. These results are shown in Figs. 7.1.5, 7.1.6, 7.1.7, 7.1.8 and 7.1.9 Columns f).

The results of the extension of the EigenBlockCD-2 algorithm for 3D volumes are shown in Figs. 7.1.5, 7.1.6, 7.1.7, 7.1.8 and 7.1.9 Columns f). As confirmed by the radiologist, these results show that the extended version of our EigenBlockCD-2 algorithm for 3D volumes detect clinical and significant changes while rejecting changes due to patient position or other acquisition related artifacts which are clearly visible by the simple differencing method as shown in Columns d) and f).

7.2 Summary

In this section we presented the solution to the change detection problem for 3D volumetric data. The EigenBlockCD-2 is extended to process 3D MR images as obtained from MRI machine software. Interpolation is used to generate extra slices between existing slices to include details and help co-registration process. We describe in details both the co-registration and the change detection steps.

The co-registration step is formulated as a minimization problem where the cost function is defined as the L_2 norm of the difference between one of the original volumes and the other volume transformed by an affine transformation. Simulations with real 3D volumetric data show that co-registration step performs an excellent alignment of the reference and the test MR volumes.

The change detection problem is defined as a 2D optimization problem. The simulations with real 3D MR images demonstrate that the EgienBlockCD-2 algorithm detects the significant changes between consecutive MR scans while rejecting changes due to patient position and noise.

Chapter 8

Subspace Learning - Structure Principle Components Analysis

8.1 Motivation

As discussed in Chapters 3, 4, 5 and 7, the dictionaries used in the AEDL, the EigenBlockCD and the EigenBlockCD-2 algorithms are formed by stacking training blocks from the inquiry block as shown in Figs. 8.1.2 and 8.1.1. For our application these dictionaries are often very large. For example, Fig 8.1.1 shows a dictionary Φ for a block of interest in 2D of size 361×1369. Fig 8.1.2 illustrates the construction of the dictionary Φ for a block of interest in 3D of size of which is 6859×50653.

Dictionary atoms are highly correlated vectors in 361 dimensional space for the 2D case and 6859 dimensional space for the 3D case as spatially they are highly overlapped. We use principal component analysis (PCA) to find a more effective orthonormal basis to re-express the data set instead of the standard coordinate basis. Such a basis is a linear combination of the original basis and best represents the data set. In our change detection algorithms we use PCA for dimensionality reduction, feature extraction and recognition, background modeling, computational efficiency, sparse representation of the data (AEDL) and de-correlating highly correlated data.

Although PCA achieves dimensionality reduction and feature extraction, its results can be highly influenced by outliers [79]. To visualize this, let us consider a toy example.

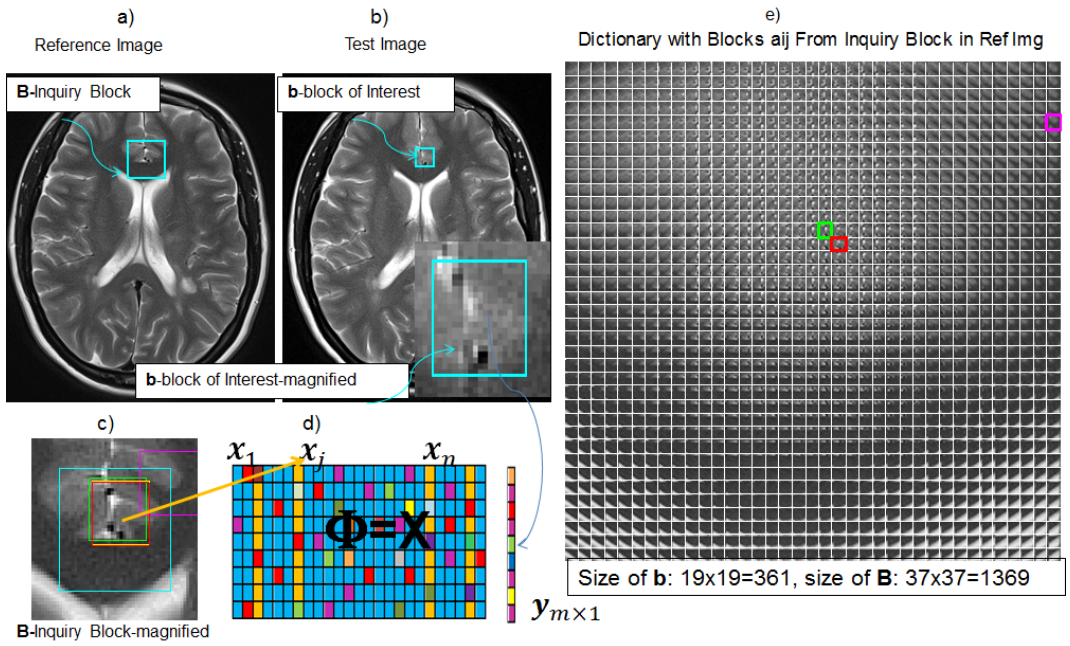

Figure 8.1.1: a) The reference image is a 2D real T2-weighted MR image, b) The test image is its follow up scan containing the magnified block of interest of size 19×19, c) The enlarged corresponding inquiry block $\mathbf{B}$ of size 37×37, d) The dictionary of size 361×1369 formed by stacking training blocks from inquiry blocks, e) The dictionary with 1369 training blocks of size 19×19.

Let $\mathbf{X}$ be a 2 by 10 data matrix, i.e., 10 observations in 2-dimensional space:

$$\mathbf{X} = \begin{pmatrix} -2 & 0.5 & 1 & 1.5 & 2 & 4 & 5 & 6 & 7 & 7.5 \\ -8 & 3 & 9 & 5.5 & 7 & 15 & 25 & 30 & 33 & 40 \end{pmatrix}$$

The mean and the standard deviation of $\mathbf{X}$ are:

$$\mu_x = \begin{pmatrix} 3.25 \\ 15.95 \end{pmatrix}, \sigma_x = \begin{pmatrix} 3.13 \\ 15.38 \end{pmatrix}$$

The data matrix $\mathbf{X}$ is first centered by subtracting the mean and scaling by standard deviation. The centered and scaled data matrix is:

$$\mathbf{X} = \begin{pmatrix} -1.678 & -0.879 & -0.719 & -0.559 & -0.399 & 0.240 & 0.559 & 0.879 & 1.198 & 1.358 \\ -1.558 & -0.842 & -0.452 & -0.680 & -0.582 & -0.062 & 0.589 & 0.914 & 1.109 & 1.564 \end{pmatrix}$$

Fig. 8.1.3 shows a strong positive correlation among the data points. This means that the data can be expressed without much loss in one dimension. The matrix $\mathbf{U}$ of

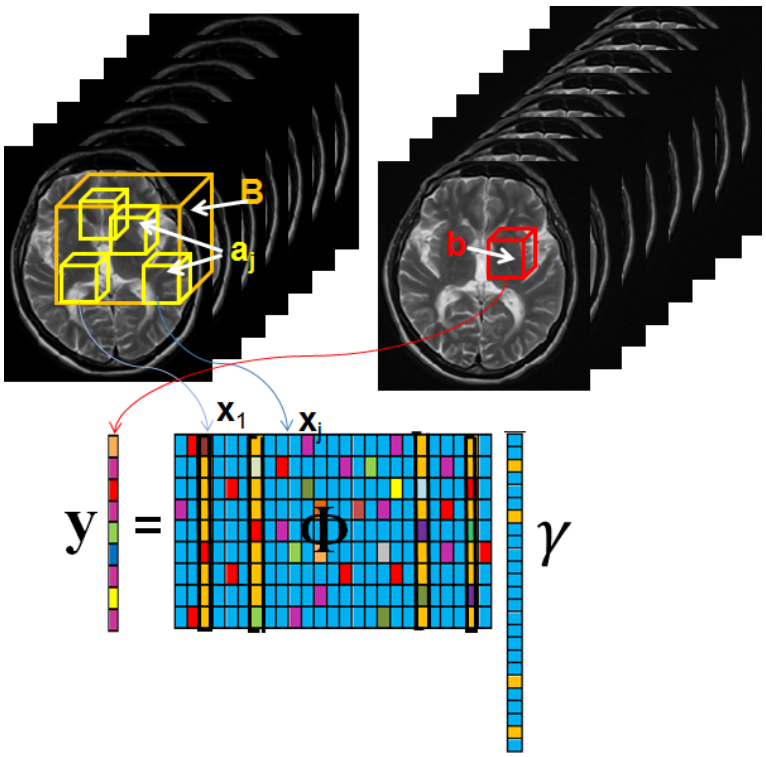

Figure 8.1.2: An example of an 3-dimensional volume of the brain. A block of interest $\mathbf{b}$ from the test volume is stacked as a column vector $\mathbf{y}$. Overlapping training blocks $\mathbf{a}_j$ are extracted from inquiry block $\mathbf{B}$ in the reference volume and are stacked as column vectors $\mathbf{x}_j$ of the dictionary $\mathbf{\Phi}$.

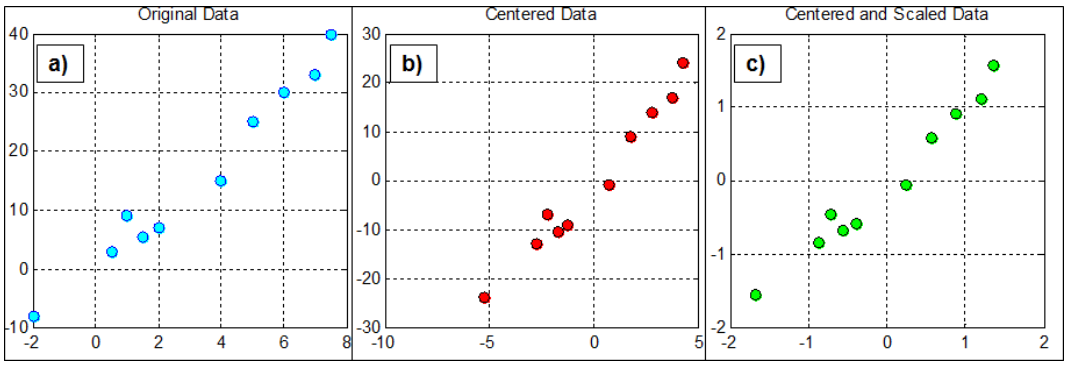

Figure 8.1.3: a) original data, b) centered data, c) centered and scaled data

eigenvectors and its corresponding matrix Λ of the eigenvalues of the data matrix $\mathbf{X}$ are:

$$\mathbf{U} = \begin{pmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{pmatrix} \text{ and } \Lambda = \begin{pmatrix} 1.9845 & 0 \\ 0 & 0.0155 \end{pmatrix}$$

The first and the second columns of $\mathbf{U}$ are the first and the second principal components. Fig. 8.1.4 a) shows that all the data points either lie on the first principal component (PC1) or are very close to it. The angle that the PC1 forms with the first

axis is: $\theta = \arctan \frac{0.7071}{0.7071} = 45^o$. We project the data onto PC1 and compute the L_2

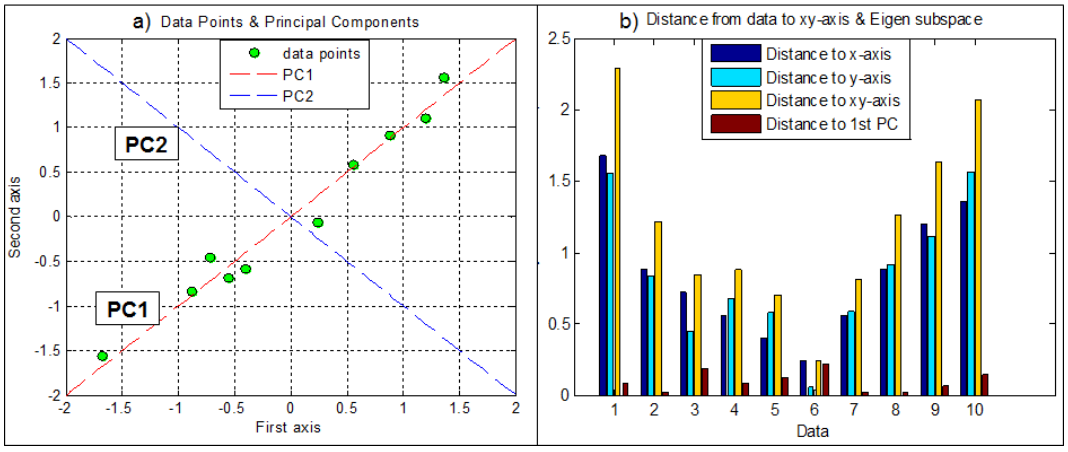

Figure 8.1.4: a) Centered and scaled data with the first and the second principal components. b) The L_2 error of the projections of each data point to the x-axis, the y-axis, the xy-axis and the first principal component PC1.

errors of the projections between each data point and the x-axis, the y-axis, the xy-axis and the PC1. Fig. 8.1.4 b) shows that there is not much information lost among the data if these data are examined in one dimensional space represented by the first principal component instead of examining them in the two dimensional original space.

Let us now add an outlier to the data, i.e., a point not consistent with the rest of the data as shown in Fig. 8.1.5. The matrix of eigenvectors $\mathbf{U}_{new}$ and its corresponding

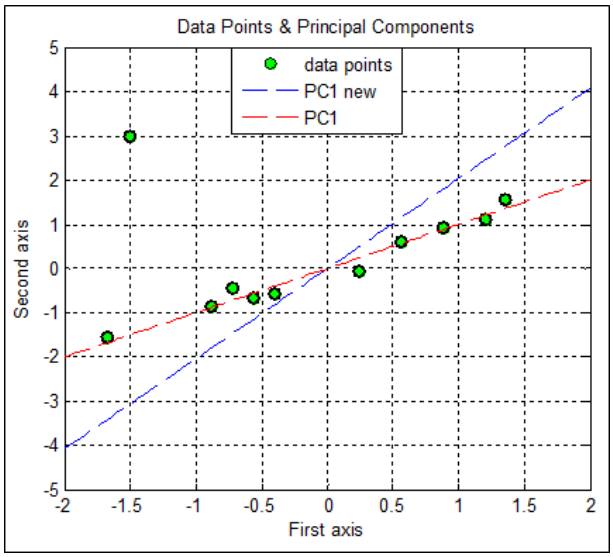

Figure 8.1.5: Centered and scaled data in green. PC1 is the original first principal component and PC1 new is the new first principal component in the presence of outliers.

matrix of the eigenvalues Λ_{new} for this perturbed data set are:

$$\mathbf{U}_{new} = \begin{pmatrix} -0.4404 & -0.8978 \\ -0.8978 & 0.4404 \end{pmatrix} \quad \Lambda_{new} = \begin{pmatrix} 2.2377 & 0 \\ 0 & 1.0123 \end{pmatrix}$$

The angle that the first principal component PC1_{new} forms with the first axis is: $\theta = \arctan \frac{-0.8978}{-0.4404} = 63.8687^o$. Just one outlier has rotated the old principal component basis by 19^o. But the trend of the data is still along the first original principal component as Fig. 8.1.5 shows, that is, the data could be expressed more precisely by the original PC1 then by the PC1_{new}. Our goal is to make these outliers less "influential" from the rest of the data. The most commonly used approaches to handle the outliers tend to robustify the covariance matrix. This is done by replacing the original covariance matrix with a robust estimator of the covariance matrix [79, 93]. An important step of these approaches is finding weights that could make use of some information that is application specific. A weighted PCA would then use a robust covariance matrix that gives higher weights to data that are considered to be more important.

In our solution to change detection problem, from all the dictionary training blocks, i.e., 1369 blocks in the 2D example shown in Fig. 8.1.1), only a subset of them are very close to the block of interest. We want to discover the subset containing the most "influential" blocks from all the dictionary training blocks. This can be done by giving more weights to the atoms which have similar structure to the block of interest. The question is how to define and compute the weights for the dictionary atoms such that it would increase the efficiency of our algorithms to approximate the background between the test and the reference blocks in a subspace created by eigenvectors of the weighted covariance matrix $\mathbf{\Omega}_w$, compared to the performance of the same algorithms in the subspace created by the eigenvectors of the original covariance matrix $\mathbf{\Omega} = \mathbf{X}\mathbf{X}^T$.

Let us first name and list different PCA that we will discuss in Chapter 8:

- PCA or the L_2-PCA - Represent the standard principal component analysis which uses the Frobenious (Euclidean) norm.
- The $L_{2,1}$-PCA or the R_1-PCA [81] - Represent the $L_{2,1}$ principal component analysis which uses the $L_{2,1}$ norm.

- The S_2-PCA -Represent the structure principal component analysis that we propose in the thesis and that uses weights induced from Structure Similarity Index (SSIM) and the Frobenious (Euclidean) norm. Sometimes we refer to S_2-PCA as just S-PCA or L_2-S-PCA

- $S_{2,1}$-PCA or $L_{2,1}$-S-PCA -The $L_{2,1}$ structure principal component analysis that we propose in the thesis that uses weights induced from Structure Similarity and the $L_{2,1}$ norm.

The rest of this chapter is organized as follows: In Sections 8.2.1 and 8.3.1 we review the principal component analysis using two different metrics, the L_2-PCA and the $L_{2,1}$-PCA. In Sections 8.2.2 and 8.3.2 we introduce our structure principal component analysis, the S_2-PCA and the $S_{2,1}$-PCA, followed by theoretical analysis and numerical implementation of the solution to the structure principal component analysis problem and their applications to the change detection problem. In Section 8.4 we present numerical comparisons among different PCA analysis.

8.2 S_2-PCA With SSIM Weights

PCA is the oldest and most well known multivariate analysis technique [79]. PCA takes high-dimensional data and uses the dependencies between the variables to represent them in a more tractable, lower-dimensional form without losing too much information. It is also known as the Karhunen- Loeve transformation, the Hotelling transformation, the method of empirical orthogonal functions and singular value decomposition.

The main purpose of principal component analysis is to reduce the dimensionality of a data set described by a large number of interrelated variables. The dimensionality reduction should retain most of the variation present in the data. The reduction is achieved by transforming the original set of variables to a new set of variables, called principal components. Principal components are uncorrelated ordered set of variables where the first few retain most of the variation present in all the observations of the original variables.

8.2.1 The Existence and Uniqueness of L_2 - PCA

Computation of the principal components reduces to the solution of an eigenvalue-eigenvector problem for a positive-semidefinite symmetric matrix [79]. Classical PCA constructs the optimal rank k subspace approximation to training data in the least square sense.

Let $\mathbf{X}_{m\times n} = [\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_j, ...\mathbf{x}_n]$ be the data matrix, where $\mathbf{x}_j \in \mathbb{R}^{m\times 1}$ are observations in m dimensional subspace, for $j = 1, ..., n$. PCA seeks to find eigenvectors of the covariance matrix $\mathbf{\Omega}$ which is computed as:

$$\mathbf{\Omega} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^T = \frac{1}{n-1}\sum_{j=1}^{n}\mathbf{x}_j\mathbf{x}_j^T \tag{8.2.1.1}$$

The element (i, j) of the covariance matrix represents the covariance between the i^{th} and j^{th} elements of $\mathbf{X}$ when $i \neq j$ and the variance of the j^{th} element of $\mathbf{X}$ when $i = j$. Hence, the covariance matrix $\mathbf{\Omega}$ is symmetric and positive definite.

PCA finds a set of orthonormal basis vectors $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k]$ with $\mathbf{u}_i \in \mathbf{R}^{m\times 1}$, by solving a constraint optimization problem: 1) minimizing the error between the original data $\mathbf{X}$ and the reconstructed data $\widehat{\mathbf{X}}$ from their projections into the eigenvectors and 2) maximizing the variance among the projected data.

Method 1 - Minimizing the Error Function

The PCA constraint minimization problem is:

$$\mathbf{U}^* = \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\arg\min}||\mathbf{X} - \widehat{\mathbf{X}}||_p \tag{8.2.1.2}$$

where $|\mathbf{X}-\widehat{\mathbf{X}}|$ represents the total approximation error which will be explained in more details later. The standard PCA uses the L_2-norm, the Frobenious norm, defined as:

$$||\mathbf{X}||_F = \Big(\sum_{j=1}^{n}\sum_{i=1}^{m}x_{ij}^2\Big)^{\frac{1}{2}} \tag{8.2.1.3}$$

Then Eq. (8.2.1.2) can be written as:

$$\mathbf{U}^* = \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\arg\min}||\mathbf{X} - \widehat{\mathbf{X}}||_2 \tag{8.2.1.4}$$

More specifically, let $\mathbf{z}_j = \mathbf{U}^T\mathbf{x}_j$ be the projection of $\mathbf{x}_j$ onto $\mathbf{U}$ as illustrated in Fig. 8.2.1, $\hat{\mathbf{x}}_j = \mathbf{U}\mathbf{U}^T\mathbf{x}_j$ is the reconstructed $\mathbf{x}_j$ and $\mathbf{e}_j = ||\mathbf{x}_j - \hat{\mathbf{x}}_j||_2$ is the error of the approximation of $\mathbf{x}_j$ with its reconstructed $\hat{\mathbf{x}}_j$. Let $f_j(\mathbf{U}) = ||\mathbf{e}_j||_2^2$ be the error

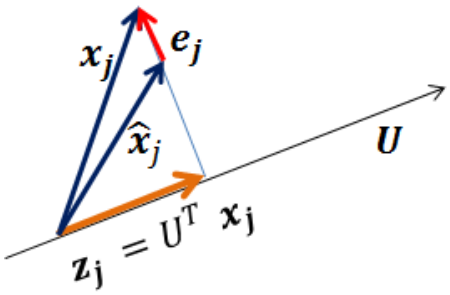

Figure 8.2.1: An illustration of the error $\mathbf{e}_j$ between $\mathbf{x}_j$ and its reconstructed vector $\widehat{\mathbf{x}}_j$ in a two-dimensional space. $\mathbf{z}_j$ is the projection of $\mathbf{x}_j$ onto subspace $\mathbf{U}$.

function of the projection vector $\mathbf{x}_j$ onto subspace $\mathbf{U}$ created by the eigenvectors of the covariance matrix $\mathbf{\Omega}$, i.e.,

$$\begin{aligned} f_j(\mathbf{U}) &= ||\mathbf{x}_j - \hat{\mathbf{x}}_j||_2^2 \\ &= ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2^2 \end{aligned}$$

The square of the L_2 norm of the total error as a function of $\mathbf{U}$ is:

$$\begin{aligned} F(\mathbf{U}) &= ||\mathbf{X} - \hat{\mathbf{X}}||_2 \\ &= \textstyle\sum_{j=1}^{n} f_j(\mathbf{U}) \\ &= \textstyle\sum_{j=1}^{n} ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2^2 \end{aligned}$$

The constraint minimization problem in Eq. (8.2.1.4) can be written as:

$$\mathbf{U}^* = \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\arg\min} ||\mathbf{X} - \mathbf{U}\mathbf{U}^T\mathbf{X}||_2^2 \tag{8.2.1.5}$$

where $\mathbf{X}$ is the $m \times n$ data matrix.

Eq. (8.2.1.5) can be written as:

$$\mathbf{U}^* = \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\arg\min} \sum_{j=1}^{n} ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2^2 \tag{8.2.1.6}$$

The global minimum of the standard PCA is provided by the SVD solution [94]. The solution is also the solution of the maximization problem that searches for a projection matrix $\mathbf{U}^*$ which maximizes the variances of $\mathbf{U}^T\mathbf{X}$ among the projected data.

$$\mathbf{U}^* = \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\arg\max}\Big(Var(\mathbf{U}^T\mathbf{X})\Big) = \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\arg\max}\Big(\mathbf{U}^T\mathbf{\Omega}\mathbf{U}\Big) \tag{8.2.1.7}$$

where $Var(\mathbf{U}^T\mathbf{X})$ is the variance of $\mathbf{U}^T\mathbf{X}$ and $\mathbf{\Omega} = \mathbf{X}\mathbf{X}^T$ is the covariance matrix.

Both methods, minimizing the total error or maximizing the variances among the projected data, require knowledge of convex optimization. For this reason we investigate the convexity of the PCA optimization problem, the convexity of the problem domain, the objective function and the convexity of the constraints [103].

1. The convexity of the PCA optimization problem domain.

Definition 8.2.1.1- Convex Sets: A set $\mathcal{S}$ is convex if and only if any linear interpolation between two points from $\mathcal{S}$ is in $\mathcal{S}$. That is:

$$\big(\mathcal{S} \text{ is convex}\big) \iff \big((\forall x_1, x_2 \in \mathcal{S} \text{ and } \lambda \in (0,1)) \implies \lambda x_1 + (1-\lambda)x_2 \in \mathcal{S}\big) \tag{8.2.1.8}$$

Fig. 8.2.2 illustrates an example of convex and non convex sets.

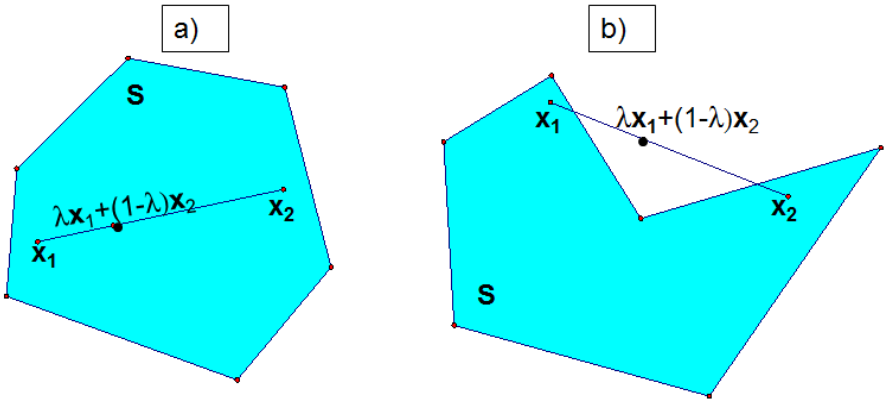

Figure 8.2.2: a) An example of a convex set and b) an example of a non convex set

Empty sets are examples of convex sets. Interesting and very useful convex sets are those formed by a given norm, i.e., the set $\{\mathbf{x} \in \mathbb{R}^{n\times 1} \mid ||\mathbf{x}|| \leq a\}$ is convex for any $a \in \mathbb{R}$. The proof is simple and derives directly from the triangle inequality property

of a norm [103]. If $\mathbf{x}_1$, $\mathbf{x}_2 \in \mathbb{R}^{n\times 1}$, $\lambda \in (0,1)$, and $a \in \mathbb{R}$ such that:

$$||\mathbf{x}_1|| \leq a \quad \text{and} \quad ||\mathbf{x}_2|| \leq a$$

then, from the triangle inequality of a norm, for any interpolated point $\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ between $\mathbf{x}_1$ and $\mathbf{x}_2$, we have:

$$||\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2|| \leq ||\lambda\mathbf{x}_1|| + ||(1-\lambda)\mathbf{x}_2|| = \lambda||\mathbf{x}_1|| + (1-\lambda)||\mathbf{x}_2||$$

It's easy to derive that the intersection of convex sets is also a convex set. If S_k is a convex sets for every k, then $\mathcal{S} = \bigcap_k S_k$ is convex. If $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{S}$ and $\lambda \in (0,1)$, then $\forall k$ $\mathbf{x}_1, \mathbf{x}_2 \in S_k$ from the fact that each S_k is convex, we have:

$$(\mathbf{x}_1, \mathbf{x}_2 \in S_k, \forall k) \implies (\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in S_k, \forall k) \implies (\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in \bigcap_k S_k = \mathcal{S}$$

The same result does not hold for the sum of convex sets.

In the PCA optimization problem we aim to find an eigenvector matrix as a more efficient basis to re-express the data set. We define the domain of our objective function $F(\mathbf{U})$ as a set of matrices which leads us to investigate the convexity of such a set.

Definition 8.2.1.2- Let $dom(F) = \mathcal{M} = \{M_{m\times r} | M \text{ is a rectangular } m \times r \text{ matrix}\}$

Property: The set $\mathcal{M}$ of all $m \times r$ matrices is convex.

Proof: Let M_1 and M_2 be two $m \times r$ matrices $\in \mathcal{M}$ and $\lambda \in (0,1)$. It is clear that any linear interpolation between two $m \times r$ matrices is a $m \times r$ matrix, that is, $\lambda M_1 + (1-\lambda)M_2 \in \mathcal{M}$ ■.

Hence, in the PCA optimization problem the domain is convex.

2. The convexity of the objective function.

Definition 8.2.1.3 - Convex Functions: A function $f : \mathbb{R}^n \to \mathbb{R}\bigcup\{+\infty\}$ is convex at $\widetilde{\mathbf{x}} \in S$, where $S \subseteq \mathbb{R}^n$ is a convex set, if and only if for any $\mathbf{x} \in S$ and any $\lambda \in (0,1)$ such that $\lambda\widetilde{\mathbf{x}} + (1-\lambda)\mathbf{x} \in S$, the following inequality holds:

$$f(\lambda\widetilde{\mathbf{x}} + (1-\lambda)\mathbf{x}) \leq \lambda f(\widetilde{\mathbf{x}}) + (1-\lambda)f(\mathbf{x}) \tag{8.2.1.9}$$

A function is convex on a set S if it is convex at every point in S. In other words, a convex function is such that a linear interpolation between any two points never is lower than the function itself as illustrated in Fig. 8.2.3 a).

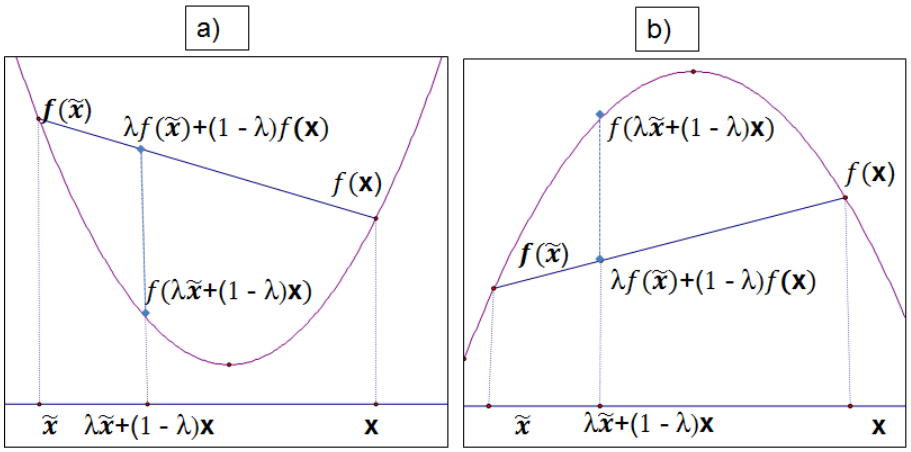

Figure 8.2.3: a) An example of a convex function and b) an example of a non-convex (concave) function

Definition 8.2.1.4 - Strictly Convex Functions: A function $f : \mathbb{R}^n \to \mathbb{R} \bigcup \{+\infty\}$ is strictly convex at $\tilde{\mathbf{x}} \in S$, where $S \subseteq \mathbb{R}^n$ is a convex set, if and only if for any $\mathbf{x} \in S$ and any $\lambda \in (0, 1)$ such that $\lambda\widetilde{\mathbf{x}} + (1 - \lambda)\mathbf{x} \in S$ the following inequality holds:

$$f(\lambda\widetilde{\mathbf{x}} + (1 - \lambda)\mathbf{x}) < \lambda f(\widetilde{\mathbf{x}}) + (1 - \lambda)f(\mathbf{x}) \tag{8.2.1.10}$$

Definition 8.2.1.5 - Concave and Strictly Concave Functions: A function $f : \mathbb{R}^n \to \mathbb{R} \bigcup \{+\infty\}$ is concave at $\widetilde{\mathbf{x}} \in S$, where $S \subseteq \mathbb{R}^n$ is a convex set, if $-f$ is convex. That is, for any $\mathbf{x} \in S$ and any $\lambda \in (0, 1)$ such that $\lambda\widetilde{\mathbf{x}} + (1 - \lambda)\mathbf{x} \in S$ we have the following inequality:

$$f(\lambda\widetilde{\mathbf{x}} + (1 - \lambda)\mathbf{x}) \geq \lambda f(\tilde{\mathbf{x}}) + (1 - \lambda)f(\mathbf{x}) \tag{8.2.1.11}$$

Similarly, a function is concave on a set S if it is concave at every point in S. This means that a concave function is such that a linear interpolation between any two points is never greater than the function itself as illustrated in Fig. 8.2.3 b). A function is strictly concave only if the strict inequality holds.

Convexity of Composite Functions: Let $S \subseteq \mathbb{R}^n$ be a convex set and $P \subseteq \mathbb{R}$. Let

$g : S \to \mathbb{R}$ be a convex function on S and $f : P \to \mathbb{R}$ be a convex non-decreasing function, then the composite $f(g)$ is convex on $\mathbf{x} \in \mathbb{R}^n$ such that $g(\mathbf{x}) \in P$. Proof can be found in [103]. Similar property holds for concavity:

Concavity of Composite Functions: [103]Let $S \subseteq \mathbb{R}^n$, $P \subseteq \mathbb{R}$, $g : S \to \mathbb{R}$ a concave function on the convex set S and $f : P \to \mathbb{R}$ a concave non-decreasing function, then the composite $f(g)$ is concave on $\mathbf{x} \in \mathbb{R}^n | g(\mathbf{x}) \in P$.

Convexity of Linear Combinations of Convex Functions: [103] Let $S \subseteq \mathbb{R}^n$, $f_k :\in \mathbb{R}^n \cup \{+\infty\}, k \in \mathcal{K}$, with $\mathcal{K}$ finite, be a collection of convex functions at $\tilde{\mathbf{x}} \in S$. Then the linear combination $f(\mathbf{x})$ of f_k with $\alpha_k \geq 0$ defined as below is convex:

$$f(\mathbf{x}) = \sum_{k \in \mathcal{K}} \alpha_k f_k(\mathbf{x})$$

Similar results hold for a collection of concave function which is a concave function. The theorem below gives the necessary and sufficient conditions for a function in C^2 to be convex [103].

Theorem 8.2.1.1: Convexity Characterization in C^2

Let f be in C^2 on an open, convex set $S \subseteq \mathbf{R}^n$.

a) f is convex on $S \iff \nabla^2 f(\mathbf{x})$ is positive semidefinite $\forall x \in S$

b) $\nabla^2 f(\mathbf{x})$ is positive definite $\forall \mathbf{x} \in S \implies$ that f is strictly convex function on S.

The proof of the Theorem 8.2.1.1 can be found in [103].

In the PCA minimization problem in Eq. (8.2.1.5), the error function $f(\mathbf{U})$, where $f(\mathbf{U}) = ||\mathbf{X} - \mathbf{U}\mathbf{U}^T\mathbf{X}||_2^2$, is a convex function of $\mathbf{U}\mathbf{U}^T$ as a quadratic function with positive leading coefficients. This means that if there is a feasible solution $\mathbf{U}\mathbf{U}^T$ to the PCA minimization problem then the solution is global and unique.

Let us investigate the convexity of the error function $f_j(\mathbf{U})$ as a function of $\mathbf{U}$, where $f_j(\mathbf{U}) = ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2^2$ is the error function corresponding to the vector $\mathbf{x}_j$. The error function f_j can be written as $f_j = h[g(\mathbf{U})]$, where $h = g^2$ and $g(\mathbf{U}) = (\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j)$.

We check the sign of $\nabla^2 f_j(\mathbf{U})$.

$$\begin{aligned}\nabla^2 f_j(\mathbf{U}) &= g'(\mathbf{U})^T \cdot \nabla^2 f_j[g(\mathbf{U})] \cdot g'(\mathbf{U}) + \nabla f_j[g(\mathbf{U})] \cdot g''(\mathbf{U}) \\ &= (-2 \cdot \mathbf{U^T x}_j)^T \cdot (2) \cdot (-2 \cdot \mathbf{U}^T \mathbf{x}_j) + 2 \cdot (\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{X})^T \cdot (-2\mathbf{x}_j) \\ &= 8 \cdot \mathbf{x}_j^T \mathbf{U}\mathbf{U}^T \mathbf{x}_j - 4 \cdot \mathbf{x}_j^T \mathbf{x}_j + 4\mathbf{x}_j^T \mathbf{U}\mathbf{U}^T \mathbf{x}_j \\ &= 12 \cdot \mathbf{x}_j^T \mathbf{U}\mathbf{U}^T \mathbf{x}_j - 4 \cdot \mathbf{x}_j^T \mathbf{x}_j \\ &= 12 \cdot ||\mathbf{U}^T \mathbf{x}_j||_2^2 - 4 \cdot ||\mathbf{x}_j||_2^2\end{aligned} \tag{8.2.1.12}$$

This means that $\nabla^2 f_j(\mathbf{U})$ changes signs, that is, $\nabla^2 f_j(\mathbf{U})$ is positive definite if the length of the projected vector is greater than $\frac{1}{\sqrt{3}}$ of the length of the original vector itself, that is $||\mathbf{U}^T \mathbf{x}_j||_2^2 > \frac{1}{3}||\mathbf{x}_j||_2^2$. Similarly, $\nabla^2 f(\mathbf{U}) = 12 \cdot \mathbf{X}^T \mathbf{U}\mathbf{U}^T \mathbf{X} - 4 \cdot \mathbf{X}^T \mathbf{X} = 12 \cdot ||\mathbf{U}^T \mathbf{X}||_2^2 - 4 \cdot ||\mathbf{X}||_2^2$, where $f(\mathbf{U}) = ||\mathbf{X} - \mathbf{U}\mathbf{U}^T \mathbf{X}||_2^2$ is the total error function, i.e., $f(\mathbf{U}) = \sum_{j=1}^{n} f_j(\mathbf{U})$. Although the objective function is neither convex nor concave, there are subsets of the function domain for which the objective function is strictly convex and also there are subsets for which the objective function is concave on them.

Before describing the conditions for the constraints let us define the convex optimization problem:

Definition 8.2.1.1: Convex Optimization Problem The standard form of convex optimization problem is:

$$\begin{aligned}\text{minimize} & f(x) \\ \text{subject to:} & g_i(x) \leq b_i, i \in \mathcal{I} \\ & h_k(x) = d_k,\ k \in \mathcal{E}\end{aligned}$$

where $\mathcal{I}$ and $\mathcal{E}$ are inequality and equality indexes respectively [103].

The domain of the objective function must be convex, the objective function f must be convex and the equality constraints must be affine. It can be easily seen that the equality constraints in the PCA optimization problem are quadratic, i.e., $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. To eliminate the constraint a new objective function, the Lagrangian, is introduced.

Theorem 8.2.1.2 - Existence of the Solutions to the PCA Problem [79]: PCA has a solution to Eq. (8.2.1.6) and can be obtained by the principal components of the

covariance matrix $\mathbf{\Omega}$.

Using the Karush Kuhn Tucker (KKT) necessary condition of the Lagrangian, it can be easily proven that columns of $\mathbf{U}$ are orthonormal eigenvectors and the Lagrangian multiplier $\mathbf{\Lambda}$ is the eigenvalue matrix of the covariance matrix $\mathbf{\Omega}$. The complete proof can be found in [79].

Until now we know from the necessary condition that there is a local minimum to the PCA minimization problem. If $\mathbf{U}^*$ is a stationary point for the objective function $f(\mathbf{U})$ as a function of $\mathbf{U}$, then $\mathbf{U}^*\mathbf{U}^{*T}$ is a local minimum of the objective function f as a function of $\mathbf{U}\mathbf{U}^T$. As we showed previously, function $f(\mathbf{U}\mathbf{U}^T)$ is strictly convex. From the fundamental theorem of global optimality, the local minimum is global and unique if the objective function is strictly convex [103]. This means that $f(\mathbf{U}^*\mathbf{U}^{*T})$ is global minimum and $\mathbf{U}^*\mathbf{U}^{*T}$ is a unique global minimum point. Since the global minimum value of f is unique, then $f(\mathbf{U}^*)$ is global minimum and the point $\mathbf{U}^*$ is global minimum point which is also unique up to an orthogonal transformation.

Moreover, if the domain of the objective function is restricted to satisfy the inequality $||\mathbf{U}^T\mathbf{x}_j||_2^2 > \frac{1}{3}||\mathbf{x}_j||_2^2$ for all atoms $\mathbf{x}_j$ then the objective function is strictly convex leading to a global and unique minimum of the objective function on the restricted domain.

Method 2 - Maximizing the Variance

The PCA maximization problem Eq.(8.2.1.7) maximizes the variances of $\mathbf{U}^T\mathbf{X}$ under the same constraints $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. The constraint optimization problem is given by:

$$\mathbf{U}^* = \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\arg\max}\Big(Var(\mathbf{U}^T\mathbf{X})\Big) = \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\arg\max}\Big(\mathbf{U}^T\mathbf{\Omega}\mathbf{U}\Big) \tag{8.2.1.13}$$

Where $Var(\mathbf{U}^T\mathbf{X})$ is the variance of $\mathbf{U}^T\mathbf{X}$ and $\mathbf{\Omega} = \mathbf{X}\mathbf{X}^T$ is the covariance matrix.

Theorem 8.2.1.3 The optimal solution to Eq.(8.2.1.13) are given by the principal components of the covariance matrix $\mathbf{\Omega}$ [79].

The solution obtained by minimizing the total error in Theorem 8.2.1.2 and the one obtained by maximizing the variance in Theorem 8.2.1.3 are the same up to an orthonormal transformation. Let $\mathbf{U}_1$ and $\mathbf{U}_2$ be two different solutions to the PCA minimization problem, where $\mathbf{U}_1$ and $\mathbf{U}_2$ are the $m \times r$ eigenvector matrices of the

covariance matrix $\mathbf{\Omega}$. From the equality constraints we have:

$$\mathbf{U}_1^T\mathbf{U}_1 = \mathbf{I}_{r\times r}$$
$$\mathbf{U}_2^T\mathbf{U}_2 = \mathbf{I}_{r\times r}$$

From $\mathbf{U}_1^T\mathbf{U}_1 = \mathbf{U}_2^T\mathbf{U}_2$, it derived that the solution to the PCA optimization problem is unique up to an orthogonal transformation.

8.2.2 The Existence and Uniqueness of S_2-PCA With SSIM Weights

In most multivariate data analysis methods, including PCA, it is usually assumed that each data point (observations) contributes equally to the process. However, this assumption is not true in every application. Joliffe in [79] classifies some observations as outliers or influential, where the latter is consisted of outliers whose effect is large on the results of multivariate analysis. Several approaches such as those in [80, 81, 83, 84, 85, 87, 88, 89, 90, 93, 104] have been proposed to reduce the effects of such observations and find a more robust solution. Using weights with either observations or variables or both have been a common method in least squares approximation [79]. The effect of differing weights for observations in a PCA problem is examined in [92].

We introduce our newly developed Structure Principles Components Analysis (S_2-PCA) in which we weigh the observations, i.e., the dictionary atoms, by Structure Similarity measure. More specifically, let $\mathbf{\Phi}$ be the dictionary created by n training blocks from the inquiry block in m dimensional space:

$$\mathbf{X}_{m\times n} = \mathbf{\Phi}_{m\times n} = [\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_j, ...\mathbf{x}_n]$$

where $\mathbf{x}_j$ is the j^{th} atom of the dictionary formed by the j^{th} training block.

$$\mathbf{x}_j = \begin{bmatrix} \mathbf{x}_{1j} \\ \mathbf{x}_{2j} \\ \vdots \\ \mathbf{x}_{mj} \end{bmatrix}_{m\times 1} \qquad j = 1, 2, ...n$$

Columns of matrix $\mathbf{X}$ represent n-samples (observations) and the m-rows represent the m-dimensions (variables). We introduce an $n \times n$ diagonal weight matrix $\mathbf{W}$ for the observations, i.e., the training blocks. Elements on the main diagonal of $\mathbf{W}$ are weights w_{jj} for $j = 1, 2, ...n$.

The S_2-PCA aims to find eigenvectors of the weighted covariance matrix as the most efficient basis to project the dictionary data matrix $\mathbf{X}$. We want to give more weight to the training blocks which are similar to the block of interest and less weight to the ones that are less similar to it. Similarity between two images can be measured using the **S**tructural **SIM**ilarity index (**SSIM**) defined in [45]. It is a similarity measure of the test image being compared to the reference image.

If $\mathbf{x}_j$ is the j^{th} column vector in the dictionary representing the $\mathbf{a}_j$ training block and $\mathbf{y}$ is the column vector representing the block of interest $\mathbf{b}$, then we denote by S_j the SSIM index between the training block $\mathbf{a}_j$ and the block of interest $\mathbf{b}$.

SSIM index has three components; 1) the luminance comparison $l(\mathbf{x}_j, \mathbf{y})$, 2) the contrast comparison $c(\mathbf{x}_j, \mathbf{y})$ and 3) the structure comparison $s(\mathbf{x}_j, \mathbf{y})$, i.e.,

$$l(\mathbf{x}_j, \mathbf{y}) = \frac{2\mu_{x_j}\mu_y + c_1}{\mu_{x_j}^2 + \mu_y^2 + c_1},\ c(\mathbf{x}_j, \mathbf{y}) = \frac{2\sigma_{x_j}\sigma_y + c_2}{\sigma_{x_j}^2 + \sigma_y^2 + c_2},\ s(\mathbf{x}_j, \mathbf{y}) = \frac{\sigma_{x_j,y} + c_3}{\sigma_{x_j}\sigma_y + c_3} \qquad (8.2.2.1)$$

where μ_{x_j}, μ_y, σ_{x_j} and σ_y, are the mean and the standard deviation of $\mathbf{x}_j$ and $\mathbf{y}$ blocks respectively. Also, c_1, c_2, k_1, k_2 and L are constants: for 8 bits images, $L = 2^{(\#\ of\ bits\ per\ pixel)} - 1 = 255$, $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$, $k_1 = 0.01$, and $k_2 = 0.03$.

Then the formula for SSIM index is:

$$SSIM(\mathbf{x}_j, \mathbf{y}) = [(l(\mathbf{x}_j, \mathbf{y})]^\alpha \times [c(\mathbf{x}_j, \mathbf{y})]^\beta \times [s(\mathbf{x}_j, \mathbf{y})]^\gamma \qquad (8.2.2.2)$$

We ignore the luminance comparison since inhomogeneity changes introduced by magnetic field bias is not related to clinical changes. By choosing $c_3 = \frac{c_2}{2}$ and $\alpha = \beta = \gamma = 1$ the Eq.(8.2.2.2) for SSIM index is:

$$SSIM(\mathbf{x}_j, \mathbf{y}) = [c(\mathbf{x}_j, \mathbf{y})] \times [s(\mathbf{x}_j, \mathbf{y})]$$

$$= \left(\frac{2\sigma_{x_j}\sigma_y + c_2}{\sigma_{x_j}^2 + \sigma_y^2 + c_2}\right)\left(\frac{\sigma_{x_j,y} + c_3}{\sigma_{x_j}\sigma_y + c_3}\right)$$

$$= \left(\frac{2\sigma_{x_j}\sigma_y + c_2}{\sigma_{x_j}^2 + \sigma_y^2 + c_2}\right)\frac{2\sigma_{x_j,y} + c_2}{2\sigma_{x_j}\sigma_y + c_2}\Big)$$

$$= \left(\frac{2\sigma_{x_j,y} + c_2}{\sigma_{x_j}^2 + \sigma_y^2 + c_2}\right)$$

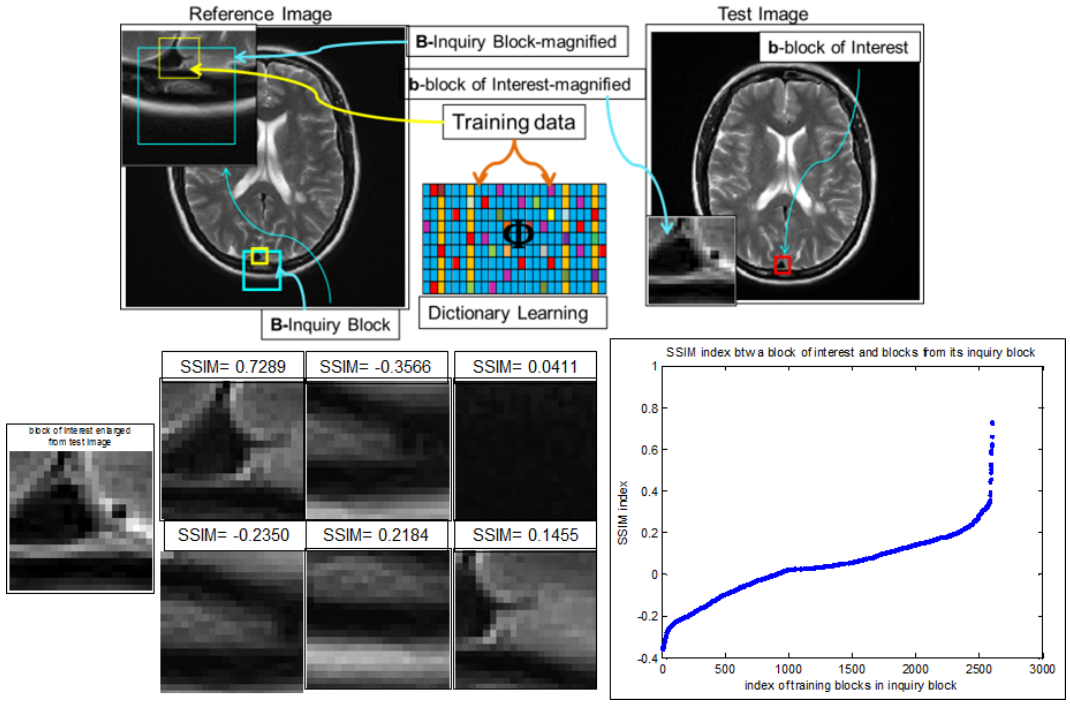

Figure 8.2.4: Top row from left to right: Reference image with the inquiry block, the dictionary and test image with the block of interest magnified. Bottom row from left to right: the block of interest from test image, training blocks with different SSIM index and the scatter plot of SSIM indexes of all the training blocks from the dictionary in ascending order.

We use the SSIM indexes as weights on the main diagonal of the weight matrix $\mathbf{W}$,

i.e., $S_j = w_{jj}$

$$S_j = SSIM(\mathbf{x}_j, \mathbf{y}) = \Big(\frac{2\sigma_{x_j,y} + c_2}{\sigma_{x_j}^2 + \sigma_y^2 + c_2} \Big) \tag{8.2.2.3}$$

The weight matrix $\mathbf{W}$ is:

$$W_{n\times n} = \begin{pmatrix} S_1 & 0 & \cdots & 0 \\ 0 & S_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_n \end{pmatrix} \tag{8.2.2.4}$$

where $S_j \in [-1, 1]$ for $j = 1, 2, ...n$.

The higher the SSIM S_j index is between the training blocks and the block of interest, the more "alike" the two blocks are, that is, the more similar structure they have as shown in Fig.8.2.4. Negative values of the SSIM index correspond to the negative values of the structure comparison index $c(\mathbf{x}_j, \mathbf{y})$. This is because the luminance and the contrasts indexes have always non negative values. Negative values of the structure comparison index between $\mathbf{x}_j$ and $\mathbf{y}$ correspond to negative values of correlation coefficients, which indicates that an increase in the intensity values of the block of interest $\mathbf{y}$ predicts a decrease in the intensity values in the training block $\mathbf{x}_j$. For this reason, the negative values of S_j's are set to zero because they are unlikely to be from the same anatomic location as the block of interest.

$$S_j = \begin{cases} S_j & \text{if } S_j \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

The $m \times m$ weighted covariance matrix $\mathbf{\Omega}_w$ is:

$$\mathbf{\Omega}_w = \mathbf{X}\mathbf{W}\mathbf{X}^T = \sum_{j=1}^{n} S_j \mathbf{x}_j \mathbf{x}_j^T \tag{8.2.2.5}$$

where $\mathbf{x}_j$ is the j^{th} atom of the weighted and centered matrix $\mathbf{X}$.

The weighted covariance matrix $\mathbf{\Omega}_w$ satisfies the following properties:

Property 8.2.2.1: $\mathbf{\Omega}_w$ is symmetric.

Proof: Need to prove that $(\mathbf{\Omega}_w)^T = \mathbf{\Omega}_w$.

$$\begin{aligned}(\mathbf{\Omega}_w)^T &= (\mathbf{X}\mathbf{W}\mathbf{X}^T)^T \\ &= (\mathbf{X}^T)^T\mathbf{W}^T\mathbf{X}^T \\ &= \mathbf{X}\mathbf{W}\mathbf{X}^T \\ &= \mathbf{\Omega}_w \quad \blacksquare\end{aligned}$$

Property 8.2.2.2: $\mathbf{\Omega}_w$ is positive definite.

Proof: Need to prove that $\mathbf{z}^T\mathbf{\Omega}_w\mathbf{z} > 0$, for vector $\mathbf{z} \neq 0$, $\mathbf{z} \in \mathbb{R}^m$.

Write $\mathbf{W} = (\sqrt{\mathbf{W}})^T(\sqrt{\mathbf{W}}) = w^T w$, then;

$$\begin{aligned}\mathbf{z}^T\mathbf{\Omega}_w\mathbf{z} &= \mathbf{z}^T(\mathbf{X}\mathbf{W}\mathbf{X}^T)\mathbf{z} \\ &= \mathbf{z}^T\mathbf{X}w^T w\mathbf{X}^T\mathbf{z} \\ &= (w\mathbf{X}^T\mathbf{z})^T(w\mathbf{X}^T\mathbf{z}) \\ &= ||w\mathbf{X}^T\mathbf{z}||^2 \geq 0\end{aligned}$$

$||w\mathbf{X}^T\mathbf{z}||^2 \geq 0$ because $||w\mathbf{X}^T\mathbf{z}||$ is the length of the $w\mathbf{X}^T\mathbf{z}$ vector, specifically the L_2 norm or the Euclidean norm. In our application, the training blocks and the block of interest are taken from two images of the same anatomical structure at two consecutive times. This means that $\mathbf{W}$ is very unlikely to be zero. Therefore, from $\mathbf{W} \neq 0$ and $\mathbf{z} \neq 0$, it derives that $||w\mathbf{X}^T\mathbf{z}||^2 \neq 0$, or that $\mathbf{z}^T\mathbf{\Omega}_w\mathbf{z} > 0$ for any non zero vector $\mathbf{z} \in \mathbb{R}^m$ $\blacksquare$.

Property 8.2.2.3: $\mathbf{\Omega}_w$ has positive and real eigenvalues if $\mathbf{W} \neq 0$.

We first prove that $\mathbf{\Omega}_w$ has real eigenvalues.

Proof: Let $\lambda = a + ib$ be a complex eigenvalue of $\mathbf{\Omega}_w$, i.e.,

$$\mathbf{\Omega}_w\mathbf{x} = \lambda\mathbf{x} \tag{8.2.2.6}$$

then $\overline{\lambda} = a - ib$ is the complex conjugate eigenvalue of $\mathbf{\Omega}_w$, i.e.,

$$\mathbf{\Omega}_w\overline{\mathbf{x}} = \overline{\lambda}\overline{\mathbf{x}} \tag{8.2.2.7}$$

We take the dot product of Eq.(8.2.2.6) with $\overline{\mathbf{x}}$

$$\overline{\mathbf{x}}^T \mathbf{\Omega}_w \mathbf{x} = \overline{\mathbf{x}}^T \lambda \mathbf{x} \tag{8.2.2.8}$$

We take the transpose of both sides in Eq.(8.2.2.7) we have:

$$\begin{aligned}(\mathbf{\Omega}_w \overline{\mathbf{x}})^T &= (\overline{\lambda}\overline{\mathbf{x}})^T \\ \overline{\mathbf{x}}^T \mathbf{\Omega}_w^T &= \overline{\mathbf{x}}^T \overline{\lambda}^T \\ \overline{\mathbf{x}}^T \mathbf{\Omega}_w &= \overline{\mathbf{x}}^T \overline{\lambda}^T\end{aligned}$$

By taking the dot product of last equation with $\mathbf{x}$ we have:

$$\overline{\mathbf{x}}^T \mathbf{\Omega}_w \mathbf{x} = \overline{\mathbf{x}}^T \overline{\lambda} \mathbf{x} \tag{8.2.2.9}$$

From Eq.(8.2.2.8) and Eq.(8.2.2.9), we get $\overline{\mathbf{x}}^T \lambda \mathbf{x} = \overline{\mathbf{x}}^T \overline{\lambda} \mathbf{x}$ or $\overline{\mathbf{x}}^T \mathbf{x} \lambda = \overline{\mathbf{x}}^T \mathbf{x} \overline{\lambda}$, with $\overline{\mathbf{x}}^T \mathbf{x}$ being the length of vector $\mathbf{x}$. It derives that the two eigenvalues are equal, that is, $a + ib = a - ib$ leading to $b = 0$. ■

Now we prove that $\mathbf{\Omega}_w$ has positive eigenvalues.

Proof: If λ and $\mathbf{x}$ are an eigenvalue and an eigenvector of $\mathbf{\Omega}_w$ respectively, then we have:

$$\mathbf{\Omega}_w \mathbf{x} = \lambda \mathbf{x}$$

By taking the dot product of both sides with $\mathbf{x}$ we have:

$$\mathbf{x}^T \mathbf{\Omega}_w \mathbf{x} = \mathbf{x}^T \lambda \mathbf{x}$$

From property 8.2.2.2, $\mathbf{\Omega}_w$ is positive definite, i.e., $\mathbf{x}^T \mathbf{\Omega}_w \mathbf{x} > 0$ for any non zero vector $\mathbf{x} \in \mathbb{R}^m$. Since $\mathbf{x}^T \lambda \mathbf{x} > 0$ and $\mathbf{x}^T \mathbf{x} = ||\mathbf{x}||^2$ then $\lambda > 0$. ■

Property 8.2.2.4: $\mathbf{\Omega}_w$ has orthogonal eigenvectors.

Proof: Need to prove that eigenvectors are orthogonal when they correspond to different λ's. Let λ_i and λ_j be two different eigenvalues of $\mathbf{\Omega}_w$. Let $\mathbf{x_i}$ and $\mathbf{x_j}$ the two

corresponding eigenvectors in $\mathbb{R}^m$. Then, λ_i and $\mathbf{x_i}$ satisfy the equation below

$$\mathbf{\Omega}_w \mathbf{x}_i = \lambda_i \mathbf{x}_i \tag{8.2.2.10}$$

Take the dot products with $\mathbf{x}_j$

$$(\mathbf{\Omega}_w \mathbf{x}_i)^T \mathbf{x}_j = (\lambda_i \mathbf{x}_i)^T \mathbf{x}_j$$
$$\mathbf{x}_i^T \mathbf{\Omega}_w \mathbf{x}_j = \mathbf{x}_i^T \lambda_1 \mathbf{x}_j$$

similarly, λ_j and $\mathbf{x_j}$ satisfy the equation:

$$\mathbf{\Omega}_w \mathbf{x}_j = \lambda_j \mathbf{x}_j \tag{8.2.2.11}$$

Take the dot products with $\mathbf{x}_i$

$$\mathbf{x}_i^T (\mathbf{\Omega}_w \mathbf{x}_j) = \mathbf{x}_i^T (\lambda_j \mathbf{x}_j)$$
$$\mathbf{x}_i^T \mathbf{\Omega}_w \mathbf{x}_j = \mathbf{x}_i^T \lambda_j \mathbf{x}_j$$

One can easily observe that $\mathbf{x}_i^T \lambda_i \mathbf{x}_j = \mathbf{x}_i^T \lambda_j \mathbf{x}_j$ and since $\lambda_i \neq \lambda_j$, then $\mathbf{x}_i^T \mathbf{x}_j = 0$, which means that $\mathbf{x}_i \perp \mathbf{x}_j$ proving $\mathbf{\Omega}_w$ has orthogonal eigenvectors. ■

Solution to structure principal component analysis, S_2-PCA

S_2- PCA finds a set of $k \leq m$ orthonormal basis vectors, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k]$, $\mathbf{u}_i \in \mathbf{R}^m$, in two ways; 1) by minimizing the error function and 2) by maximizing the variance of the projected data into the eigenspace. Let us again consider the data matrix $\mathbf{X}_{m \times n} = [\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_j, ...\mathbf{x}_n]$. The covariance matrix is:

$$\mathbf{\Omega}_w = X\mathbf{W}\mathbf{X}^T = \sum_{j=1}^{n} S_j \mathbf{x}_j \mathbf{x}_j^T$$

where $\mathbf{x}_j$ is the j^{th} atom of $\mathbf{X}$ corresponding to the j^{th} training block of the dictionary. Observations are first weighed by $\sqrt{\mathbf{W}}$ and then the weighted observations of the dictionary are centered by subtracting the mean-block $\mathbf{x'}_\mu$:

$$\mathbf{X}' = \mathbf{X}\sqrt{\mathbf{W}} \tag{8.2.2.12}$$

$$\mathbf{X'}_c = \mathbf{X'} - \mathbf{X'}_\mu \tag{8.2.2.13}$$

where

$$\mathbf{X'}_\mu = [\mathbf{x'}_\mu, \mathbf{x'}_\mu, .., \mathbf{x'}_\mu]$$

$$\mathbf{x'}_\mu = \begin{bmatrix} \frac{1}{n}\sum_{j=1}^{n} \mathbf{x'}_{1j} \\ \frac{1}{n}\sum_{j=1}^{n} \mathbf{x'}_{2j} \\ \vdots \\ \frac{1}{n}\sum_{j=1}^{n} \mathbf{x'}_{mj} \end{bmatrix}, \text{for } j = 1, 2, ...n$$

For simplicity and without any loss of generality we denote $\mathbf{X'}_c$ as $\mathbf{X}$. Then the expression for the weighted covariance matrix is given as:

$$\mathbf{\Omega}_w = \mathbf{X}\mathbf{X}^T = \sum_{j=1}^{n} \mathbf{x}_j \mathbf{x}_j^T$$

As in Section 8.2.1 the error of the approximation of $\mathbf{x}_j$ with its reconstructed $\hat{\mathbf{x}}_j$ represent the loss function and is defined by $f_j(\mathbf{U}) = ||\mathbf{x}_j - \hat{\mathbf{x}}_j||_p$ as shown in Fig. 8.2.1, where $\hat{\mathbf{x}}_j = \mathbf{U}\mathbf{U}^T\mathbf{x}_j$.

In our S_2-PCA subspace learning, we replace L_p with L_2 norm. The loss function of the projection vector $\mathbf{x}_j$ onto subspace $\mathbf{U}$ which is created by the eigenvectors of the weighted covariance matrix $\mathbf{\Omega}_w$ is defined as:

$$\begin{aligned}
f_j(\mathbf{U}) &= ||\mathbf{x}_j - \hat{\mathbf{x}}_j||_2^2 \\
&= ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2^2 \\
&= (\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j)^T(\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j) \\
&= (\mathbf{x}_j^T - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T)(\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j) \\
&= \mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j + \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j \\
&= \mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j + \mathbf{x}_j^T\mathbf{U}(\mathbf{U}^T\mathbf{U})\mathbf{U}^T\mathbf{x}_j \\
&= \mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j + \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j \\
&= \mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j \\
&= (\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j)
\end{aligned}$$

In the above calculation we replaced $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. The square of the L_2 norm of the total error as a function of $\mathbf{U}$ is then:

$$\begin{aligned} F(\mathbf{U}) &= \textstyle\sum_{j=1}^{n} f_j(\mathbf{U}) \\ &= \textstyle\sum_{j=1}^{n} (\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j) \end{aligned}$$

Method 1 - Minimizing the error

We solve the S_2-PCA problem by minimizing $F(\mathbf{U})$ under $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ constraint, i.e., by solving the constraint minimization problem

$$\begin{aligned} \min_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} F(\mathbf{U}) &= \min_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} \textstyle\sum_{j=1}^{n} ||\mathbf{x}_j - \hat{\mathbf{x}}_j||_2^2 \\ &= \min_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} \textstyle\sum_{j=1}^{n} (\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j) \end{aligned}$$

The constraint optimization problem is written as:

$$\mathbf{U}^* = \arg\min_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} \sum_{j=1}^{n} (\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j) \tag{8.2.2.14}$$

Theorem 8.2.2.1: S_2-PCA has a solution to Eq.(8.2.2.14) given by the eigenvectors of the weighted covariance matrix $\mathbf{\Omega}_w$.

Proof: We define a new objective function, called the Lagrangian, to eliminate the constraint and introduce new unknowns.

Let $\mathbb{L}(\mathbf{U}, \mathbf{\Lambda})$ be the Lagrangian given by: $\mathbb{L}(\mathbf{U}, \mathbf{\Lambda}) = F(\mathbf{U}) + \Lambda(\mathbf{U}^T\mathbf{U} - \mathbf{I}))$, where $\mathbf{\Lambda}$ the Lagrangian multiplier. The Lagrangian for this problem can be expressed as:

$$\mathbb{L}(\mathbf{U}, \mathbf{\Lambda}) = \sum_{j=1}^{n} (\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j) + T_r\mathbf{\Lambda}(\mathbf{U}^T\mathbf{U} - \mathbf{I})$$

We need to find a matrix $\mathbf{U}^*$ for which perturbations that satisfy the constraints do not change the objective function $F(\mathbf{U})$. Instead we find the stationary points of Lagrangian $\mathbb{L}$ with respect to both $\mathbf{U}$ and $\mathbf{\Lambda}$. The extrema of the unconstrained objective $\mathbb{L}$ are the extrema of the original constrained problem. The extrema to $\mathbb{L}$ occurs when:

$$\nabla F(\mathbf{U}) + \Lambda\nabla(\mathbf{U}^T\mathbf{U} - \mathbf{I})) = 0 \tag{8.2.2.15}$$

This can be expressed as:

$$\frac{\partial \mathbb{L}}{\partial \mathbf{U}} = 0 \text{ and } \frac{\partial \mathbb{L}}{\partial \mathbf{\Lambda}} = 0 \tag{8.2.2.16}$$

Under the KKT necessary conditions the partial derivatives must equal zero. Finding partial derivatives with respect to $\mathbf{U}$ in the first expression of Eq.(8.2.2.16) and considering that $\mathbf{\Omega}_w = \sum_{j=1}^{n} \mathbf{x}_j \mathbf{x}_j^T$ we have:

$$\begin{aligned} \frac{\partial \mathbb{L}}{\partial \mathbf{U}} &= 0 \\ -2\sum_{j=1}^{n} \mathbf{x}_j \mathbf{x}_j^T \mathbf{U} + 2\mathbf{U}\mathbf{\Lambda} &= 0 \\ -2\mathbf{\Omega}_w \mathbf{U} + 2\mathbf{U}\mathbf{\Lambda} &= 0 \\ \mathbf{\Omega}_w \mathbf{U} &= \mathbf{U}\mathbf{\Lambda} \\ \mathbf{U}^T \mathbf{\Omega}_w \mathbf{U} &= \mathbf{\Lambda} \end{aligned}$$

It derives from the second expression of Eq.(8.2.2.16) that the constraints are satisfied by finding the partial derivatives with respect to $\mathbf{\Lambda}$:

$$\begin{aligned} \mathbf{U}^T \mathbf{U} - \mathbf{I} &= 0 \\ \mathbf{U}^T \mathbf{U} &= \mathbf{I} \end{aligned}$$

This shows that columns of $\mathbf{U}^*$ are orthonormal eigenvectors and the Lagrangian multiplier $\mathbf{\Lambda}$ is the eigenvalue matrix of the covariance matrix $\mathbf{\Omega}_w$. ■

Theorem 8.2.2.2: S_2-PCA solution is rotationally invariant.

Proof: This derives directly from the the fact that L_2 norm of a vector is rotational invariant. If $\boldsymbol{R}$ is a rotation matrix, then we need to show that $||\boldsymbol{R}(\mathbf{x}_j) - \boldsymbol{R}(\mathbf{U}\mathbf{U}^T\mathbf{x}_j)||_2^2 = ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2^2$.

From orthogonality of $\boldsymbol{R}$ transform it derives that $\boldsymbol{R}^T\boldsymbol{R} = \mathbb{I}$. That is:

$$\begin{aligned} ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2^2 &= ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2^2 \\ &= ||\boldsymbol{R}^T\boldsymbol{R}(\mathbf{x}_j) - \boldsymbol{R}^T\boldsymbol{R}(\mathbf{U}\mathbf{U}^T\mathbf{x}_j)||_2^2 \\ &= ||\boldsymbol{R}^T\Big(\boldsymbol{R}(\mathbf{x}_j) - \boldsymbol{R}(\mathbf{U}\mathbf{U}^T\mathbf{x}_j)\Big)||_2^2 \\ &= ||\boldsymbol{R}(\mathbf{x}_j) - \boldsymbol{R}(\mathbf{U}\mathbf{U}^T\mathbf{x}_j)||_2^2 \quad \blacksquare \end{aligned}$$

Method 2 - Maximizing the Variance

The S_2-PCA problem can also be formulated as a maximization problem, where the variances of $\mathbf{U}^T\mathbf{X}$ are maximized, under the same constraints $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. The constraint optimization problem is:

$$\mathbf{U}^* = \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\arg\max}\Big(Var(\mathbf{U}^T\mathbf{X})\Big) = \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\arg\max}\Big(\mathbf{U}^T\mathbf{\Omega_w}\mathbf{U}\Big) \tag{8.2.2.17}$$

Where $Var(\mathbf{U}^T\mathbf{X})$ is the variance of $\mathbf{U}^T\mathbf{X}$ and $\mathbf{\Omega_w}$ is the weighted covariance matrix.

Theorem 8.2.2.3 The optimal solution to Eq. (8.2.2.17) is given by the eigenvectors of the weighted covariance matrix $\mathbf{\Omega_w}$.

Note: The solution obtained by maximizing the variances of $\mathbf{U}^T\mathbf{X}$ is the same as the one obtained by minimizing the total error showed in Theorem 8.2.2.1.

Proof: As in the Theorem 8.2.2.1, we define the Lagrangian as a new objective function to eliminate the constraint. We denote $G(\mathbf{U})$ as:

$$G(\mathbf{U}) = \mathbf{U}^T\mathbf{\Omega_w}\mathbf{U} = \mathbf{U}^T\mathbf{X}\mathbf{X}^T\mathbf{U}$$

Let $\mathbb{L}(\mathbf{U}, \mathbf{\Lambda})$ be the Lagrangian and $\mathbf{\Lambda}$ is the Lagrangian multiplier. The expression for $\mathbb{L}(\mathbf{U}, \mathbf{\Lambda})$ can be written as:

$$\mathbb{L}(\mathbf{U}, \mathbf{\Lambda}) = G(\mathbf{U}) - \Lambda T_r(\mathbf{U}^T\mathbf{U} - \mathbf{I}))$$

Where T_r is the trace of a matrix. By replacing $G(\mathbf{U})$ in the above equation we have:

$$\mathbb{L}(\mathbf{U}, \mathbf{\Lambda}) = \mathbf{U}^T\mathbf{\Omega_w}\mathbf{U} - T_r\mathbf{\Lambda}(\mathbf{U}^T\mathbf{U} - \mathbf{I}) \tag{8.2.2.18}$$

The extrema of the unconstrained objective $\mathbb{L}$ are the extrema of the original constrained problem. Under the necessary condition, the optimal solution of $\mathbb{L}$ satisfies:

$$\nabla G(\mathbf{U}) - \Lambda\nabla(\mathbf{U}^T\mathbf{U} - \mathbf{I})) = \mathbf{0} \tag{8.2.2.19}$$

That is, a solution $\mathbf{U}^*$ to our constraint optimization problem has zero the partial derivatives in Eq.(8.2.2.18) with respect to $\mathbf{U}$ and also with respect to Λ.

$$\frac{\partial \mathbb{L}}{\partial \mathbf{U}} = 0 \text{ and } \frac{\partial \mathbb{L}}{\partial \mathbf{\Lambda}} = 0 \tag{8.2.2.20}$$

The first equation of Eq.(8.2.2.20) gives:

$$\begin{aligned} \frac{\partial \mathbb{L}}{\partial \mathbf{U}} &= 0 \\ 2\mathbf{\Omega}_w\mathbf{U} - 2\mathbf{U}\mathbf{\Lambda} &= 0 \\ \mathbf{\Omega}_w\mathbf{U} &= \mathbf{U}\mathbf{\Lambda} \\ \mathbf{U}^T\mathbf{\Omega}_w\mathbf{U} &= \mathbf{\Lambda} \end{aligned}$$

The second equation of Eq.(8.2.2.20) ensures that the constraints are satisfied by finding the partial derivatives with respect to $\mathbf{\Lambda}$:

$$\begin{aligned} \mathbf{U}^T\mathbf{U} - \mathbf{I} &= 0 \\ \mathbf{U}^T\mathbf{U} &= \mathbf{I} \end{aligned}$$

This shows that columns of $\mathbf{U}$ are the corresponding orthonormal eigenvectors and the Lagrangian multiplier $\mathbf{\Lambda}$ is the eigenvalue matrix of the weighted covariance matrix $\mathbf{\Omega}_w$. Moreover, we showed that the solutions obtained by minimizing the total error in Theorem 8.2.2.1 and the one obtained by maximizing the variance in Theorem 8.2.2.3 are equal.

Theorem 8.2.2.4 - Global Optimal Solution to S_2-PCA Problem: The S_2-PCA problem has a global optimal solution given by the eigenvectors of the weighted covariance matrix $\mathbf{\Omega}_w$.

Proof: In the S_2-PCA minimization problem Eq.(8.2.2.14), the objective function is the error function $f(\mathbf{U}) = ||\mathbf{X} - \mathbf{U}\mathbf{U}^T\mathbf{X}||_2^2$. Let $\mathbf{U}^*$ be a stationary point for the objective function $f(\mathbf{U})$, then $\nabla f(\mathbf{U}^*) = 0$ from the necessary condition. Assuming that $\mathbf{X} \neq \mathbf{0}re$, and $\mathbf{U}^* \neq \mathbf{0}$ we have:

$$\nabla f(\mathbf{U}^*) = 2 \cdot (\mathbf{X} - \mathbf{U}^*\mathbf{U}^{*T}\mathbf{X})^T \cdot (-2\mathbf{U}^{*T}\mathbf{X}) = 0 \implies (\mathbf{X} - \mathbf{U}^*\mathbf{U}^{*T}\mathbf{X}) = 0$$

We compute $\nabla f(\mathbf{U}^*\mathbf{U}^{*T})$. Since $\mathbf{U}^{*T}\mathbf{U}^* = \mathbf{I}$ and $(\mathbf{X} - \mathbf{U}^*\mathbf{U}^{*T}\mathbf{X}) = 0$, we have:

$$\nabla f(\mathbf{U}^*\mathbf{U}^{*T}) = 2 \cdot (\mathbf{X} - \mathbf{U}^*\mathbf{U}^{*T}\mathbf{X})^T \cdot (-2\mathbf{X}) = 0,$$

Hence, we derive that $\mathbf{U}^*\mathbf{U}^{*T}$ is a stationary point of f as a function of $\mathbf{U}\mathbf{U}^T$ which means that it has a local minimum at $\mathbf{U}^*\mathbf{U}^{*T}$. The error function f as a function of $\mathbf{U}\mathbf{U}^T$ is strictly convex as a quadratic function with positive leading coefficients. Therefore, from the fundamental theorem of global optimality, the local minimum is global and unique if the objective function is strictly convex [103]. This means that $f(\mathbf{U}^*\mathbf{U}^{*T})$ is a global minimum and $\mathbf{U}^*\mathbf{U}^{*T}$ is a unique global minimum point. The global minimum of $f(\mathbf{U})$ is the same global minimum of $f(\mathbf{U}\mathbf{U}^T)$. Therefore, $f(\mathbf{U}^*)$ is a global minimum and since $\mathbf{U}^*\mathbf{U}^{*T}$ is unique, the point $\mathbf{U}^*$ is unique up to an orthogonal transformation.

8.2.3 Numerical Implementation of the S_2-PCA

Description

Structure-2 Principles Components Analysis (S_2-PCA) algorithm uses SSIM coefficients to compute a weighted data matrix on which it applies the standard PCA to obtain principal components. This specific weighting works well with our change detection algorithms as illustrated in experiments section. To ensure that we do not use negative coefficients, we set all negative coefficients to zero. The S_2-PCA algorithm can thus be described as follows:

Structure-2 Principles Components Analysis - S_2-PCA Algorithm:

INPUT: Data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ and vector $\mathbf{y} \in \mathbb{R}^{m \times 1}$

PROCEDURE:

Compute $S_j = \frac{2\sigma_{x_j,y} + c_2}{\sigma_{x_j}^2 + \sigma_y^2 + c_2}$, for $j = 1, 2, \cdots, n$

If $S_j < 0$, set S_j to zero

Compute the weight matrix $\mathbf{W}$ according to Eq. (8.2.2.4)

Compute the weighted data matrix: $\mathbf{X}_w = \mathbf{X}\sqrt{\mathbf{W}}$

Compute $\mathbf{U}$ from standard PCA on weighted data matrix

OUTPUT: matrix of eigenvectors **U**

8.3 $S_{2,1}$-PCA with SSIM Weights

Principal component analysis is widely used in many applications for dimensionality reduction and feature extraction [79]. The classical PCA is based on the L_2 norm and therefore is very sensitive to noise and outliers [79]. The L_2-norm tends to increase the outliers with large norms and treats the variables (dimensions) and the data samples (observations) the same.

Several variations of PCA have been developed over the last decade to create a robust solution as in [82, 80, 83, 81, 85, 87, 13, 19, 88]. Many of the studies have applied the L_1-norm to address some of the drawbacks of the L_2 based PCA as in [82, 83, 81, 85, 87, 88].

The L_1 norm used in L_1-PCA variations originates from LASSO [91] and is defined as:

$$||\mathbf{X}||_1 = \sum_{j=1}^{n}\sum_{i=1}^{m}|x_{ij}| \tag{8.3.0.1}$$

The L_1 norm has been widely used in sparse representations, compressed sensing, machine learning and statistics. The L_1 norm emphasizes the sparse solutions and is less sensitive to the outliers. However, it is computationally more expensive than the L_2 norm and also is not invariant to rotation.

The L_1-norm used by Baccini et al. [82] and Ke and Kanade [83] is not invariant to rotation and the performance is usually very poor when applied to K-means clustering [81]. To overcome this limitation, Ding et al. [81] proposed the R_1-PCA, a rotational invariant L_1 norm based PCA. However, the R_1-PCA iteratively performs the subspace iteration algorithm in the high-dimensional original space and is computationally expensive [88].

Kwak et al. in [85, 87] and Nie et al. in [88] proposed two different algorithms to solve the L_1 norm maximization problem. Kwak in [85, 87] proposed a greedy strategy to solve the L_1 maximization problem, whereas Nie et al. in [88] developed a non-greedy solution. Feng et al in [89] noted that although the non-greedy solution used by Nie et al. performs better in some situations, the results suggest that the L_1-PCA-

VM (variance maximization) is reasonable only with the greedy solutions. Further explorations with other norms such as $L_{2,1}$ has been investigated.

8.3.1 The Existence and Computations of R_1 - PCA Solutions

The $L_{2,1}$ -PCA or R_1 - PCA in [81] uses the $L_{2,1}$ norm defined as:

$$||\mathbf{X}||_{2,1} = \sum_{j=1}^{n} \Big(\sum_{i=1}^{m} x_{ij}^2 \Big)^{\frac{1}{2}} = ||\mathbf{X}||_{R_1} \tag{8.3.1.1}$$

The $L_{2,1}$ norm, also called as R_1 -norm, is a hybrid of L_1 and L_2 norms. One of the advantages of the $L_{2,1}$ norm is that it treats the variables (dimensions) and the data samples (observations) differently. Distances in spatial dimensions are measured in L_2 norm, while the summation over different observations (data samples) is computed using L_1 norm. Index i sums over spatial dimensions and index j sums over data samples, where $i = 1, ..., m$ and $j = 1, ..., n$.

R_1 - PCA has a global and unique solution up to an orthonormal transformation; it is rotational invariant and robust to outliers [81]. However, the solution is not straightforward because Huber's M-estimator is used [85].

Let $\mathbf{X}_{m \times n} = [\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_j, ...\mathbf{x}_n]$ be the data matrix where $\mathbf{x}_j \in \mathbb{R}^{m \times 1}$ are observations in m dimensional subspace, for $j = 1, ..., n$. The R_1-PCA [81] finds the eigenvectors of the weighted covariance matrix $\mathbf{\Omega}_w$ expressed as

$$\mathbf{\Omega}_w = \sum_{j=1}^{n} w_j \mathbf{x}_j \mathbf{x}_j^T \tag{8.3.1.2}$$

Here the weights w_j are calculated using Huber's M-estimator as:

$$w_j = \begin{cases} 1 & \text{if } ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j|| \leq c \\ \dfrac{c}{||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||}, & \text{otherwise} \end{cases}$$

The R_1-PCA finds a set of orthonormal basis vectors $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k]$, $\mathbf{u}_i \in \mathbb{R}^{m\times 1}$ by solving the following constraint minimization problem:

$$\mathbf{U}^* = \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\arg\min} ||\mathbf{X} - \widehat{\mathbf{X}}||_{R_1} \tag{8.3.1.3}$$

where $\mathbf{X} - \widehat{\mathbf{X}}$ represents the total approximation error. If $f_j(\mathbf{U})$ represents the error function of the projection vector $\mathbf{x}_j$ onto subspace $\mathbf{U}$ formed by eigenvectors of the covariance matrix $\mathbf{\Omega}_w$, then the expression for the error function is given as:

$$\begin{aligned} f_j(\mathbf{U}) &= \rho\big(||\mathbf{x}_j - \hat{\mathbf{x}}_j||_2\big) \\ &= \rho\big(||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2\big) \\ &= \rho\big(\sqrt{(||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2^2)}\big) \\ &= \rho\big(\sqrt{(\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j)^T(\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j)}\big) \\ &= \rho\big(\sqrt{(\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j)}\big) \end{aligned}$$

Where where $\rho(t)$ is the loss function computed using the Huber's M-estimator, i.e.,

$$\rho(t) = \begin{cases} t^2 & \text{if} \quad |t| \leq c \\ 2c|t| - c^2 & \text{if} \quad |t| \geq c \end{cases}$$

Then the $L_{2,1}$ norm of the total error as a function of $\mathbf{U}$ is:

$$\begin{aligned} F(\mathbf{U}) &= ||\mathbf{X} - \hat{\mathbf{X}}||_{2,1} \\ &= \textstyle\sum_{j=1}^n f_j(\mathbf{U}) \\ &= \textstyle\sum_{j=1}^n \rho\big(\sqrt{(\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j)}\big) \end{aligned}$$

The constraint minimization problem in Eq.(8.3.1.3) can be written as:

$$\mathbf{U}^* = \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\arg\min} \sum_{j=1}^{n} \rho\big(\sqrt{(\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j)}\big) \tag{8.3.1.4}$$

Theorem 8.3.1.1 - Existence of the Solution to R_1-PCA Problem: R_1-PCA has a solution to Eq.(8.2.1.6) given by the eigenvectors of the weighted covariance ma-

trix $\mathbf{\Omega}_w$.

Theorem 8.3.1.2 - Rotational Invariance of the Solution to R_1-PCA: R_1-PCA solution is rotational invariant.

Theorem 8.3.1.3 - Global Optimal Solution to R_1-PCA: The global optimal solution for R_1-PCA are given by the principal eigenvectors of the weighted covariance matrix $\mathbf{\Omega}_w$. R_1-PCA has a global optimal solution which is unique up to an orthogonal transformation.

The proofs of Theorems 8.3.1.1, 8.3.1.2 and 8.3.1.3 can be found in [81].

To compare the sensitivity of standard PCA and the R_1-PCA to outliers, we ran a set of experiments. We applied both PCA and the R_1-PCA to our toy example introduced in Section 8.1. We have already showed in Fig. 8.1.5 that although the trend of the data is still along the first old principal component, a single outlier perturbed the principal component basis by 19^o. This means that the data could be expressed more precisely by the first original principal component PC1 and then by the new first principal component $PC1_{new}$.

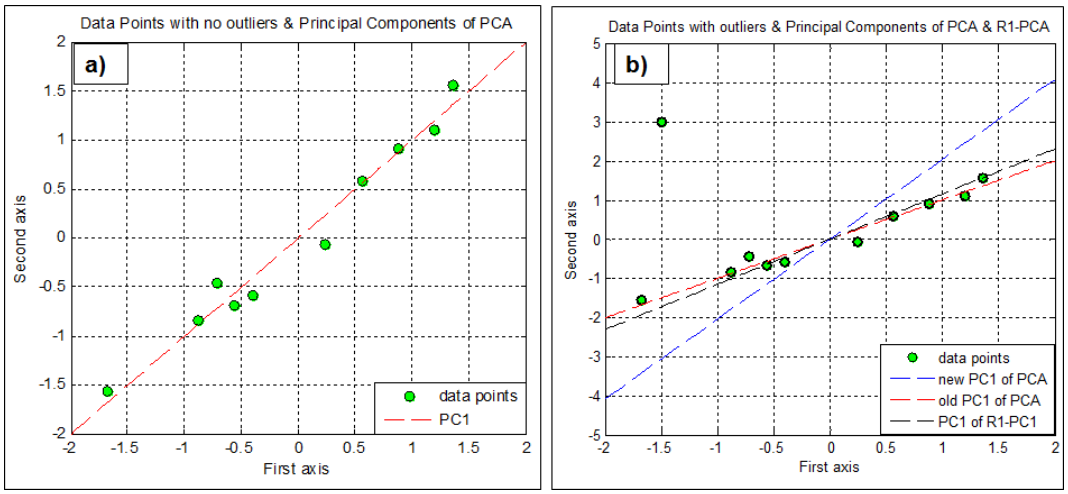

Figure 8.3.1: a) data points with no outliers and the first principal component PC1 of standard PCA. b) data points with one outlier, original PC1, new PC1 and R1-PC1.

We applied the R_1-PCA to the perturbed data and computed the principal components of the R_1-PCA. Fig. 8.3.1 a) shows that the angle which the data points with no outliers forms with the first principal component axis is 45^o. We added one outlier to the data and calculated the new first principal component of the standard PCA and the first principal component of the R_1-PCA as shown in Fig. 8.3.1 b). The angle between

the new PC1 obtained from the standard PCA and the first axis is 63.87^o whereas the the angle that the first PC1 obtained from the R_1-PCA forms with first axis is 49^o. This shows that R_1-PCA is less sensitive to outliers as Fig. 8.3.1 b) illustrates.

8.3.2 The Existence and Computation of $S_{(2,1)}$-PCA Solutions

In this section we discuss the existence and computation of the solution of the structure principal component analysis which uses the $L_{(2,1)}$ norm, named the $S_{(2,1)}$-PCA. The use of the $L_{(2,1)}$ norm allows the variables and the observations to be treated differently. It satisfies the same properties as the standard PCA because the variables are computed using the L_2 norm, i.e., the solution is global and rotationally invariant. Also, it satisfies the properties of the L_1-PCA because the summation over different observation is computed using L_1 norm which is more robust to outliers than the standard PCA.

Let Φ be the dictionary created by n training blocks from the inquiry block in m dimensional space

$$\mathbf{X}_{m\times n} = \Phi_{m\times n} = [\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_j, ...\mathbf{x}_n]$$

where $\mathbf{x}_j$ is the j^{th} atom of the dictionary formed by the j^{th} training block.

Atoms are columns of matrix $\mathbf{X}$ representing n-samples (observations) and rows represent the dimensions or the variables

$$\mathbf{x}_j = \begin{bmatrix} \mathbf{x}_{1j} \\ \mathbf{x}_{2j} \\ \vdots \\ \mathbf{x}_{mj} \end{bmatrix}, \mathbf{x}_j \in \mathbb{R}^{m\times 1}, j = 1, 2, ...n$$

In Section 8.2.2, we give more weights to training blocks which are similar to the block of interest and less weight to the ones that are less similar to it. The weights are computed using SSIM indexes. The weight matrix is an $n \times n$ diagonal matrix $\mathbf{W} = \Big(w_j\Big)$, for $j = 1, 2, ...n$.

But for the $S_{(2,1)}$-PCA, the weight matrix $\mathbf{W}$ becomes $\mathbf{W} = (S_j \cdot w_j)$, a product of two $n \times n$ diagonal matrices where S_j are the SSIM indexes as weights on the main diagonal of the weight matrix S and the weights w_j are calculated using Huber's

M-estimator. As in Section 8.2.2, the weights S_j are:

$$S_j = SSIM(\mathbf{x}_j, \mathbf{y}) = \Big(\frac{2\sigma_{x_j,y} + c_2}{\sigma_{x_j}^2 + \sigma_y^2 + c_2} \Big) \tag{8.3.2.1}$$

And the weight matrix S for the observations is:

$$S_{n\times n} = \begin{pmatrix} S_1 & 0 & \cdots & 0 \\ 0 & S_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_n \end{pmatrix} \tag{8.3.2.2}$$

where $S_j \in [-1, 1]$, $j = 1, 2, ...n$. The w_j are introduced due to the use of $L_{(2,1)}$ norm in the optimization which we will show later. The w_j are computed as in [81]:

$$w_j = \begin{cases} 1 & \text{if } ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j|| \leq c \\ \dfrac{c}{||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||}, & \text{otherwise} \end{cases} \tag{8.3.2.3}$$

where parameter c is the "cutoff" for its regularization effect of the weights in the weighted covariance matrix [81].

Larger values of the SSIM index between the training blocks and the block of interest correspond to training blocks with similar structure to the block of interest as shown in Fig.8.2.4. As in Section 8.2.2, the negative values of S_j's are set to zero,

$$S_j = \begin{cases} S_j & \text{if } S_j \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

because they are unlikely to be from the same anatomic location as the block of interest. The overall weight matrix $\mathbf{W}$ is formed by combining both weights

$$W_{n\times n} = \begin{pmatrix} S_1 \cdot w_1 & 0 & \cdots & 0 \\ 0 & S_2 \cdot w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_n \cdot w_n \end{pmatrix}$$

Hence the $m \times m$ weighted covariance matrix $\mathbf{\Omega}_w$ is:

$$\mathbf{\Omega}_w = \mathbf{X}\mathbf{W}\mathbf{X}^T = \sum_{j=1}^{n} S_j w_j \mathbf{x}_j \mathbf{x}_j^T \tag{8.3.2.4}$$

where $\mathbf{x}_j$ is the j^{th} atom of the weighted and centered matrix $\mathbf{X}$. The weighted covariance matrix $\mathbf{\Omega}_w$ satisfies all the properties proved in Section 8.2.2:

Property 8.3.2.1: $\mathbf{\Omega}_w$ is symmetric.

Property 8.3.2.2: $\mathbf{\Omega}_w$ is positive definite if $\mathbf{W} \neq \mathbf{0}$.

Property 8.3.2.3: $\mathbf{\Omega}_w$ has real and positive eigenvalues.

Property 8.3.2.4: $\mathbf{\Omega}_w$ has orthogonal eigenvectors.

The proofs of properties 8.3.2.1, 8.3.2.2, 8.3.2.3, and 8.3.2.4 are similar to the ones proved in Section 8.2.2.

Solution to structure principal component analysis, $\mathbf{S}_{(2,1)}$-PCA

$\mathrm{S}_{(2,1)}$- PCA finds a set of $k \leq m$ orthonormal basis vectors $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k]$ in two ways: 1) minimizing the error function or 2) maximizing the variance of the projected data into the eigenspace, where $\mathbf{u}_i \in \mathbf{R}^m$. Let us again consider the data matrix $\mathbf{X}_{m\times n} = [\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_j, ...\mathbf{x}_n]$. The covariance matrix is:

$$\mathbf{\Omega}_w = X\mathbf{W}\mathbf{X}^T = \sum_{j=1}^{n} S_j w_j \mathbf{x}_j \mathbf{x}_j^T$$

where $\mathbf{x}_j$ is the j^{th} atom of $\mathbf{X}$ corresponding to the j^{th} training block of the dictionary.

The covariance matrix $\mathbf{\Omega}_w$ depends explicitly on weights S_j and implicitly on weights w_j. For this reason the observations are first weighted by $\sqrt{\mathbf{S}}$, i.e., $\mathbf{X}' = \mathbf{X}\sqrt{\mathbf{S}}$ and then the weighted observations of the dictionary are centered by subtracting the

mean-block $\mathbf{x'}_\mu$, $\mathbf{X'}_c = \mathbf{X'} - \mathbf{X'}_\mu$: where $\mathbf{X'}_\mu$, and $\mathbf{X'}_\mu$ respectively are:

$$\mathbf{X'}_\mu = [\mathbf{x'}_\mu, \mathbf{x'}_\mu, .., \mathbf{x'}_\mu]$$

$$\mathbf{x'}_\mu = \begin{bmatrix} \frac{1}{n}\sum_{j=1}^{n} \mathbf{x'}_{1j} \\ \frac{1}{n}\sum_{j=1}^{n} \mathbf{x'}_{2j} \\ \vdots \\ \frac{1}{n}\sum_{j=1}^{n} \mathbf{x'}_{mj} \end{bmatrix}$$

for $j = 1, 2, \cdots, n$.

For simplicity and without any loss of generality, $\mathbf{X'_c}$ is denoted by $\mathbf{X}$. The weighted covariance matrix is:

$$\mathbf{\Omega}_w = \sum_{j=1}^{n} w_j \mathbf{x}_j \mathbf{x}_j^T$$

As in Section 8.3.1, if $f_j(\mathbf{U})$ represents the error function of the projection vector $\mathbf{x}_j$ onto subspace $\mathbf{U}$ created by the eigenvectors of the weighted covariance matrix $\mathbf{\Omega}_w$. Then the error function is:

$$\begin{aligned} f_j(\mathbf{U}) &= w_j \, ||\mathbf{x}_j - \widehat{\mathbf{x}}_j||_2 \\ &= w_j \, ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2\big) \\ &= w_j \, \sqrt{(||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2^2)} \\ &= w_j \, \sqrt{(\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j)^T(\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j)} \\ &= w_j \, \sqrt{(\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j)} \end{aligned}$$

The $L_{2,1}$ norm of the total error as a function of $\mathbf{U}$ is thus computed as:

$$\begin{aligned} F(\mathbf{U}) &= \textstyle\sum_{j=1}^{n} f_j(\mathbf{U}) \\ &= \textstyle\sum_{j=1}^{n} w_j \, \sqrt{(\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j)}e \end{aligned}$$

Method 1 - Minimizing the error

We solve the $S_{(2,1)}$-PCA problem by minimizing $F(\mathbf{U})$ under $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ constraint, i.e.,

by solving the constraint minimization problem

$$\min_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} F(\mathbf{U}) = \min_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} \sum_{j=1}^{n} w_j \sqrt{\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j} \tag{8.3.2.5}$$

The constraint minimization problem in Eq.(8.3.2.5) can be written as:

$$\mathbf{U}^* = \arg\min_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} \sum_{j=1}^{n} w_j \sqrt{\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j} \tag{8.3.2.6}$$

Theorem 8.3.2.1: $S_{(2,1)}$-PCA has a solution to Eq.(8.3.2.6) given by the principal components of the weighted covariance matrix $\mathbf{\Omega}_w$.

Proof: The Lagrangian is defined as a new objective function in which the constraint are implicit. Let $\mathbb{L}(\mathbf{U}, \mathbf{\Lambda}) = F(\mathbf{U}) + \Lambda(\mathbf{U}^T\mathbf{U} - \mathbf{I}))$ where $\mathbb{L}(\mathbf{U}, \mathbf{\Lambda})$ is the Lagrangian and $\mathbf{\Lambda}$ the Lagrangian multiplier. The Lagrangian for this problem can be expressed as:

$$\mathbb{L}(\mathbf{U}, \mathbf{\Lambda}) = \sum_{j=1}^{n} w_j \sqrt{\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j} + T_r\mathbf{\Lambda}(\mathbf{U}^T\mathbf{U} - \mathbf{I})$$

We need to find a matrix $\mathbf{U}^*$ for which perturbations that satisfy the constraints do not change the objective function $F(\mathbf{U})$. Instead we find the stationary points of Lagrangian $\mathbb{L}$ with respect to both $\mathbf{U}$ and $\mathbf{\Lambda}$. The extrema of the unconstrained objective $\mathbb{L}$ are the extrema of the original constrained problem. The extrema to $\mathbb{L}$ occurs when:

$$\nabla F(\mathbf{U}) + \Lambda\nabla(\mathbf{U}^T\mathbf{U} - \mathbf{I})) = 0 \tag{8.3.2.7}$$

This can be expressed as:

$$\frac{\partial \mathbb{L}}{\partial \mathbf{U}} = 0 \quad \text{and} \quad \frac{\partial \mathbb{L}}{\partial \mathbf{\Lambda}} = 0 \tag{8.3.2.8}$$

Under the KKT necessary conditions the partial derivatives must equal zero. Finding partial derivatives with respect to $\mathbf{U}$ in the first expression of Eq.(8.3.2.8) and

considering that $\mathbf{\Omega}_w = \sum_{j=1}^{n} w_j \mathbf{x}_j \mathbf{x}_j^T$ we have:

$$\begin{aligned} \frac{\partial \mathbb{L}}{\partial \mathbf{U}} &= 0 \\ -2\left(\sum_{j=1}^{n} w_j \mathbf{x}_j \mathbf{x}_j^T \right)\mathbf{U} + 2\mathbf{U}\mathbf{\Lambda} &= 0 \\ -2\mathbf{\Omega}_w \mathbf{U} + 2\mathbf{U}\mathbf{\Lambda} &= 0 \\ \mathbf{\Omega}_w \mathbf{U} &= \mathbf{U}\mathbf{\Lambda} \\ \mathbf{U}^T \mathbf{\Omega}_w \mathbf{U} &= \mathbf{\Lambda} \end{aligned}$$

The second expression of Eq.(8.3.2.8) ensures that the constraints are satisfied. We find the partial derivatives with respect to $\mathbf{\Lambda}$, that is, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. This shows that $\mathbf{U}$ are orthonormal eigenvectors and the Lagrangian multiplier $\mathbf{\Lambda}$ is the eigenvalue matrix of the covariance matrix $\mathbf{\Omega}_w$. ■

Theorem 8.3.2.2: $S_{(2,1)}$-PCA solution is rotationally invariant.

Proof: The proof is similar to the one provided in Section 8.2.2. It derives from the the fact that the $L_{(2,1)}$ norm of a vector equals the L_2 norm of that vector and the well known fact that L_2 norm is rotational invariant. So, if $\boldsymbol{R}$ is a rotation matrix, then $||\boldsymbol{R}(\mathbf{x}_j) - \boldsymbol{R}(\mathbf{U}\mathbf{U}^T\mathbf{x}_j)||_{2,1} = ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2$. From orthogonality of R transform, that is, $\boldsymbol{R}^T\boldsymbol{R} = \mathbb{I}$, we have that $||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_{2,1} = ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2 = ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2 = ||\boldsymbol{R}^T\boldsymbol{R}(\mathbf{x}_j) - \boldsymbol{R}^T\boldsymbol{R}(\mathbf{U}\mathbf{U}^T\mathbf{x}_j)||_2 = ||\boldsymbol{R}^T(\boldsymbol{R}(\mathbf{x}_j)) - \boldsymbol{R}(\mathbf{U}\mathbf{U}^T\mathbf{x}_j))||_2 = ||\boldsymbol{R}(\mathbf{x}_j) - \boldsymbol{R}(\mathbf{U}\mathbf{U}^T\mathbf{x}_j)||_2$. This means that $S_{(2,1)}$-PCA solution is rotationally invariant. ■

Method 2 - Maximizing the Variance

The $S_{(2,1)}$-PCA problem can also be formulated as a maximization problem, where the variances of $\mathbf{U}^T\mathbf{X}$ are maximized, under the same constraints $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. The constraint optimization problem is:

$$\mathbf{U}^* = \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\arg\max}\Big(Var(\mathbf{U}^T\mathbf{X})\Big) \underset{\mathbf{U}^T\mathbf{U}=\mathbf{I}}{=} \arg\max\Big(\mathbf{U}^T\mathbf{\Omega}_\mathbf{w}\mathbf{U}\Big) \tag{8.3.2.9}$$

where $Var(\mathbf{U}^T\mathbf{X})$ is the variance of $\mathbf{U}^T\mathbf{X}$ and $\mathbf{\Omega}_\mathbf{w}$ is the weighted covariance matrix.

Theorem 8.3.2.3 The optimal solution to Eq.(8.3.2.9) is given by the eigenvectors of the weighted covariance matrix $\mathbf{\Omega}_\mathbf{w}$.

Proof: As in Theorem 8.3.2.1 we define the Lagrangian $\mathbb{L}(\mathbf{U}, \mathbf{\Lambda})$ as a new objective function to eliminate the constraint:

$$\mathbb{L}(\mathbf{U}, \mathbf{\Lambda}) = G(\mathbf{U}) - \mathbf{\Lambda} T_r(\mathbf{U}^T\mathbf{U} - \mathbf{I}))$$

where

$$G(\mathbf{U}) = \mathbf{U}^T\mathbf{\Omega_w}\mathbf{U} = \sum_{j=1}^{n} \mathbf{U}^T w_j \mathbf{x}_j \mathbf{x}_j^T \mathbf{U}$$

and $\mathbf{\Lambda}$ is the Lagrangian multiplier. By substituting $G(\mathbf{U}) = \sum_{j=1}^{n} \mathbf{U}^T w_j \mathbf{x}_j \mathbf{x}_j^T \mathbf{U}$, we have:

$$\mathbb{L}(\mathbf{U}, \mathbf{\Lambda}) = \sum_{j=1}^{n} \mathbf{U}^T w_j \mathbf{x}_j \mathbf{x}_j^T \mathbf{U} - T_r \mathbf{\Lambda}(\mathbf{U}^T\mathbf{U} - \mathbf{I}) \tag{8.3.2.10}$$

The extrema of the unconstrained objective $\mathbb{L}$ are the extrema of the original constrained problem. By the necessary condition the optimal solution of $\mathbb{L}$ satisfies:

$$\nabla G(\mathbf{U}) - \Lambda \nabla(\mathbf{U}^T\mathbf{U} - \mathbf{I}) = 0 \tag{8.3.2.11}$$

We need to find a matrix $\mathbf{U}^*$, a solution to our constraint optimization problem for some $\mathbf{\Lambda}$. Under the KKT necessary conditions the partial derivatives must equal zero. The partial derivatives of Eq.(8.3.2.10) with respect to $\mathbf{U}$ and to Λ are given as:

$$\frac{\partial \mathbb{L}}{\partial \mathbf{U}} = 0 \quad \text{and} \quad \frac{\partial \mathbb{L}}{\partial \mathbf{\Lambda}} = 0 \tag{8.3.2.12}$$

Thus, from the first equation of Eq.(8.3.2.12), we have:

$$\begin{aligned}
\frac{\partial \mathbb{L}}{\partial \mathbf{U}} &= 0 \\
2\sum_{j=1}^{n} w_j \mathbf{x}_j \mathbf{x}_j^T \mathbf{U} - 2\mathbf{U}\mathbf{\Lambda} &= 0 \\
2\mathbf{\Omega}_w \mathbf{U} - 2\mathbf{U}\mathbf{\Lambda} &= 0 \\
\mathbf{\Omega}_w \mathbf{U} &= \mathbf{U}\mathbf{\Lambda} \\
\mathbf{U}^T\mathbf{\Omega}_w\mathbf{U} &= \mathbf{\Lambda}
\end{aligned}$$

The second expression of Eq.(8.3.2.12) ensures that the constraints are satisfied by finding the partial derivatives with respect to $\mathbf{\Lambda}$, that is, $\mathbf{U}^T\mathbf{U} - \mathbf{I} = 0$ or $\mathbf{U}^T\mathbf{U} = \mathbf{I}$.

This shows that $\mathbf{U}$ are orthonormal eigenvectors and the Lagrangian multiplier $\mathbf{\Lambda}$ is the eigenvalue matrix of the weighted covariance matrix $\mathbf{\Omega}_w$. Moreover, we showed that the solution obtained by minimizing the total error in Theorem 8.3.2.1 and the one obtained by maximizing the variance in Theorem 8.3.2.3 are equal.

Theorem 8.3.2.4 - Global Optimal Solution to $S_{(2,1)}$-PCA Problem:
The $S_{(2,1)}$-PCA problem has global optimal solution given by the eigenvectors of the weighted covariance matrix $\mathbf{\Omega}_w = \mathbf{X}\mathbf{W}\mathbf{X}^T$ where

$$W_{n\times n} = \begin{pmatrix} S_1 \cdot w_1 & 0 & \cdots & 0 \\ 0 & S_2 \cdot w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_n \cdot w_n \end{pmatrix}$$

Proof: In the $S_{(2,1)}$-PCA minimization problem given by Eq. (8.3.2.6), the error function f_j is given as $f_j(\mathbf{U}) = w_j\sqrt{\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{U}\mathbf{U}^T\mathbf{x}_j} = w_j||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2 = |w_j\sqrt{||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j}||_2^2|$. The error function f_j as a function of $\mathbf{U}\mathbf{U}^T$ is strictly convex, therefore, the total error is convex as the sum of convex functions $F(\mathbf{U}\mathbf{U}^T) = \sum_{j=1}^n f_j(\mathbf{U}\mathbf{U}^T)$. From the fundamental theorem of global optimality, the local minimum is global and unique for strictly convex functions [103]. This means that if $\mathbf{U}^*\mathbf{U}^{*T}$ is a stationary point, then the function value at that point reaches the global minimum. Moreover, $\mathbf{U}^*\mathbf{U}^{*T}$ is a unique global minimum point. If $\mathbf{U}^*$ is a stationary point of function $F(\mathbf{U})$, then it is easy to see that $\mathbf{U}^*\mathbf{U}^{*T}$ is a stationary point of F as a function of $\mathbf{U}\mathbf{U}^T$, from which it derives that the global minimum of F is reached for $\mathbf{U}\mathbf{U}^T$. The global minimum of F as a function of $\mathbf{U}$ is the same as the global minimum of F as a function of $\mathbf{U}\mathbf{U}^T$. Therefore, $F(\mathbf{U}^*)$ is a global minimum and since $\mathbf{U}^*\mathbf{U}^{*T}$ is unique, the point $\mathbf{U}^*$ is unique up to an orthogonal transformation.

8.3.3 Numerical Implementation of the $S_{(2,1)}$-PCA

Description

Structure-(2,1) Principles Components Analysis ($S_{(2,1)}$-PCA) algorithm uses SSIM coefficients to compute a weighted data matrix. However, unlike S_2-PCA which applies

the standard PCA on the weighted data matrix, $S_{(2,1)}$-PCA applies R_1-PCA [81] to obtain principal components. The use of the $L_{(2,1)}$-norm makes the $S_{(2,1)}$-PCA algorithm less sensitive to outliers than the standard PCA and the S_2-PCA. As in S_2-PCA algorithm, we set any negative SSIM coefficient to zero. With this algorithm, we compare two different norms, L_2 and $L_{(2,1)}$. The $S_{(2,1)}$-PCA algorithm can thus be described as:

Structure-(2,1) Principles Components Analysis - $S_{(2,1)}$-PCA Algorithm:

INPUT: Data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ and vector $\mathbf{y} \in \mathbb{R}^{m \times 1}$

INITILIZE:

Compute $S_j = \frac{2\sigma_{x_j,y}+c_2}{\sigma_{x_j}^2+\sigma_y^2+c_2}$, for $j = 1, 2, \cdots, n$

If $S_j < 0$, set S_j to zero

Compute the observation weight matrix $\mathbf{S}$ according to Eq. (8.3.2.2)

Compute the weighted data matrix: $\mathbf{X}_w = \mathbf{X}\sqrt{\mathbf{S}}$

set $\mathbf{X}_w$ to $\mathbf{X}$, and center $\mathbf{X}$ with zero mean

Compute the initial weighted covariance matrix $\mathbf{\Omega}_0 = \mathbf{X}\mathbf{X}^T$

Compute the matrices of eigenvalues and eigenvectors $\mathbf{\Lambda}_0$, $\mathbf{U}_0$ of $\mathbf{\Omega}_0$

Compute the initial error $e_j = (\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j\mathbf{U}_0\mathbf{U}_0^T\mathbf{x}_j)^{\frac{1}{2}}$, for $j = 1, 2, \cdots, n$

Compute c = median($\mathbf{e}_j$)

PROCEDURE:

Set $\mathbf{\Omega}_0, \mathbf{\Lambda}_0, \mathbf{U}_0$ to $\mathbf{\Omega}, \mathbf{\Lambda}, \mathbf{U}$

Update $\mathbf{U}$ using the power method

Compute the weights for variables $w_j = \frac{c}{\sqrt{\mathbf{x}_j^T\mathbf{x}_j - \mathbf{x}_j\,\mathbf{U}\mathbf{U}^T\mathbf{x}_j}}$, for $j = 1, 2, \cdots, n$

Compute the weighted covariance matrix $\mathbf{\Omega}_w = \sum_{j=1}^{n} w_j\mathbf{x}_j\mathbf{x}_j^T$

Compute $\mathbf{U}_{new} = \mathbf{\Omega}_w\ \mathbf{U}$

Orthogonalize $\mathbf{U}_{new}$

Set $\mathbf{U}_{new}$ to $\mathbf{U}$

Compute $\mathbf{\Lambda} = \mathbf{U}^T\ \mathbf{\Omega}_w\ \mathbf{U}$

Repeat until converges

OUTPUT: matrix of eigenvectors $\mathbf{U}$

8.4 Applications of S_2-PCA and $S_{(2,1)}$-PCA to Change Detection

In this section we describe the experiments and the results in details. Note that S_2-PCA is sometimes referred as S-PCA in this section.

We conducted experiments for two different blocks of interest **b** from real MR images. We compared the estimated the background of **b** backgr(**b**) by three approaches: 1) SSIM alone, 2) the EigenBlockCD algorithm + PCA, and 3) the EigenBlockCD algorithm + S-PCA.

In the first set of experiments we investigate the use of SSIM alone as a similarity measure to compute the background of the block of interest, namely as the best matching block. We provide two examples for two different blocks of interest **b**. In each example, the background of **b**, namely as backgr(**b**), is determined from inquiry blocks **B** using: SSIM alone, the EigenBlockCD algorithm + standard PCA and the EigenBlockCD algorithm + S-PCA. Various experiments for different blocks of interest **b** are shown in Figs. 8.4.1 through 8.4.7.

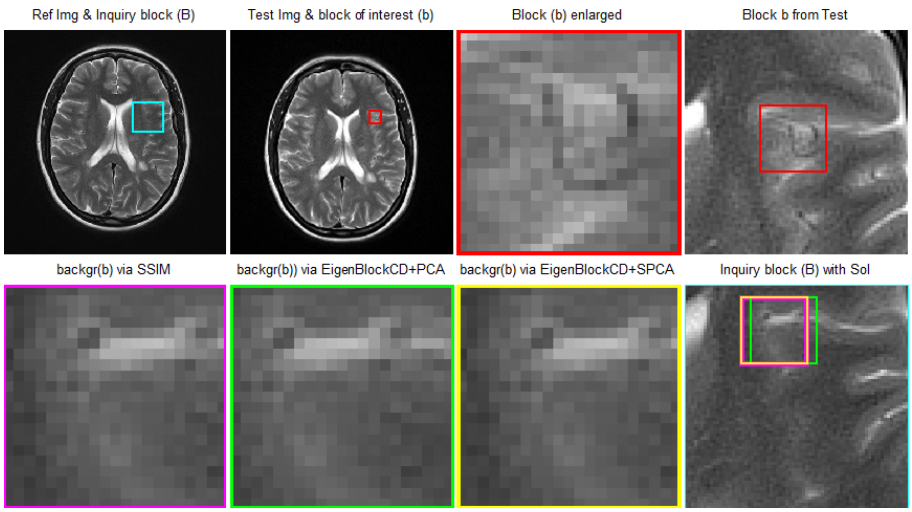

Figure 8.4.1: Top row from left to right: Reference image, test image, block of interest enlarged and a sub-image from the test image containing the block of interest. Bottom row from left to right: backgr(**b**) via SSIM, the backgr(**b**) via the EigenBlockCD algorithm + PCA, the backgr(**b**) via the EigenBlockCD algorithm+S-PCA and the inquiry block **B** containing the three solutions.

Two MR images of the brain of the same patients taken at two different times are used in our experiments, namely as reference and test images. For the first block of

interest **b** as shown in Figs. 8.4.1 to 8.4.4, we compare the estimated backgr(**b**) in four different situations: 1) no changes were added to input images in order to learn the backgr(**b**) which is considered the true backgr(**b**) for that particular block of interest. Fig. 8.4.1 show that the backgr(**b**) computed by the three algorithms are very similar. Moreover, the backgr(**b**) via SSIM and S-PCA are exactly the same. We consider that block as the true backgr(**b**). 2) Small changes are added to the block of interest in

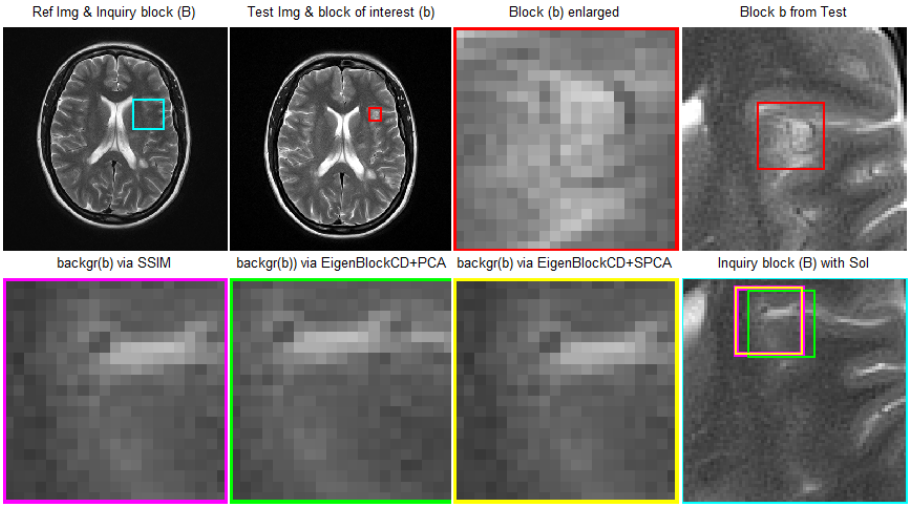

Figure 8.4.2: Top row from left to right: Reference image, test image, block of interest enlarged and a sub-image from the test image containing the block of interest. Bottom row from left to right: backgr(**b**) via SSIM, the backgr(**b**) via the EigenBlockCD algorithm + PCA, the backgr(**b**) via the EigenBlockCD algorithm+S-PCA and the inquiry block **B** containing the three solutions.

the test image by increasing the intensity of the pixels in the block of interest **b**. As Fig. 8.4.2 shows, the backgr(**b**) obtained by the three algorithms is again almost the same. Moreover, the backgr(**b**) via SSIM and the S-PCA are the same. 3) Moderate changes are added to the block of interest in the test images in a similar fashion, to mimic the appearance of a new lesson. The backgr(**b**) obtained by the EigenBlockCD + PCA and the EigenBlockCD + S-PCA algorithms are almost the same as those shown in Fig. 8.4.3. However, the backgr(**b**) computed by SSIM alone is different block in the inquiry block, which means that it is different from the true backgr(**b**) defined previously. 4) Similarly, moderate changes are added to backgr(**b**) in the reference image, to mimic the disappearance of an existing lesson. The backgr(**b**) obtained by EigenBlockCD+ PCA and EigenBlockCD + S-PCA algorithms are almost the same, as shown in Fig. 8.4.4. But the backgr(**b**) computed by SSIM alone is different block

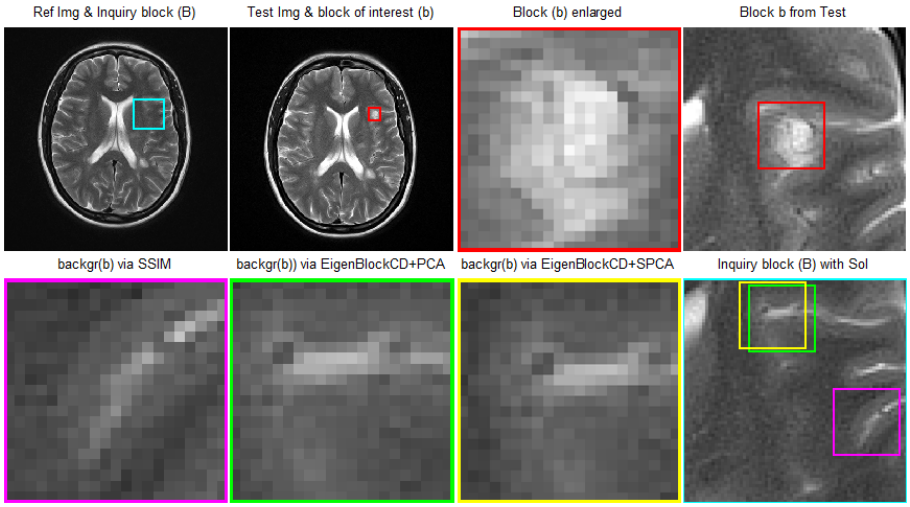

Figure 8.4.3: Top row from left to right: Reference image, test image, block of interest enlarged and a sub-image from the test image containing the block of interest. Bottom row from left to right: backgr($\mathbf{b}$) via SSIM, the backgr($\mathbf{b}$) via the EigenBlockCD algorithm + PCA, the backgr($\mathbf{b}$) via the EigenBlockCD algorithm+S-PCA and the inquiry block $\mathbf{B}$ containing the three solutions.

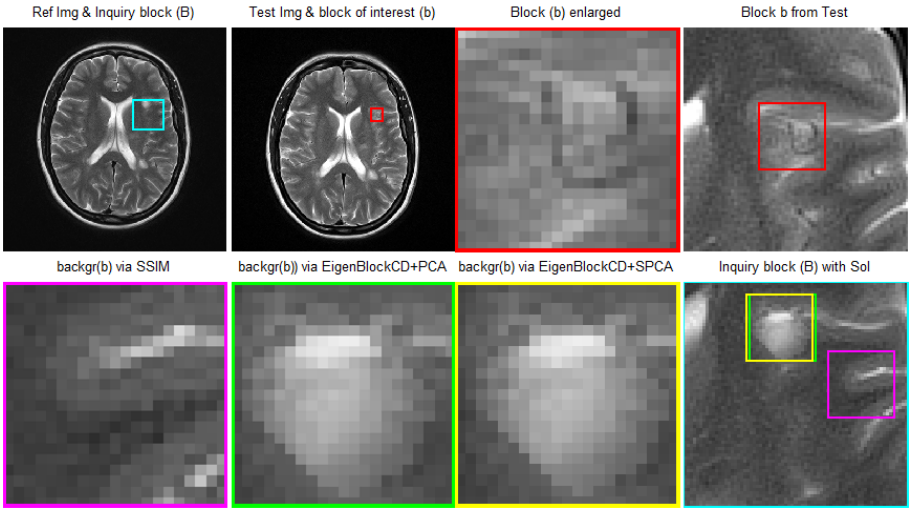

Figure 8.4.4: Top row from left to right: Reference image, test image, block of interest enlarged and a sub-image from the test image containing the block of interest. Bottom row from left to right: backgr($\mathbf{b}$) via SSIM, the backgr($\mathbf{b}$) via the EigenBlockCD algorithm + PCA, the backgr($\mathbf{b}$) via the EigenBlockCD algorithm+S-PCA and the inquiry block $\mathbf{B}$ containing the three solutions.

in the inquiry block, that is, it is different from the true backgr($\mathbf{b}$) defined previously.

For the second block of interest $\mathbf{b}$ as shown in Figs. 8.4.5 to 8.4.7, we compare the estimated backgr($\mathbf{b}$) in three different situations: 1) to learn the backgr($\mathbf{b}$), no changes were added to input images. The backgr($\mathbf{b}$) is considered the true backgr($\mathbf{b}$) for that particular block of interest. Fig. 8.4.5 show that the backgr($\mathbf{b}$) computed by the three algorithms are very similar. Moreover, the backgr($\mathbf{b}$) via SSIM and S-PCA are exactly

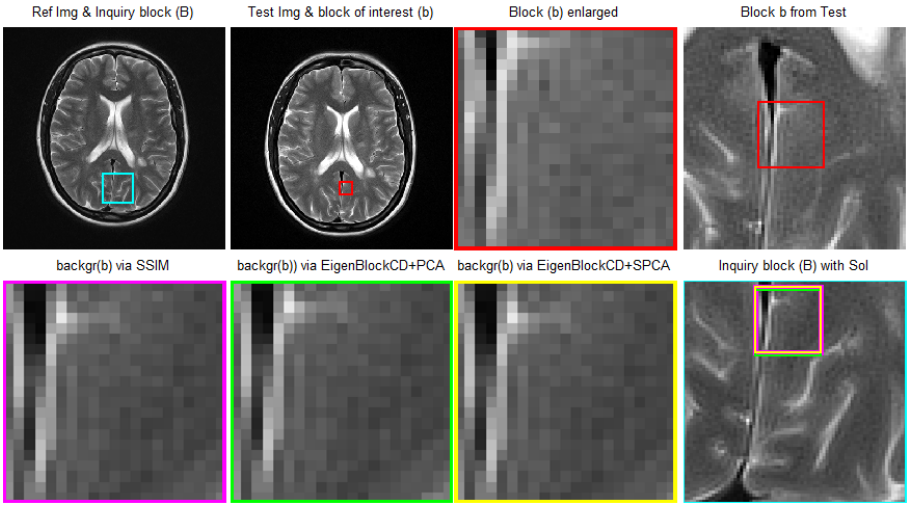

Figure 8.4.5: Top row from left to right: Reference image, test image, block of interest enlarged and a sub-image from the test image containing the block of interest. Bottom row from left to right: backgr(**b**) via SSIM, the backgr(**b**) via the EigenBlockCD algorithm + PCA, the backgr(**b**) via the EigenBlockCD algorithm+S-PCA and the inquiry block **B** containing the three solutions.

the same. We consider that block as the true backgr(**b**). 2) Moderate changes are

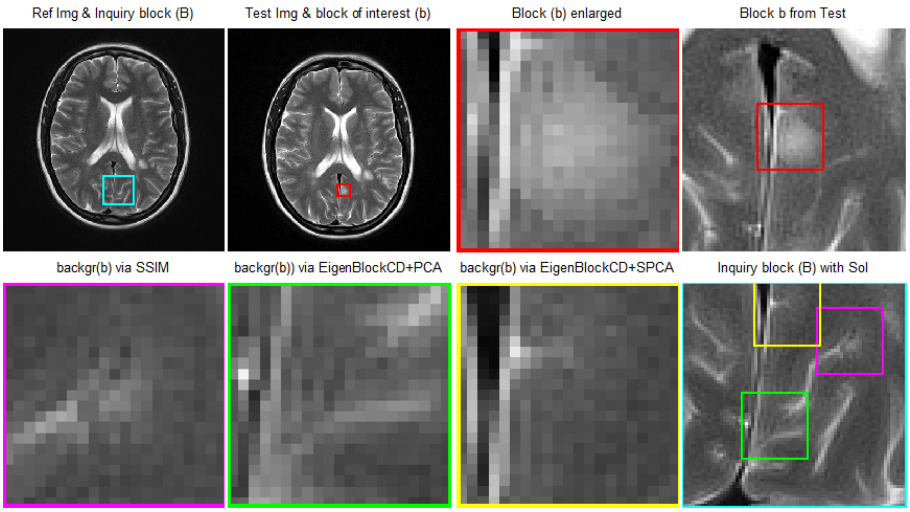

Figure 8.4.6: Top row from left to right: Reference image, test image, block of interest enlarged and a sub-image from the test image containing the block of interest. Bottom row from left to right: backgr(**b**) via SSIM, the backgr(**b**) via the EigenBlockCD algorithm + PCA, the backgr(**b**) via the EigenBlockCD algorithm+S-PCA and the inquiry block **B** containing the three solutions.

added to the block of interest in the test image to mimic the appearance of a new lesson. As Fig. 8.4.6 shows, only backgr(**b**) obtained by EigenBlockCD + S-PCA is close to the true backgr(**b**). In contrary, the backgr(**b**) computed by SSIM alone and EigenBlockCD + PCA represent different blocks in the inquiry block, that is, they are different from the true backgr(**b**). 3) Moderate changes are added to the backgr(**b**) in

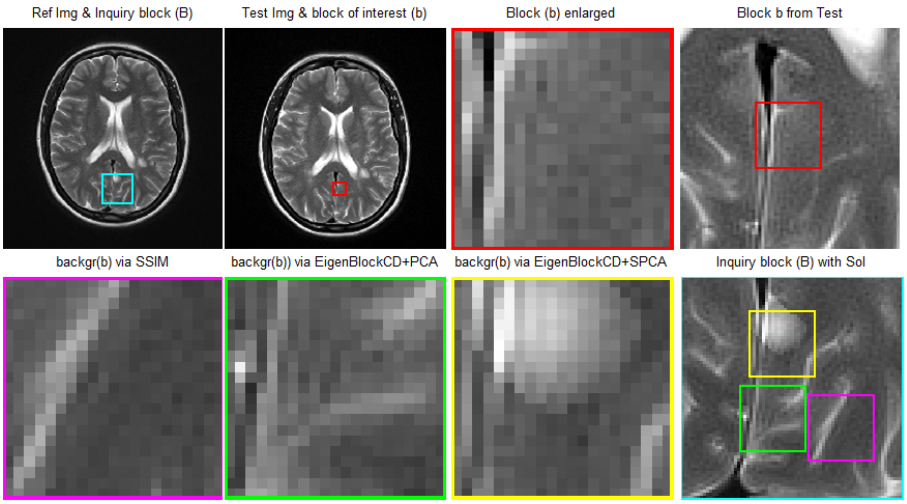

Figure 8.4.7: Top row from left to right: Reference image, test image, block of interest enlarged and a sub-image from the test image containing the block of interest. Bottom row from left to right: backgr(**b**) via SSIM, the backgr(**b**) via the EigenBlockCD algorithm + PCA, the backgr(**b**) via the EigenBlockCD algorithm+S-PCA and the inquiry block **B** containing the three solutions.

the reference image which mimics a disappearance of an existing lesson. As Fig. 8.4.7 shows, only the backgr(**b**) obtained by the EigenBlockCD + S-PCA algorithm is close to the true backgr(**b**). In contrary, the backgr(**b**) computed by SSIM alone and the EigenBlockCD + PCA algorithm represent different blocks in the inquiry block, that is, they are different from the true backgr(**b**).

Dimensions of the Dictionary				
Variance	PCA	R1-PCA	S_2-PCA	$S_{(2,1)}$-PCA
100%	441	441	441	441
99%	137	114	56	45
98%	84	71	43	21
96%	40	36	19	10
94%	23	22	12	7
92%	16	15	9	6
90%	12	12	7	4
88%	10	10	6	4
86%	9	9	5	3
84%	8	8	4	3
82%	7	7	4	3
80%	6	6	3	2
78%	6	5	3	2
76%	5	5	3	2
74%	5	5	2	2
60%	4	4	2	1
40%	2	2	1	1
20%	1	1	1	1

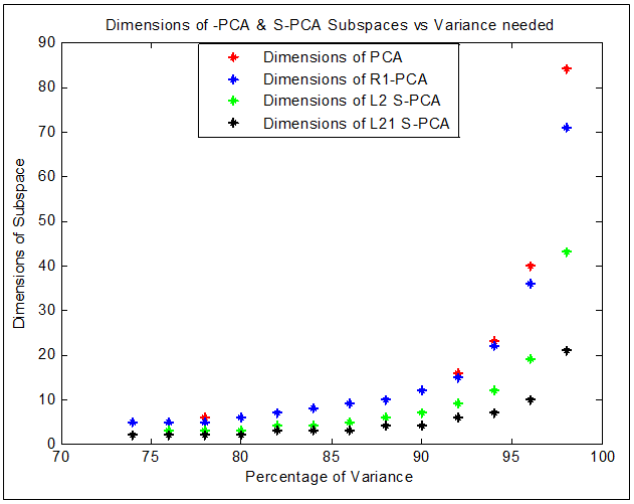

Figure 8.4.8: Left: Table showing the percentage of the variance used to create the subspaces. The first, second, third, fourth and the fifth columns show the variance, the dimensions of the PCA, the R_1-PCA, the S_2-PCA and $S_{(2,1)}$-PCA subspaces respectively. Right: Scatter plot of dimensions of the four subspaces versus the percentage of the variance in the data.

The second set of experiments compare the efficiency of dimension reduction by the

standard PCA, the R_1-PCA and our two structure-PCAs, the S_2-PCA and the $S_{(2,1)}$-PCA. The efficiency of dimension reduction is measured by the dimension of subspaces versus percentage of the data, variance contained in the subspaces. The lower the values of dimension of subspaces for the same percentage of variance in the data the more efficient the projections of the data onto these subspaces. For the same percentage of the variance in the data, dimensions of the subspace created via the S-PCA are much lower than dimensions of subspace via the standard PCA and the R_1-PCA as shown in Fig 8.4.8. Significant dimensionality reduction leads to a sparser representation of the data and increases computation efficiency.

Percentage of variance versus number of eigenvalues for the first 15 eigenvalues

Number of Eigenvectors	PCA	R1-PCA	S_2-PCA	$S_{(2,1)}$-PCA
1	24	23	55	67
2	44	42	75	82
3	59	57	80	87
4	71	71	85	90
5	78	78	87	92
6	80	81	89	93
7	83	83	90	94
8	85	85	91	95
9	87	87	92	96
10	88	88	93	96
11	89	89	94	96
12	90	90	94	97
13	91	91	95	97
14	91	92	95	97
15	92	92	95	97

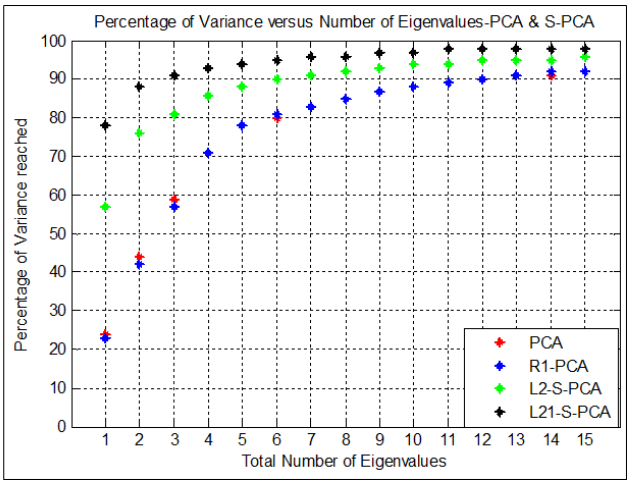

Figure 8.4.9: Left: Percentage of variance versus number of eigenvalues for the first fifteen eigenvalues. First column shows the first number of eigenvalues, second, third, fourth and fifth columns show the percentage of the variance. Right: Scatter plot showing the percentage of the variance vs the number of eigenvalues for the four subspaces, the PCA, the R_1-PCA, the S_2-PCA and the $S_{(2,1)}$-PCA.

In the third set of experiments we computed the percentage variance average for the first 15 principle components. The percentage variance average reached by the same first number of principal components is higher for the structure S_2-PCA and $S_{(2,1)}$-PCA than for the standard PCA and the R_1-PCA as shown in Fig. 8.4.9. For example, 55% and 67% of the total variance in the S_2-PCA and the $S_{(2,1)}$-PCA is reached by the first principal component versus 24% and 23% in the PCA and in the R_1-PCA respectively. 75% and 82% of the total variance in the S_2-PCA and the $S_{(2,1)}$-PCA is reached by the first two principal components versus 44% and 42% in the PCA and in the R_1-PCA respectively. This indicates that we could reduce the dimensions to as low as 3 from 441 and preserve 87% of the total variance in the $S_{(2,1)}$-PCA or 80% of the total variance

in the S_2-PCA.

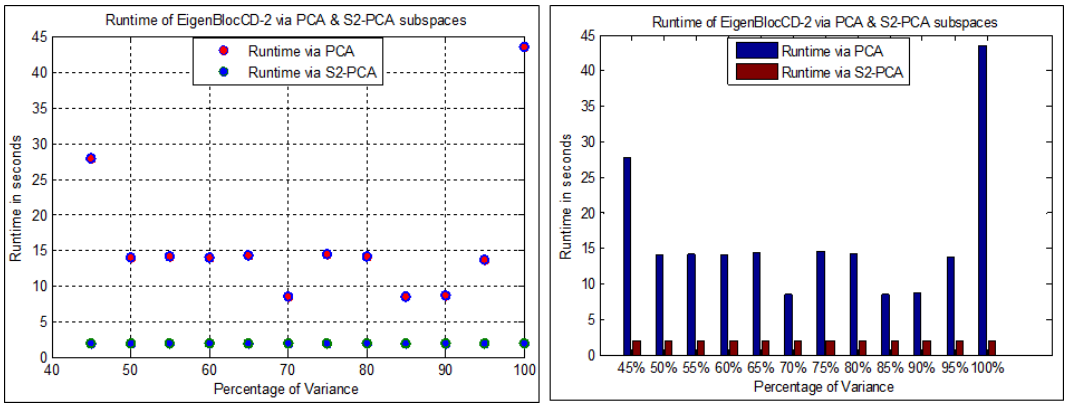

Figure 8.4.10: Left: Scatter plot of the runtime of the EigenBlockCD-2 algorithm via PCA and S_2-PCA versus the percentage of the variance in the data. Right: Bar graph of the runtime of the EigenBlockCD-2 algorithm via PCA and S_2-PCA versus the percentage of the variance in the data

In the fourth set of experiments we compare the runtime of the EigenBlockCD-2 algorithm with data projected into PCA or S_2-PCA subspaces with different percentage of the variance in the data. The algorithm via S_2-PCA runs much faster than via PCA as shown in Fig. 8.4.10. As the variance in the data increases, the runtime via PCA increases whereas the runtime via S_2-PCA stays almost the same as shown in Fig. 8.4.10. We didn't compare the runtime of the R_1-PCA and $S_{(2,1)}$-PCA because they both are iterative and computationally expensive.

Variance %	PCA		S_2-PCA	
	Shift_v	Shift_h	Shift_v	Shift_h
10	14	-12	20	5
23	14	-12	20	5
44	22	12	20	5
50	23	12	20	5
60	21	4	20	5
70	21	4	20	5
80	21	4	20	5
90	20	4	20	5
92	20	5	20	5
⋮	⋮	⋮	⋮	⋮
100	20	5	20	5

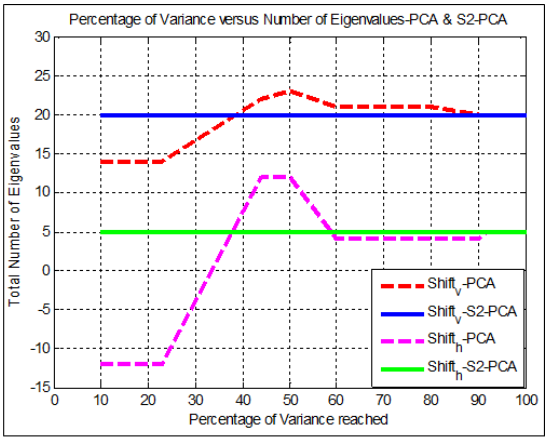

Figure 8.4.11: Left: Table showing the alignment parameters (horizontal and vertical shifts) calculated by the EigenBlockCD-2 via PCA and S_2-PCA for different percentage of variance in the data. Right: scatter plot showing the same alignment parameters.

The fifth set of experiments test the effect of estimating the alignment parameters used in the co-registration step by the EigenBlockCD-2 algorithm via PCA and S_2-PCA for different percentage of variance in the data, the horizontal and the vertical

shifts. The co-registration step via PCA is not stable for smaller percentages of the variance because the obtained co-registration parameters are different each time the percentage of the variance changes as the table and the scatter plots in Fig. 8.4.11 show. On the other hand, the co-registration step via S_2-PCA is stable even for very small percentages of the variance (i.e., 10%).

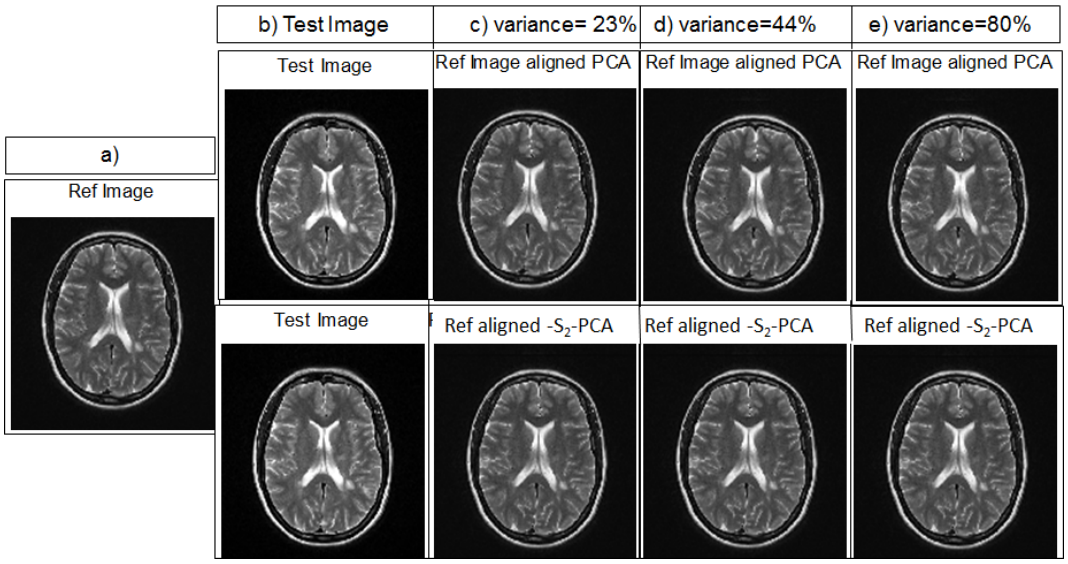

Figure 8.4.12: a) Reference image is a real T2 MR of the brain in 2010. b) column, follow up T2 MR image in 2012 named as test image. c) to e) Alignment of reference image to test via PCA (top) and S_2-PCA (bottom) with 23 %, 44%, and 80%, of variance in the data respectively.

As variance increases, the two translation parameters computed via PCA are getting closer to the corresponding parameters computed via S_2-PCA. Fig. 8.4.12 visually demonstrates the performance of the algorithm employing PCA in the co-registration step is as good as the algorithm employing S_2-PCA if total percentage of the variance in the data is 92%. When percentage of the variance increases from 23% to 44%, the vertical shift needed to align the reference and test images estimated by EigenBlockCD-2 via PCA, changes from 14 to 22 pixels while the vertical shift identified via S_2-PCA remains at 20 pixels. The horizontal shift identified by EigenBlockCD-2 via PCA changes from -12 to 12 pixels, while the horizontal shift identified via S_2-PCA is always 5 pixels.

In the seventh set of experiments, the 2601 training blocks from the dictionary are projected into the PCA and S_2-PCA subspaces. Distances between each training block and its approximated (reconstructed) block using L_1 and L_2 norms are computed as: $||\mathbf{x}_j - \hat{\mathbf{x}}_j||_1 = ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_1$ and $||\mathbf{x}_j - \hat{\mathbf{x}}_j||_2 = ||\mathbf{x}_j - \mathbf{U}\mathbf{U}^T\mathbf{x}_j||_2$. Note that S-PCA

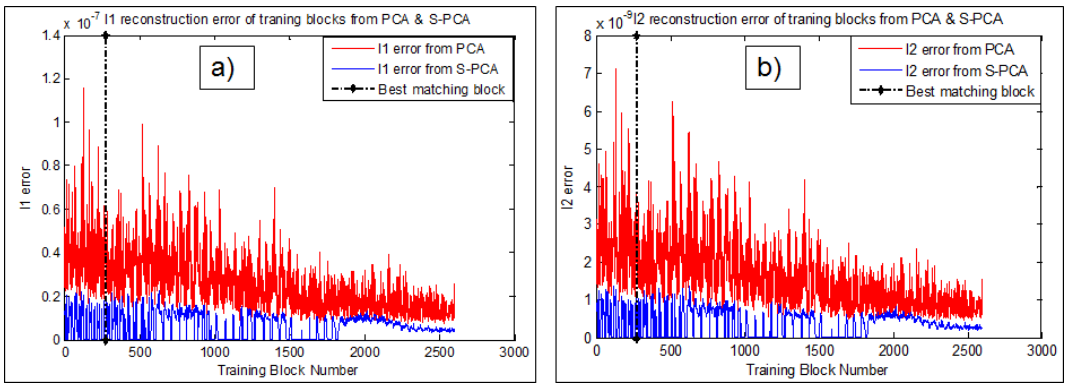

Figure 8.4.13: a) and b) The L_1 and the L_2 projection errors of the 2601 training blocks into PCA and S_2-PCA respectively.

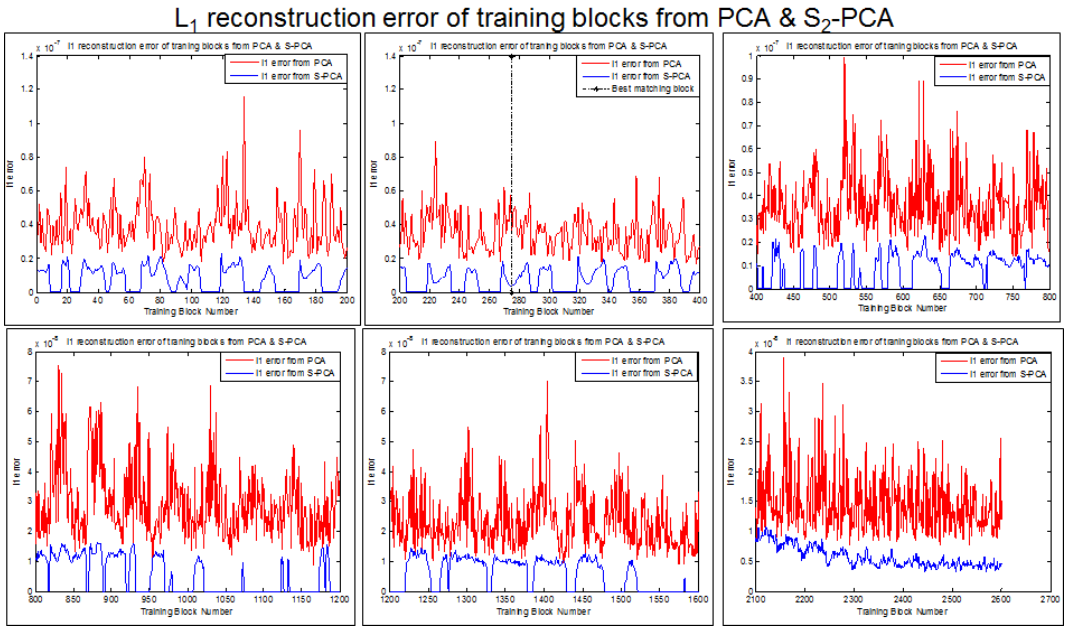

Figure 8.4.14: Shows a more detailed view of Fig. 8.4.13 a), i.e.,projection errors of 200 or 400 training blocks.

in these figures means S_2-PCA. Results in Figs. 8.4.13, 8.4.14, and 8.4.15 show that the L_1 and L_2 errors of the projected training blocks onto S_2-PCA subspace are much lower than the L_1 and L_2 errors of the projected training blocks onto the standard PCA, that is, the S_2-PCA subspace approximate the original data with more accuracy.

In the last set of experiments, we applied the EigenBlockCD-2 algorithm to compute the best matching blocks in the reference image to the block of interest in the test image, with the training blocks projected onto PCA and S_2-PCA subspaces. Fig. 8.4.16 shows that when the dictionary is projected into the subspace obtained by PCA, the EigenBlockCD-2 algorithm computes different best matching blocks to a selected

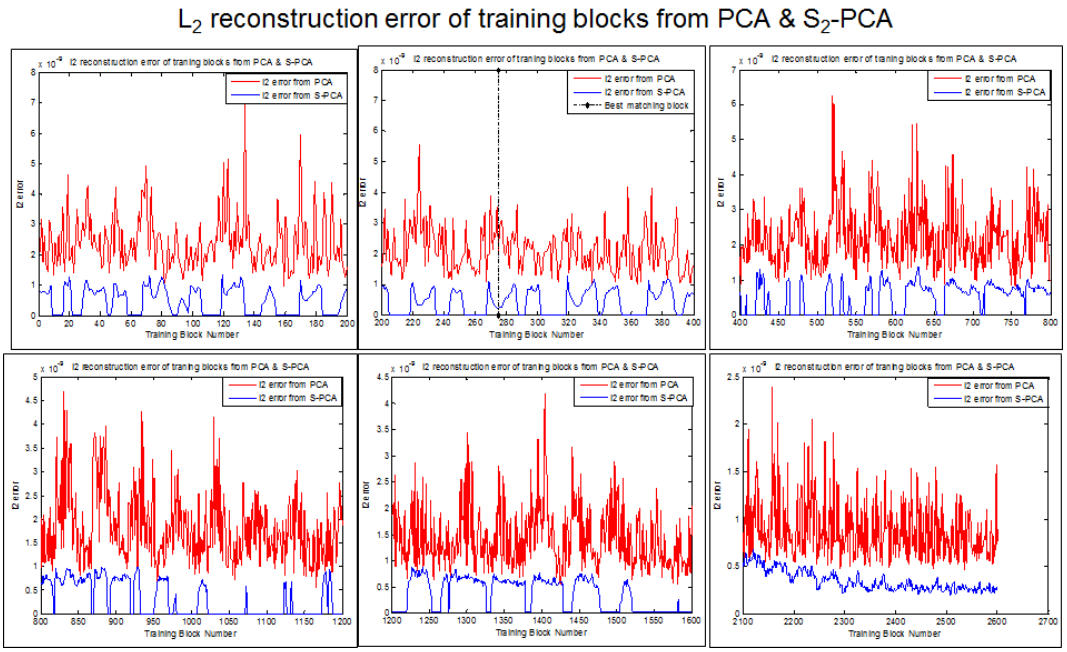

Figure 8.4.15: Shows a more detailed view of Fig. 8.4.13 b), i.e.,projection errors of 200 or 400 training blocks.

block of interest for different percentages of the variance in the data. In contrary, the EigenBlockCD-2 algorithm always computes the same best matching block to the block of interest when the dictionary is projected into the subspace obtained by S_2-PCA even for very low percentages of the variance in the data.

8.5 Summary

We revisited PCA and R_1-PCA, and introduced our newly developed structure principle component analysis S_2-PCA and $S_{(2,1)}$-PCA. We investigated several approaches from the recent literature which address the robustness of PCA and introduced our robustyfing approaches, tailored to our application, through special weights on the main diagonal of covariance matrix. Since the objective of change detection is to estimate the common background between the same anatomical images taken at different times, we chose SSIM indixes between a block of interest and blocks of the same size from its corresponding inquiry block as the proper weights for our change detection algorithms and provided the necessary proofs for our structure PCA problems, the S_2-PCA and the $S_{(2,1)}$-PCA problems.

We provided a formulation of PCA problem based on convex optimization theory

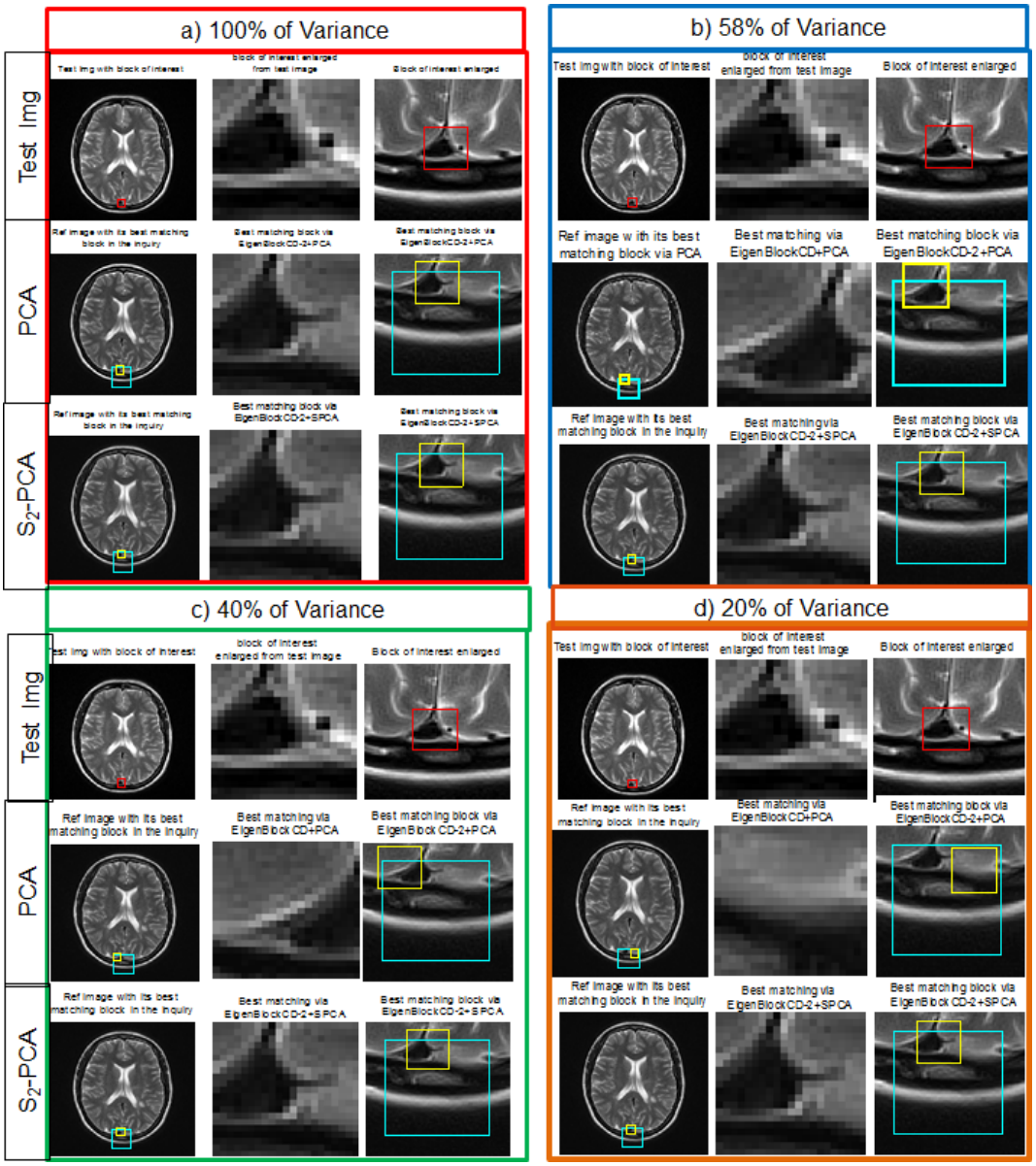

Figure 8.4.16: a) - d) Block detection by the EigenBlockCD-2 via PCA and S_2-PCA with 100%, 58%,40% and 20% of the variance in the data respectively. First rows from left to right in a)-d) show the test image, the block of interest enlarged and a sub-image from the test image containing a larger size of the block of interest. Second & third rows in a)-d) show the best matching block computed by the EigenBlockCD-2 and PCA and S_2-PCA respectively, i.e., a) second rows, from left to right: the reference image containing the best matching block, the computed best matching block enlarged, a sub-image in the reference image containing a larger size of the computed best matching block with 100% of the variance.

and extended this to our S_2-PCA. The use of $L_{(2,1)}$ norm was inspired by the differences in physical meaning of the data matrix $\mathbf{X}$. From this observation we proposed the $S_{(2,1)}$-PCA employing the $L_{(2,1)}$ norm.

Our various experiments showed that the both structure principal component analysis, the S_2-PCA and the $S_{(2,1)}$-PCA, have advantages over the standard PCA and the

R_1-PCA for change detection in the following aspects:

1. Dimensions of both S_2-PCA and $S_{(2,1)}$-PCA decrease much faster than those of the standard PCA and the R_1-PCA as the variance decreases. This effectively finds the optimal subspaces to describe the background of the block of interest, and hence reduces data dimensions dramatically and increases computational efficiency.

2. As the variance decreases, the EigenBlockCD-2 detect changes more accurately when the data are projected onto S_2-PCA then onto standard PCA.

3. The S_2-PCA subspace provides an optimal subspace to search for the background of the block of interest in the reference image than the PCA subspace.

4. Runtime simulations show that the use of the S_2-PCA reduces the computational time by more than 60% on average.

5. Experiments on identifying the best matching block via the EigenBlockCD-2 for the projected data onto standard PCA and onto S_2-PCA show that as the variance decreases the performance of the EigenBlockCD-2 with data projected onto the S_2-PCA is better than when data is projected onto the standard PCA. The best matching block identified in the PCA case is the same only for higher percentage of the variance in the data.

Chapter 9

Conclusions And Future Work

9.1 Summary of Dissertation

Change detection methods play a critical role in medical image processing. In this dissertation, we investigated the existing mathematical frameworks used in change detection problems, proposed three new adaptive dictionary learning based change detection algorithms for detecting the changes of consecutive MR images as well as serial MR images in two dimensional space, extended the algorithms for detecting changes of consecutive volumetric data and proposed a new robust subspace learning as a most efficient way to represent the data.

First, we defined a new mathematical model to describe change detection problem in general. Our model includes a set of transform functions, a cost function to be minimized and a similarity measure. We then formulated the change detection problem as finding an optimal solution for all its parameters. The mathematical model led us to investigate concrete solutions to the change detection problem.

Based on such general model, we extended previous work in the area of background modeling and dictionary learning techniques and proposed the Adaptive Eigenblock Dictionary Learning (AEDL) algorithm to automatically detect the changes of two or more consecutive medical images of the same anatomical structures taken at different times. The AEDL algorithm uses knowledge of compressed sensing to reconstruct the background of the block of interest. The algorithm detects the significant changes between two consecutive medical images, i.e., the foreground, by approximating the

background of a block of interest with the use of L_1 minimization techniques. The AEDL algorithm also uses the knowledge of principal component analysis (PCA) as a feature extraction, sparsifying and a dimensionality reduction tools. Numerical simulations with both synthetic and real MR images demonstrate that the AEDL algorithm is able to detect clinical changes in MR images of the brain.

While the AEDL provides a new approach among other change detection algorithms and those for MR images in particular, we also looked into other change detection methods which are computationally more efficient.

We proposed the EigenBlock Change Detection EigenBlockCD) algorithm to detect the changes between two or more consecutive MR scans. The EigenBlockCD is based on L_2 minimization, i.e., it uses L_2 norm as similarity measure. The use of PCA with the EigenBlockCD further reduces the data redundancy of dictionary and increases the performance of the algorithm. The numerical simulations show that the EigenBlockCD algorithm finds the significant changes in MR images of the brain in less time compared to the AEDL algorithm.

Although the EigenBlockCD algorithm provides an improvement in terms of computational time, it does not perform well in case of large shifts and rotations. We proposed an improved version of the EigenBlockCD algorithm, namely the EigenBlockCD-2 algorithm, by adding a co-registration step first. As a result, the EigenBlockCD-2 algorithm detects changes between two or more MR images even in the presence of large shifts and rotations. We implemented the same approach for the AEDL algorithm, i.e., the AEDL-2 algorithm. Simulations with synthetic and real MR images demonstrate an excellent performance of the improved versions of the EigenBlockCD-2 and the AEDL-2 algorithms.

The comparison of the AEDL and EigenBlockCD algorithms, led us to look into a thorough analysis of similarity measures. We studied and compared the use of L_1 and L_2 norms as similarity measures of the EigenBlockCD algorithm and provided the most significant features of these norms relevant to the change detection problem. We showed that L_2 norm is invariant to rotational transforms, preserves the distances and the orders. We presented counter examples to showed that L_1 norm is not invariant

to rotations, does not preserve the distances and orders and more importantly it does not preserve the structure.

All the proposed change detection algorithms use PCA for dimensionality reduction of the dictionary and decreasing the computational time. This led us to looking into subspace learning frameworks. We proposed a new robust version of principal component analysis, named Structure Principles Component Analysis (S-PCA), which robustifies the covariance matrix by applying weights based on structure similarity index (SSIM).

The key to the S-PCA is to give more weight to the training samples with a similar structure to the test sample (block of interest) and less weights to those training samples that do not "look like" the test sample. We explored the weighted structured PCA using two different norms, i.e., the $L_2 and \mathrm{L}_{2,1}$ norms. We incorporated S-PCA into the EigenBlockCD-2 algorithm. The numerical results and simulations with real MR images show that the L_2 - S-PCA outperforms the standard L_2 - PCA, the $L_{2,1}$- S-PCA outperforms the $L_{2,1}$ - PCA and finally, the $L_{2,1}$- S-PCA outperforms L_2 - S-PCA.

We extended our change detection algorithms to detect changes between the consecutive 3D volumetric data. We implemented the 3D version of AEDL-2 and EigneBlockCD-2. The simulation with real volumetric MR data offered a very good performance of the 3D versions as confirmed by the radiologist.

Finally, we provided a thorough performance analysis of our change detection algorithms. We used the well known criteria from change detection field and compared our results with those published in the change detection literature. Our experiments showed that both the AEDL-2 and EigneBlockCD-2 algorithms outperfom the existing algorithms.

9.2 Future Work

The background modeling and dictionary learning techniques are widely used for solving change detection problems in a broad range of applications and are still an exciting area of research.

Also, the machine learning theory and its applications have seen a significant progress and have attracted great interest from mathematicians, computer scientists and other cross-disciplinary researchers and industry practitioners.

We will explore other versions of subspace learning and look into adapting them to solving change detection problems in medical imaging.

We will explore the three dimensional dictionary techniques as another way for implementing change detection algorithms for MR volumetric data.

Another aspect that we want to pursue is using our algorithms as routine tools by the radiologists. We would like to investigate parallelizing implementation of the algorithms which we think is necessary for the 3D versions.

We will also look into the possibility of testing our algorithms with a massive set of clinical data.

Bibliography

[1] Turk, M., Pentland, A.: Eigenfaces for Recoregnition. J. Cognitive Neuroscience. vol. 3, no. 1, pp. 71-86, 1991.

[2] Bosc, M., Heitz, F., Armspach, J.P., Namer, I., Gounot, D., Rumbach, L. : Automatic Change Detection in Multi-Modal Serial MRI: Application to Multiple Sclerosis Lesion Evolution, Neuroimage, Vol. 20, 2003.

[3] Patriarche, J.W., Erickson, B.J., A Review of the Automated Detec-tion of Change in Serial Imaging Studies of the Brain. J. Digital Imaging. vol.17, no.3, pp.158-174, 2004.

[4] Radke, R.J., Andra S., Al-Kofahi O., Roysam B.: Image Change Detection Algorithms: a Systematic Survey. IEEE Trans Image Process. Vol. 14, 2005.

[5] Candès, E. J., Romberg, J., Tao, T.: Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information. IEEE Transactions on Information Theory, vol. 52, no. 2. pp. 489-509, 2006.

[6] Candès, E., Romberg, J., Sparsity and Incoherence In Compressive Sampling. J. Inverse Problems. vol. 23 no. 3, pp. 969-985, 2007.

[7] Patriarche, J.W., Erickson, B.J., Part 1. Automated Change Detection and Characterization in Serial MR Studies of Brain-Tumor Patients. J. Digital Imaging. vol. 20, no. 3, pp.203-222, 2007.

[8] Patriarche, J.W., Erickson, B.J: Part 2. Automated Change Detection and Characterization Applied to Serial MR of Brain Tumors may Detect Progression Earlier than Human Experts. J. Digital Imaging. vol. 20, no. 4, pp. 321-328, 2007.

[9] Patriarche, J.W., Erickson, B.J., Change Detection & Characterization: a New Tool for Imaging Informatics and Cancer Research. J. Cancer Inform. vol.4, pp.1-11, 2007.

[10] Cevher, V., Sankaranarayanan, A., Duarte. M., Reddy, D., Baraniuk, R., Chellappa, R.: Compressive Sensing for Background Subtraction. In: 10th European Connference on Computer Vision, vol. 5303, pp. 155-168, 2008.

[11] Seo, H.J., Milanfar, P., A Non-Parametric Approach to Automatic Change Detection in MRI Images of the Brain. In:IEEE International Symposium On Biomedical Imaging: From Nano to Macro, pp. 245-248, IEEE, Boston MA, 2009.

[12] Needell, D., J A Tropp, J.A, CoSaMP: Iterative Signal Recovery From Incomplete and Inaccurate Samples. J. Applied and Computational Harmonic Analysis. vol. 26, no. 3, Publisher: Elsevier Inc., pp. 301-321, 2009.

[13] Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y., Robust Face Recognition Via Sparse Representation. Transactions on Pattern Analysis and Machine In-telligence, IEEE, vol. 31, no. 2, pp. 210-227, 2009.

[14] Wright, J. Ma, Y., Mairaly, J., Sapiroz, G., Huangx, T., Yan, S.,e Sparse Representation For Computer Vision And Pattern Recognition. In: IEEE, vol.98, no. 6, pp. 1031-1044, 2010.

[15] Nguyen, L.H., Tran, T.D: A Sparsity Driven Joint Image Registration And Change Detection Technique For SAR Imagery. ICASSP, pp. 2798-2801, 2010.

[16] Andrew Wagner, A., Wright, J., Ganesh, A., Zhou, Z., Mobahi, H., Ma, Y., Towards a Practical Face Recognition System: Robust Alignment and Illumination by Sparse Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 2, pp. 372-386, 2012.

[17] Simulated Brain Database, `http://mouldy.bic.mni.mcgill.ca/brainweb`

[18] Lustig, M., Donoho, D., Pauly, J.M: Sparse MRI: The application of compressed

sensing for rapid MR imaging. Magnetic Resonance in Medicine, vol.58, no. 6, pp. 1182-95, 2007.

[19] Candés, E., Li, Xiaodong., Ma, Yi., Wright, J., Robust Principal Component analysis. Journal of the ACM, vol. 58, no. 3, 2011.

[20] Zhao, C., Wang, X., Cham, W,K., Background Subtraction via Robust Dictionary Learning, EURASIP Journal on Image and Video Processing, vol, 2011.

[21] Aharon, M., Elad, M., Bruckstein A., K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation, IEEE Transactions on Signal Processing, vol. 54, no. 11, pp. 4311 - 4322, 2006.

[22] Ravishankar, S., Bresler, Y., MR Image Reconstruction From Highly Undersampled k-Space Data by Dictionary Learning, IEEE Transactions on Medical Imaging, vol. 30, no. 5, 2011

[23] Nika, V., Babyn, P., Zhu, H. AEDL Algorithm for Change Detection in Medical Images An Application of Adaptive Dictionary Learning Techniques, in Proc. 15th Int. Conf. on Medical Imaging Computing and Computer Assisted Intervention, MICCAI: STMI, pp 14-21, 2012.

[24] Nika, V., Babyn, P., and Zhu, H., EigenBlock Algorithm for Change Detection An Application of Adaptive Dictionary Learning Techniques," Journal of Computational Science, Vol. 5, Issue 3, pp 527 - 535, 2013.

[25] Nika, V., Babyn, P., and Zhu, H., Change Detection of Medical Images Using Dictionary Learning Techniques and PCA, Medical Imaging 2014: Computer-Aided Diagnosis, in Proc. of SPIE Vol. 9035, 903506, 2014.

[26] Nika, V., Babyn, P., Zhu, H., Change detection of medical images using dictionary learning techniques and principal component analysis. J. Med. Imag., Vol. 1, issue. 2, pp 024502, 2014.

[27] Patriarche, J.W., Erickson, B.J. A Review of the Automated Detection of Change in Serial Imaging Studies of the Brain, J. Digital Imaging, 17(3): 158-174, 2004.

[28] Singh, A. Review Article Digital change detection techniques using remotely-sensed data, International Journal of Remote Sensing, 10:6, 989-1003, 1989.

[29] Ilsever.M, Únsalan, C. Two-Dimensional Change Detection Methods:Remote Sensing Applications,SpringerBriefs in Computer Science, May, 2012.

[30] Aach, T., Kaup, A., Statistical model-based change detection in moving video, Signal Processing, vol. 31, pp. 165-180, March 1993.

[31] Bruzzone, L., Prieto, D. F., Automatic analysis of the difference image for unsupervised change detection, IEEE Transactions on Geoscience and Remote Sensing, vol. 38, no. 3, pp. 1171-1182, May 2000.

[32] Malila, W. A., Change Vector Analysis: an Approach for Detecting Forest Changes with Landsat, in Proceedings, 6th Annual Symposium on Machine Processing of Remotely Sensed Data, pp.326-335,1980.

[33] Johnson, R.D., Kasischke, E.S. Change vector analysis: A technique for the multispectral monitoring of land cover and condition, International Journal Remote Sensing 19(3), 411-426,1998.

[34] Di Stefano, L., Mattoccia, S., Mola, M., A change-detection algorithm based on structure and colour, IEEE Conference on Advanced Video and Signal-Based Surveillance 2003, pp. 252-259,2003.

[35] M. J. Black, D. J. Fleet, and Y. Yacoob, Robustly estimating changes in image appearance, Computer Vision and Image Understanding, vol. 78, no. 1, pp. 8 - 31, 2000.

[36] Bruzzone, L., Prieto, D. F., An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images, IEEE Trans. Image Processing, vol. 11, no. 4, pp. 452 - 466, April 2002.

[37] Lee, J., Park, M., An Adaptive Background Subtraction Method Based on Kernel Density Estimation, Sensors 12, 12279-12300, 2012.

[38] Elgammal, A., Duraiswami, R., Harwood, D., Davis, L., Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance, Proceedings of the IEEE, vol. 90, NO. 7, 2002.

[39] Yasuyo, K., Background modeling by combining joint intensity histogram with time-sequential data, IEEE, 1051-4651, 10, 2010.

[40] Bouwmans, ElBaf, T., F., Vachon, B., Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey, Recent Patents on Computer Science 1, 3, 219-237, 2008.

[41] Stauffer, C., Grimson, W.E.L., Adaptive background mixture models for real-time tracking, Proceedings IEEE, Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, 1999.

[42] Rousseau, F., Faisan, S., Heitz, F., Armspach, J., Chevalier, Y., Blanc, F., Seze, J., Rumbach, L., An a contrario approach for change detection in 3D multimodal images:Application to Multiple Sclerosis in MRI, IEEE Engineering in Medicine and Biology Society (EMBS), pp. 2069-2072, August 2007.

[43] Shen, S., Szameitat, A., and Sterr, A., Detection of infarct lesions from single MRI modality using inconsistency between voxel intensity and spatial location-a 3D automatic approach, IEEE Transactions on Information Technology in Biomedicine, vol. 12, pp. 532 - 540, 2008.

[44] Manjo, J.V., Caballero, C.J., Lull, J.J., Marti, G., Bonmati, M.L., Robles, M., MRI denoising using Non-Local Means, Medical Image Analysis, 12, 514 - 523, 2008.

[45] Wang, Zh., Image Quality Assessment: From Error Visibility to Structural Similarity, IEEE Transactions On Image Processing, vol. 13, no. 4, 2004.

[46] Simulated Brain Database, `http://www.sph.sc.edu/comd/rorden/mricro.htm`

[47] Seo, H. J., and Milanfar, P., Using local regression kernels for statistical object de-

tection, Proceedings of IEEE International Conference on Image Processing (ICIP), pp. 2380-2383, 2008.

[48] Padhani, AR., Ollivie, L., The RECIST (Response Evaluation Criteria in Solid Tumors) criteria, implications for diagnostic radiologists. Br J Radiol 74, 983-986, 2001.

[49] Clarke, LP., Velthuizen, RP., Clark, M., et al., MRI measurement of brain tumor response: comparison of visual metric and automatic segmentation. Magn Reson Imaging 16, 71-279, 1998.

[50] Weiner, HL., Guttmann, CR., Khoury, SJ., et al, Serial magnetic resonance imaging in multiple sclerosis, correlation with attacks, disability, and disease stage. J Neuroimmunol 104,164-173, 2000.

[51] Jack, CR., Petersen, RC., Xu, Y, et al, Rate of medial temporal lobe atrophy in typical aging and Alzheimer's disease. Neurology 51:993-999, 1998.

[52] Haney, SM., Thompson, PM., Cloughesy, TF., et al, Mapping therapeutic response in a patient with malignant glioma. J Comput Assist Tomogr 25, 529-536, 2001.

[53] Haney, SM., Thompson, PM., Cloughesy, TF., et al, Tracking tumor growth rates in patients with malignant gliomas, a test of two algorithms. Am J Neuroradiol 22, 73- 82, 2001.

[54] Rey, D., Subsol, G., Delingette, H., et al, Automatic detection and segmentation of evolving processes in 3D medical images, Application to multiple sclerosis. Med Image Anal 6, 63-179, 2002.

[55] Gerig, G., Welti, D., Guttmann, CRG., et al, Exploring the Discrimination Power of the Time Domain for Segmentation and Characterization of Lesions in Serial MR Data Boston, MICCAI, 1998.

[56] Meier, DS., Guttmann, CR., Time-series analysis of MRI intensity patterns in multiple sclerosis. Neuroimage 20,1193-1209, 2003.

[57] Lladò, X., Ganiler, O., Oliver, A., Martì, R., Freixenet, J., Valls, L., Vilanova, J. C., Ramiò-Torrentà, L., Rovira, À., Automated detection of multiple sclerosis lesions in serial brain MRI, Neuroradiology 54, 787-807, 2012

[58] Candès, E., Tao, T., Near optimal signal recovery from random projections: Universal encoding strategies?, IEEE Trans. Inform. Theory, vol. 52, no. 12, pp. 5406-5425, Dec. 2006.

[59] Donoho, D., Compressed sensing, IEEE Trans. Inform. Theory, vol. 52, no. 4, Apr. 2006.

[60] Candès, E., Romberg, J., Tao, T., Stable signal recovery from incomplete and inaccurate measurements, Comm. Pure Appl. Math., vol. 59, no. 8, pp. 1207â“1223, Aug. 2006.

[61] Candès, E, Wakin, M. B., People Hearing Without Listening: An Introduction To Compressive Sampling. IEEE Signal Processing Magazine 25(2) 21-30, 2008.

[62] Bruckstein, A. M., Donoho, D. L., Elad, M., From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images, SIAM Review 2009 Society for Industrial and Applied Mathematics, Vol. 51, No. 1, pp. 34-81.

[63] Rubinstein, R., Bruckstein, A. M., Elad, M., Dictionaries for sparse representation modeling, IEEE Proceedings, vol. 98, no. 6, pp. 1045-1057, 2010.

[64] Curati, W.L., Williams, E.J., Oatridge, A., Hajnal, J.V., Saeed, N., Bydder, G.M., Use of subvoxel registration and subtraction to improve demonstration of contrast enhancement in MRI of the brain. Neuroradiology 38, 717-723, 1996.

[65] Shah, M., Xiao, Y., Subbanna, N., Francis, S., Arnold, D.L., Collins, D.L., Arbel, T., Evaluating intensity normalization on MRIs of human brain with multiple sclerosis, Med Image Anal 15 (2), 267-282, 2011.

[66] Tan, I.L., van Schijndel, R.A., Fazekas, F., Filippi, M., Freitag, P., Miller, D.H., Yousry, T.A., Pouwels, P.J.W., Adèr, H.J., Barkhof, F., Image registration and

subtraction to detect active T2 lesions in MS, an interobserver study, The Journal of Neurology 249(5), 767-773, 2002.

[67] Tan, I.L., van Schijndel, R.A., van Walderveen, M.A.A., Quist, M., Bos, R., Pouwels, P.J.W., Desmedt, P., Adèr, H.J., Barkhof, F., Magnetic resonance image registration in multiple sclerosis: comparison with repositioning error and observer-based variability, Journal of Magnetic Resonance Imaging 15(5):505-510, 2002.

[68] Lemieux, L., Wieshmann, U., Moran, N., Fish, D., Shorvon, S., The detection and significance of subtle changes in mixed-signal brain lesions by serial MRI scan matching and spatial normalization. Med Image Anal 2(3):227-242, 1998.

[69] Jenkinson,M., Smith, S., A global optimization method for robust affine registration of brain images, Medical Image Analysis 5(2):143-156, 2001.

[70] Jenkinson, M., Bannister, P.R., Brady, J.M., Smith, S., Improved optimization for the robust and accurate linear registration and motion correction of brain images, NeuroImage 17(2):825-841, 2002.

[71] Gerig, G., Martin, J., Kikinis, R., KÃ$\frac{1}{4}$bler, O., Shenton, M., Jolesz, F.A. Unsupervised segmentation of 3-D dual-echo MR head data. Image and Vision Computing 10:349-360, 1992.

[72] Gerig, G., Welti, D., Guttmann, C.R.G., Colchester, A.C.F., Szèkely, G., Exploring the discrimination power of the time domain for segmentation and characterization of active lesions in serial MR data. Med Image Anal 4:31-42, 2000.

[73] Meier, D.S., Guttmann, C.R.G., Time-series analysis of MRI intensity patterns in multiple sclerosis. NeuroImage 20:1193-1209, 2003.

[74] Srivastava, S., Maes, F., Vandermeulen, D., Paesschen, W.V., Dupont, P., Suetens, P., Feature-based statistical analysis of structural MR data for automatic detection of focal cortical dysplastic lesions. NeuroImage 27:253-266, 2005.

[75] Welti, D., Gerig, G., RadÃ$\frac{1}{4}$, E.W., Kappos, L., Szèkely, G., Spatiotemporal seg-

mentation of active multiple sclerosis lesions in serial MRI data, Int. Conf. Inform. Proc. Medical Imaging, pp 438-445, 2001.

[76] Thirion, J.P., Calmon, G., Deformation analysis to detect and quantify active lesions in three-dimensional medical image sequences. IEEE Trans Med Imaging 18:429-441, 1999.

[77] Pieperhoff, P., Sudmeyer, M., Homke, L., Zilles, K., Schnitzler, A., Amunts, K., Detection of structural changes of the human brain in longitudinally acquired MR images by deformation field morphometry: methodological analysis, validation and application. NeuroImage 43(2):269-287, 2008.

[78] Rey, D., Subsol, G., Delingette, H., Automatic detection and segmentation of evolving processes in 3D medical images: applications to multiple sclerosis. Med Image Anal 6, 163-179, 2002.

[79] Jolliffe, I., Principal component analysis. Springer. 2nd edition, .2002.

[80] De la Torre, F. and Black, M. J., A framework for robust subspace learning. International Journal of Computer Vision. Vol. 54, Issue 1-3, pp. 117-142, Aug.-Oct. 2003.

[81] Ding, C., Zhou, D., He, X., Zha, H., R1-PCA: Rotational invariant L1-norm principal component analysis for robust subspace factorization, in Proc. Int. Conf. Mach. Learn., pp. 281 - 288, Jun. 2006.

[82] Baccini, A., Besse, P., Falguerolles, A., âœA L1-norm PCA and a heuristic approach,â in Ordinal and Symbolic Data Analysis, E. Diday, Y. Lechevalier, and P. Opitz, Eds. Berlin, Germany: Springer, pp. 359 - 368, 1996.

[83] Ke, Q., Kanade, T., Robust L1-norm factorization in the presence of outliers and missing data. in Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2005.

[84] Candès, E. J., Li, X., Ma, Y., John Wright, J., 2011. Robust principal component analysis?. Journal of the ACM, vol 58, issue 3, Article 11, May 2011.

[85] Kwak, N., Principal component analysis based on L1-norm maximization. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 9, pp. 1672-1680, Sep. 2008.

[86] Gu, Q., Li, Z., Han, J., Joint Feature Selection and Subspace Learning. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence.

[87] Kwak, N., Principal Component Analysis by Lp-Norm Maximization. IEEE Transactions on Cybernetics, vol. 44, no. 5, May 2014.

[88] Nie, F., Huang, H., Ding, C., Luo, D., Wang, H., Robust Principal Component Analysis with Non-greedy L1-Norm Maximization. Proceedings. 22nd Int. Joint Conf. on Artificial Intelligence, p.1433-1438, 2011.

[89] Feng, D. C., Chen, F., Xu, W. L. Learning robust principal components from L1-norm maximization, Journal of Zhejiang University-Science (Computers Electronics), 13(12), 901-908, 2012.

[90] Ng, A., Feature selection, L1 vs. L2 regularization, and rotational invariance. Proceedings. International conference on Machine Learning, 2004.

[91] Tibshirani, R. Regression shrinkage and selection via the lasso. J. Royal. Statist. Soc B., 58, 267 - 288, 1996.

[92] Benasseni, J. Une am'elioration d'un r'esultat concernant l'influence d'une unit'e statistique sur les valeurs propres en analyse en composantes principales. Statistique et Analyse des Donn'ees, 11, 42 - 63, 1986.

[93] Chen, H., Robust principal component analysis with missing data and outliers, Technical report, Robust image understanding laboratory, May, 2002.

[94] Golub, G., Loan, C.V., Matrix Computation, Johns Hopkins University Press, 3 edition, 1996.

[95] Siegel, E. L., Re-thinking CAD for the next generation, SPIE 2014.

[96] Karssemeijer. N., Opportunities and challenges for diagnostic decision support systems, SPIE 2014.

[97] Zitova, B., Flusser, J., Image registration methods: a survey, Image and Vision Computing 21, 977-1000, 2003.

[98] Oliveira, P.M.F., Tavares, R.S.J., Medical image registration: a review, Computer Methods in Biomechanics and Biomedical Engineering, Vol. 17, No. 2, 73-93, 2014.

[99] Khalifa, F., Beache, M. G., Gimel'farb, G., Suri, S.j., El-Baz, S.A., State-of-the-Art Medical Image Registration Methodologies: A Survey, Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies, Springer, pp 235-280, 2011.

[100] Tang, L., Hamarneh, G., Medical Image Registration: A Review . Medical Imaging: Technology and Applications, pp 619-660, 2013.

[101] Thevenaz, P., Blu, T., Unser, M., Interpolation revisited. IEEE Trans. Med. Imaging 19, 739 - 758, 2000.

[102] Insight Segmentation and Registration Toolkit (ITK), `http://www.itk.org`.

[103] Boyd, S., Vandenberghe, L., Convex Optimization, ISBN: 9780521833783, Cambridge University Press, 2004.

[104] Liwicki, S., Tzimiropoulos, G., Zafeiriou, S., Pantic, M., International Journal of Computer Vision, Volume 101, Issue 3, pp 498-518, 2013.

[105] Bouwmans, T., Traditional and Recent Approaches in Background Modeling for Foreground Detection: An Overview, Computer Science Review, Volumes 11-12, pp 31-66, 2014.

[106] Olshausen, B.A., Field, D. J., Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature 381, pp 607-609, 1996.

[107] Olshausen, B.A., Field, D. J., Natural image statistics and ecient coding, Network Computation in Neural Systems, 7(2), pp 333â“339, 1996.

[108] Olshausen, B.A., Field, D. J., Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision Research, 37(23), pp 3311-3325, 1997.

[109] Rubinstein, R., Zibulevsky, M., Elad, M., Double Sparsity: Learning Sparse Dictionaries for Sparse Signal Approximation, IEEE Transactions on Signal Processing, vol. 58, no. 3, 2010.

[110] Candès, E. J., Wakin, M. B., Introduction to Compressive Sampling, IEEE Signal Processing Magazine, 21, 2008.

[111] Elad, M., Bruckstein, A.M., A Generalized Uncertainty Principle and Sparse Representation in Pairs of Bases, IEEE Transactions Information Theory, Vol. 48, no. 9, pp. 2558-2567, 2002.

[112] Donoho, D. L., Huo, X., âœUncertainty principles and ideal atomic decompositionâ IEEE Transaction Information Theory, vol. 47, pp. 2845â“2862, Nov. 2001.

[113] Candès, E. J., Romberg, J., Quantitative robust uncertainty principles and optimally sparse decompositions. Foundation of Computational Mathematics, 6, pp. 227-254, 2006.

[114] Mood, A., Graybill, F., Boes, D., Introduction to the Theory of Statistics (3rd ed.). McGraw-Hill. p. 229, 1974.

[115] DeGroot, M. H., Schervish, M. J., Probability and Statistics (2nd ed.). Addison-Wesley, 1980.