University of Alberta

Library Release Form

Name of Author: Meghna Singh

Title of Thesis: Theory and Methods for Efficient Spatio-Temporal Super-Resolution Imaging

Degree: Doctor of Philosophy

Year this Degree Granted: 2009

Meghna Singh
9107, 116 Street
Edmonton, AB
Canada, T6G 2V4

Date: _____

If I have seen further, it is by standing on the shoulders of giants.

– Issac Newton, Letter to Robert Hooke, February 5, 1675

University of Alberta

THEORY AND METHODS FOR EFFICIENT SPATIO-TEMPORAL SUPER-RESOLUTION IMAGING

by

Meghna Singh

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Department of Electrical and Computer Engineering

Edmonton, Alberta
Spring 2009

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Theory and Methods for Efficient Spatio-Temporal Super-Resolution Imaging** submitted by Meghna Singh in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

_____

Dr. Mrinal Mandal
Supervisor

_____

Dr. Anup Basu
Co-Supervisor

_____

Dr. Herbert Yang

_____

Dr. Nelson G. Durdle

_____

Dr. Dil Joseph

_____

Dr. Kostantinos N. Plataniotis
External Examiner

Date: _____

To my parents.

Abstract

The science of super-resolution imaging involves the construction of a single high-resolution representation by registering and fusing together multiple low-resolution representations. These candidate representations may include images and videos. This thesis covers four studies that advance the state-of-the-art in super-resolution imaging.

The first study is based on the innovative idea of computing Event Models to represent activities in the low resolution and low frame-rate input sequences. These Event Models are then used, instead of the sequences, for temporal registration. Experimental results show that the use of Event Models produces significantly better reconstruction than contemporary approaches.

Current techniques in super resolution imaging limit the input video sequences to be from the same scene or the same instance of an event. We propose a novel method for generation of high-resolution video from sequences acquired from repetitions of the same activity. The input video sequences may have different temporal scales and spatial viewpoints. Our proposed method uses Event Model techniques from the previous study and computes sequence synchronization to sub-frame accuracy. We also demonstrate the application of this new method by constructing a single 4D MRI sequence using multiple low resolution sequences.

MRI acquisition involves a fundamental trade-off between image quality and frame rate. In our review, we found that current MRI technology can only capture about seven frames per second before the image quality degrades below an acceptable threshold. The low frame rate limits the usefulness of the MRI in the study of high speed events, such as swallowing reflexes and cardiac motion. MRI can be used to capture multiple instances of the same event from multiple view points.

These sequences are temporally offset, non-uniformly scaled, from different view points and represent repetitions of the same event. Our third study builds on the techniques developed in the previous studies to register these orthogonal MRI sequences. This provides an ability to simultaneously view the orthogonal image planes from a single arbitrary view point.

In our final study we address the problem of identifying a subset of low resolution sequences, from a large set of samples, such that the resulting high resolution video has the best possible reconstruction accuracy. To compare multiple input video sequences, we first calculate an *a priori* confidence measure for each pair of video sequences. The confidence measure is derived from the registration error and non-uniformity of sequence samples. This confidence measure is a measure of diversity in the selected subset of low resolution sequences. We then use our iterative ranking algorithm to rank the low resolution sequences so that a minimum number of sequences result in desired reconstruction accuracy. Experimental results using a wide variety of input video and audio sequences show that the reconstruction accuracy of the proposed method is better than the reconstruction from a random selection of input sequences.

Acknowledgements

To be completed.

Table of Contents

1 Introduction **1**

1.1 Super-Resolution Imaging 2

1.2 Motivation 4

1.3 Problem Statement 6

1.3.1 SR Imaging at Low Frame Rates 6

1.3.2 SR Imaging from Related Scenes – Limited Variation in View Points 7

1.3.3 SR Imaging from Related Scenes – Orthogonal View Points 7

1.3.4 Choosing Sequence Combinations for SR Imaging 7

1.4 Thesis Contributions 8

1.5 Thesis Outline 9

2 Review **12**

2.1 Imaging Model and Low Resolution Image Generation 12

2.2 Categorization of Super-resolution Imaging Techniques 13

2.2.1 Image vs. Video SR 14

2.2.2 Spatial vs. Frequency Domain SR Techniques 14

2.3 Image SR – Frequency domain Techniques 15

2.4 Image SR – Spatial Domain Techniques 17

2.4.1 Registration and Interpolation of Non-uniformly Spaced Samples 17

2.4.2 Iterated Back-Projection 19

2.4.3 Projection onto Convex Sets 20

2.5 Video SR – Spatio-temporal Superresolution 22

2.5.1 Temporal Effects in Video SR . . . 23
2.5.2 Video SR Categorization . . . 23
2.5.3 Space-time SR . . . 25
2.5.4 Epitomes . . . 29
2.5.5 Frame Rate Up-conversion and Temporal Interpolation . . . 29
2.6 Video Synchronization . . . 30
2.6.1 External Gating Signal based Synchronization . . . 30
2.6.2 Image based Synchronization . . . 31
2.6.3 Video Synchronization of Same Scene . . . 33
2.6.4 Video Synchronization of Related Scenes . . . 35
2.7 Spatial Registration . . . 35
2.7.1 Feature Extraction . . . 36
2.7.2 Homography . . . 38
2.7.3 Random Sampling and Consensus . . . 39
2.8 Motion Tracking . . . 39
2.8.1 Adaptive Background Subtraction . . . 39
2.8.2 Template Matching . . . 40
2.9 Motion Modeling . . . 40
2.9.1 Parametric Motion Models . . . 40
2.9.2 Non-parametric Motion Models . . . 41
2.10 Dynamic Time Warping . . . 42
2.11 Linear Regression . . . 44
2.12 Recurrent Non-uniform Sampling . . . 46
2.12.1 Sampling formulation . . . 47
2.12.2 Reconstruction formulation . . . 47
2.13 Summary . . . 49

3 Event Models as a Tool for SR Imaging at Low Frame Rates 50
3.1 Introduction to Event Models . . . 51
3.2 System Overview . . . 53
3.2.1 Regression Model for Temporal Alignment . . . 54

3.2.2 Minimization of Alignment Error 57
3.2.3 Caspi Model for Temporal Alignment 59
3.2.4 Comparative Analysis 60
3.3 Experimental Analysis 61
3.3.1 Synthetic Data 61
3.3.2 Noise Analysis 62
3.3.3 Real Data 64
3.4 Applications 66
3.4.1 Medical Visualization 69
3.4.2 High Temporal Resolution Video Generation 74
3.5 Event Dynamics Technique and Sampling Theorem 74
3.6 Conclusions 77

4 Symmetric Transfer Error for Spatio-Temporal SR Imaging 79
4.1 Review of Synchronization Techniques 80
4.2 Proposed Approach 82
4.2.1 Spatial Alignment 84
4.2.2 Event Models 85
4.2.3 Compute Ghosts 85
4.2.4 Regularized Dynamic Time Warping and Symmetric Transfer Error 86
4.2.5 Algorithm Pseudocode 89
4.3 Experiments and Comparative Analysis 91
4.3.1 Synthetic Sequences 92
4.3.2 Real Sequences 97
4.3.3 Symmetric vs. Asymmetric Synchronization 100
4.4 Application – 4D MRI Registration 101
4.5 Summary 104

5 Spatio-Temporal SR Imaging from Orthogonal Viewpoints 108
5.1 Review of Dynamic MRI Registration 108
5.2 Proposed Method 111

5.2.1 MRI Acquisition . . . 112
5.2.2 Registration to Fiducial Volume . . . 113
5.2.3 Computing Intensity Profiles . . . 113
5.2.4 Matching Maxima in Intensity Profiles . . . 116
5.3 Results . . . 118
5.3.1 Verification . . . 122
5.4 Summary . . . 123

6 A Confidence Measure to Choose Low Resolution Sequences in SR Imaging 124
6.1 Introduction to the Problem . . . 125
6.2 Pre-processing . . . 127
6.3 Proposed Method . . . 128
6.3.1 Factors Affecting Sample Confidence . . . 129
6.3.2 Computing the Confidence Measure . . . 133
6.3.3 Iterative Rank-based Method . . . 139
6.4 Performance Evaluation of the Proposed Method . . . 142
6.4.1 Independent Evaluation of Φ_g and Φ_l . . . 142
6.4.2 Evaluation of Confidence Measure . . . 143
6.4.3 Evaluation of IRBR Method . . . 146
6.4.4 Evaluation with MR Imaging . . . 147
6.5 Summary . . . 155

7 Summary and Conclusions 158
7.1 Contributions . . . 158
7.2 Publications . . . 161
7.3 Future Research Direction . . . 163

Bibliography 165

A Weighted Least Square Regression 175

B Video DataBases **177**

B.1 Database 1 177

B.1.1 DB-1 Synthetic Sequences 178

B.1.2 DB-1 Real Sequences 180

B.2 Database 2 183

B.2.1 DB-2 Synthetic Sequences 183

B.2.2 DB-2 Real Sequences 185

B.3 Database-3 MRI video sequences 188

B.3.1 Acquisition 188

List of Tables

2.1 Review of related work. Legend: D.S.-Dynamic time shift, V.I.-View Invariant, S.A.-Sub-frame Accuracy 34

3.1 Results of Temporal Alignment (TA) with Synthetic Data for Linear Interpolation (LI) and Event Dynamics (ED) based matching. LI corresponds to Caspi Algorithm and ED corresponds to Proposed Algorithm. Unit of error is 'frames'. 62

3.2 Results of Temporal Alignment (TA) with Real Data for Linear Interpolation (LI) and Event Dynamics (ED) based matching. LI corresponds to Caspi Algorithm and ED corresponds to Proposed Algorithm. Unit of error is 'frames'. 68

4.1 Synchronization Errors for RCB and STE methods for Noisy Trajectories. Unit of measurement is the sum of absolute differences between the actual and computed frame correspondence. 93

5.1 Pseudocode for computing sROI and Sframe 117

5.2 Pseudocode for computing Cframe 119

6.1 Pseudocode for computing optimal weights. 139

6.2 Reconstruction error values for optimized and sub-optimal weights w_g and $w_l = 1 - w_g$. 139

6.3 Experimental results for independent evaluation of objective functions Φ_g and Φ_l. 143

6.4 Confidence measure χ and corresponding reconstruction error for synthetic sample sets. 145

6.5 Confidence measure χ and corresponding reconstruction error for real video sequences. 146
6.6 SNR values for 4 ROIs in LR and SR video sequences and Confidence measures. 150

B.1 DB-1 List of Real Sequences. 179
B.2 DB-2 MATLAB code to generate synthetic trajectories. 184

List of Figures

2.1 General imaging model of the generation of low resolution images. (Trans.=Transformation) 13
2.2 Super-resolution categories based on problem domain - Image SR versus Video SR. 14
2.3 Classification of SR approaches into frequency domain and spatial domain. ML-Maximum Likelihood. 15
2.4 Irani-Peleg Iterated Back Projection Algorithm for image SR (adapted from [28]). MSE- Mean Square Error. 18
2.5 Temporal aliasing (adapted from [55]). (a) Trajectory of a ball over time. (b) Trajectory sampled over time by a low frame rate camera. Perceived trajectory is along a straight line. (c) Illustration that even with perfect temporal interpolation between the sub-sampled frames of (b) the true motion trajectory cannot be recovered. 23
2.6 Illustration of Motion blur and Spatial blur caused due to exposure time and the point spread function of the camera respectively (adapted from [55]). The rectangle indicated by spatial blur illustrates the spatial area that will be averaged to a single pixel value in the discretized image. The temporal blur rectangle indicates the time-period over which temporal information will be averaged to a single frame in the video sequences. 24
2.7 Different categories of spatio-temporal SR techniques. 25
2.8 Relation between various transformations of the Shetchman model[55]. 26

2.9 Illustration of dynamic time warping of two 1D signals A and B. (a) The two signals which have the same overall shape, but vary in the temporal duration of some sections of the signal. (b) The alignment computed by DTW. 43
2.10 Dynamic Time Warping of two signals A and B. (Most algorithms assume a linear warp.) 44
2.11 Illustration of recurrent non-uniform sampling with two sample sets. 47
2.12 (a) Reconstruction from uniform samples using sinc kernels. (b) Illustration of non-uniform samples which can be expressed as linear combinations of samples from (a). 48

3.1 Systems overview of proposed technique for temporal alignment of videos of the same scene 54
3.2 Decomposition of motion patterns into constituent x and y motion. (a) Spiral motion is decomposed into constituent x and y oscillatory patterns; (b) ball-throw motion is decomposed into linear motion along the horizontal coordinate and oscillatory motion along the vertical coordinate. 58
3.3 Pictorial representation of linear interpolation in Eq. 3.9 for sub-frame temporal alignment. 60
3.4 Case study of failure of linear interpolation for temporal alignment. 61
3.5 (a)-(h) Sample trajectories from synthetic data. (a) Traj 1, (b) Traj 2, (c) Traj 3, (d) Traj 4, (e) Traj 5, (f) Traj 6, (g) Traj 7, and (h) Traj 8. 63
3.6 Plot of Error in temporal registration vs. variance of noise added to the 50 real data trajectories for Linear Interpolation (Linear Interp) and event dynamics based matching (ED Matching). 64
3.7 (a)-(f) Sample trajectories from the real data. (a) Throw 1, (b) Swing 1, (c) Swing 2, (d) Swing 3, (e) Throw 2 and (f) Throw 3. 65

3.8 (a) Original frame at correct temporal alignment. (b) Incorrectly aligned frame corresponding to (a) and (c) Superimposed frames from the original sequence and temporally aligned sequence showing incorrect registration. 67
3.9 Overview of application of method to medical visualization. 70
3.10 Background separated images from a swallow showing the bolus. . . 71
3.11 Bolus track in space-time volume of (a) swallow1 and (b) swallow2. Centroid path in space-time volume of (c) swallow1 and (d) swallow2. 71
3.12 Two MRI datasets of swallowing aligned using offset determined by ED based matching. Legend: d <sequence number>-f <frame number>. Based on centroidal paths of the bolus of water, frames from sequence 2 were placed approximately midway between frames from sequence 1, but offset from the beginning of sequence 1 by 4 frames. 73
3.13 (a)-(e) Frame from the original sequence of ball swinging, (f)-(j) Temporal registration based on linear interpolation, (k)-(o) Temporal registration based on event dynamics. (e) is the frame in the original sequence that is to be aligned. (g) is the incorrectly aligned frame. (o) is the correct alignment of (e). 75
3.14 Trajectory of the centroid x-coordinate over 58 frames in a video taken at 30fps. 76
3.15 Power spectrum of the trajectory in Fig. 3.14 showing that bandwidth can be approximated to be below 5Hz 77

4.1 System Overview of Symmetric Transfer Error (STE) Approach. DTW – Dynamic Time Warping. 82
4.2 Illustration of two distinct scenes acquired using two distinct cameras. The projections (ghosts) of scenes onto the reciprocal cameras are also shown. 83
4.3 Computing symmetric transfer error for a single frame ‘i’ in Sequence 1. 87

4.4 Illustration of computing the mapping function $\mathcal{W}$ from the cost matrix $\mathcal{D}$. 89
4.5 (Plot of STE $\mathcal{E}(w)$ versus regularization weight w for two synthetic sequences of length 100 and 140 frames. $min(\mathcal{E})$ is also indicated. . 90
4.6 Results of synchronization of synthetic trajectories using proposed Symmetric Transfer Error (STE) approach and rank-constraint based (RCB) approach. (a)-(b) Simple trajectories result in comparable synchronization between both approaches. 94
4.7 Results of synchronization of synthetic trajectories using proposed Symmetric Transfer Error (STE) approach and rank-constraint based (RCB) approach. (a)-(b) More complex trajectories demonstrate the efficacy of the symmetric minimization approach. 95
4.8 Two illustrative noisy trajectories with noise variance =0.1. 96
4.9 Performance of the STE and RCB methods for noisy trajectories with noise variance (a) 0.1 and (b) 0.01. 96
4.10 Results of synchronization using a rank-constraint based RCB method. The object being tracked is enclosed in a green rectangle. (a)-(d) Frames from Sequence 1, (e)-(h) Corresponding synchronized frames from Sequence 2. 98
4.11 Results of synchronization using Proposed method. The object being tracked is enclosed in a green rectangle. (a)-(d) Frames from Sequence 1, (e)-(h) Corresponding synchronized frames from Sequence 2. 99
4.12 Warping computed between Sequence 1 and Sequence 2 of realUofA.avi test files. Point(1)-(3) are singularities marked on the warp. 100
4.13 Symmetric vs. Asymmetric Synchronization of realUCF video files. 101
4.14 Illustration of spatial alignment of MRI slices. 102
4.15 Synchronization computed between Center-Right MRI sequences. Frame 7 of the right MRI sequence is mapped to frame 6.5 of the center MRI sequence. 103

4.16 Synchronization computed between Center-Left MRI sequences by proposed algorithm. Frame 5 of the left MRI sequence is mapped to frame 5.5 of the center MRI sequence. 105
4.17 Synchronization computed between Center-Left MRI sequences by RCB algorithm. RCB algorithm is unidirectional and limited to integer frame alignment. 106
4.18 Static visualization (one time instant) of 4D synchronized MRI data. The colormap of the 4D model is set to show regions of high intensity as red and low intensity as blue. The bolus can be easily seen as the red section close to the nasal region. 107

5.1 Illustration of MRI data acquisition planes. 110
5.2 System overview of Bi-directional, Orthogonal Dynamic MRI Registration. 111
5.3 3D path traced by the moving bolus and corresponding regions in the bidirectional planes that need to be identified. 112
5.4 Intensity profile computed for frame-26 of the center sagittal sequence. The sagittal region of interest (sROI) is also indicated. . . . 114
5.5 Intensity profile computed for frame-1 of the center sagittal sequence. 114
5.6 Dynamic Intensity Profiles over sagittal regions of interest (sROI). The peak in each profile indicates the frame at which the maximum bolus passes through that region. 116
5.7 sROI is projected onto the fiducial volume and then further projected onto the coronal imaging plane. 118
5.8 Dynamic Intensity Profiles over coronal regions of interest (cROI). The peak in each profile indicates the frame at which the maximum bolus passes through that region. 119
5.9 Frame number correspondence computed between sagittal and coronal sequences. 119

5.10 A section of the MRI sequences indicating the alignment result. Corresponding frames in the coronal and sagittal planes when (a) the soft palate has been pushed up and the bolus is ready to descend into the pharynx, (b) the epiglottis has descended and the leading edge of the bolus has reached the epiglottis, (c)-(d) continued in Fig. 5.11. 120
5.11 A section of the MRI sequences indicating the alignment result. Corresponding frames in the coronal and sagittal planes when (c) the bolus splits over the epiglottis and (d) the bolus begins its descend into the oesophagus. 121
5.12 Verification of results using SSD metric. The minimum error at offset=0 indicates the accuracy of our method. 122

6.1 Simplified flowcharts of (a) standard SR reconstruction process, (b) enhanced SR reconstruction process based on computed confidence measure and iterative greedy ranking algorithm. 126
6.2 Illustration of decrease in reconstruction error with increase in τ_n (reported as a normalized number $[0, 1]$, where 1 corresponds to the sampling rate T). 129
6.3 Modeling the error in temporal registration as a (a) Gaussian distribution, (b) Uniform distribution. 131
6.4 Effects of error in temporal registration on reconstruction. 133
6.5 Illustration of recurrent non-uniform sampling with two sample sets. 133
6.6 (a) Relationship between reconstruction error and objective function Φ_g. (b) Relationship between reconstruction error and objective function Φ_l. 136
6.7 Confidence measure values computed for two different set of weights (w_g1 and w_g2). Linear fit for w_g2 results in a smaller residual. . . . 138
6.8 Flowchart of IRBR method based on the proposed confidence measure. FR* indicates a RNUS reconstruction algorithm from [45] which was reviewed in Chapter 2. 141

6.9 (a) Sample frames from real data sequence, (b) Sample trajectory from real data sequence. 144
6.10 Performance of iterative rank based reconstruction (IRBR) algorithm with synthetic data. 147
6.11 (a) Original 'toilet.wav' audio signal, (b)-(c) Two representative sections of the original audio signal that were used in the experiments. 148
6.12 Performance of iterative rank based reconstruction (IRBR) algorithm with audio data. 149
6.13 (a) Illustrative frame from LR video2; (b) Closest corresponding frame in LR video3; (c) Intermediate frame reconstructed using (a) and (b). 149
6.14 (a) Illustration of two radial projection lines. A data point in between the projection lines is re-gridded by convolving with a symmetric Kaiser kernel. (b) A 1D Kaiser window, $\beta = 3$. 151
6.15 ROIs used to compute SNR values in Table 6.6. 152
6.16 Representative frames of SR MRI videos. (a) vid1-vid2, $\chi = 27.64$, zoomed position of the tongue shows incorrect registration, (b) vid2-vid3, $\chi = 38.7$, zoomed position of tongue shows correct registration. 154
6.17 Representative frames of SR MRI videos. (a) vid1-vid2, (b) vid2-vid3 and (c) vid1-vid3. The position of the epiglottis has been highlighted with arrows. 156
6.18 Zoomed in sections of SR MRI frames shown in Fig. 6.17. (a) vid1-vid2, (b) vid2-vid3 and (c) vid1-vid3. The position of the epiglottis has been highlighted with arrows. 157

B.1 DB-1 Synthetic 3D trajectory generated by using the MATLAB commands displayed in Section B.1.1 178
B.2 DB-1 (a)-(h) Sample trajectories from synthetic data. (a) Traj 1, (b) Traj 2, (c) Traj 3, (d) Traj 4, (e) Traj 5, (f) Traj 6, (g) Traj 7, and (h) Traj 8. 179

B.3 DB-1 Illustrative Video Frames from Swing 3 video. Sequences should be read row-wise from top-left corner, every fifth video frame is shown. 180
B.4 DB-1 Illustrative Video Frames from Throw 1 video. Sequences should be read row-wise from top-left corner, every alternate video frame is shown. 181
B.5 DB-1 (a)-(f) Sample trajectories from the real data. (a) Throw 1, (b) Swing 1, (c) Swing 2, (d) Swing 3, (e) Throw 2 and (f) Throw 3. 182
B.6 DB-2 Representative noisy synthetic trajectories with noise variance $\sigma^2 =$ (a) 0.0001, (b) 0.001, (c) 0.01, (d) 0.1. 183
B.7 DB-2 Illustrative Frames from UCF test video. Sequence should be read row-wise from Top-left corner. 186
B.8 DB-2 Illustrative Frames from UofA test video. Sequence should be read row-wise from Top-left corner. 187
B.9 DB-3 (a) Illustrative image showing major anatomical parts associated with swallowing. (b) MRI image depicting major anatomical parts. Legend: (1)– Soft Palate, (2)–Bolus, (3)–Epiglottis, (4)–Tongue, (5)– Trachea (or wind pipe), (6)–Naso-Pharynx (or nasal passage), (7)–Oesophagus, (8)– Stomach. 189
B.10 DB-3 Illustration of spatial alignment of MRI slices. 189
B.11 DB-3 MRI frames for Center Sequence. Sequence should be read from Left to Right corner. 190
B.12 DB-3 Fig. B.11 continued. 191
B.13 DB-3 MRI frames for Right Sequence. Sequence should be read from Left to Right corner. 191
B.14 DB-3 Fig. B.13 continued. 192
B.15 DB-3 MRI frames for Left Sequence. Sequence should be read from Left to Right corner. 192
B.16 DB-3 An illustrative Coronal MRI image with some anatomical regions marked. 193

B.17 DB-3 Coronal MRI frames showing the bolus approach the oro-pharyngeal cavity and splitting over the epiglottis. 194

Abbreviations

CFT	Continuous Fourier Transform
CPM	Continuous Profile Model
cROI	Coronal Region of Interest
DFT	Discrete Fourier Transform
DLT	Direct Linear Transform
DoG	Difference of Gaussian
DS	Dynamic-Time Shift
DTW	Dynamic Time Warping
ECG	Electrocardiogram
ED	Event Dynamics
EM	Event Model
fps	Frames per second
FT	Fourier Transform
FFT	Fast Fourier Transform
HDTV	High Definition Television
HMM	Hidden Markov Model
HR	High Resolution
Hz	Hertz
IBR	Intensity Based Region
IRBR	Iterative Rank Based Reconstruction
LI	Linear Interpolation
LR	Low Resolution
MAP	Maximum a-Posteriori
ML	Maximum Likelihood
MRI	Magnetic Resonance Imaging
MSE	Mean Square Error
MSER	Maximally Stable Extremal Regions
NMR	Nuclear Magnetic Resonance
NTSC	National Television System(s) Committee
PAL	Phase Alternating Line
PET	Positron Emission Tomography
POCS	Projection Onto Convex Sets

RAM	Random Access Memory
RANSAC	Random Sampling and Consensus
RCB	Rank Constraint Based
RMSE	Root Mean Square Error
RNUS	Recurrent Non-Uniform Sampling
ROI	Region of Interest
SA	Sub-frame Accurate
SIFT	Scale Invariant Feature Transform
SNR	Signal to Noise Ratio
SR	Super-Resolution
sROI	Sagittal Region of Interest
SSD	Sum of Squared Differences
SSE	Sum of Squared Error
SST	Total Sum of Squares
STE	Symmetric Transfer Error
ST-SR	Spatio-Temporal Super Resolution
SVD	Singular Value Decomposition
VI	View Invariant
WBLR	Weighted Bi-Square Linear Regression

Chapter 1

Introduction

Mankind has always endeavored to capture and record events occurring in our surrounding. These records have taken varied forms, ranging from cave paintings and manuscripts, to images and videos. The earliest imaging system was developed over a thousand years ago by Ibn al-Haytham ($\sim$ circa 1021) as a crude pinhole projection instrument. Over the next eight hundred years imaging systems developed into increasingly sophisticated instruments, with the first permanent photograph being made in 1826-27 by Joseph Nicéphore Niépce and the first color photograph being made by James Clerk Maxwell in 1861 [82]. Close to a hundred and fifty years later, highly sophisticated imaging systems have been sent on excursions to Mars and novel modalities, such as thermal imaging and magnetic resonance imaging, have been developed to acquire and process visual information in entirely new domains.

While recent advances in imaging systems have lead to significant improvements in the spatial and temporal resolution of cameras, the prohibitive cost of such systems still limits their use. Super-Resolution (SR) is a term generally applied to image processing algorithms that supplant the need for expensive imaging equipment by fusing low resolution (LR) images or sequences of images to generate a high resolution (HR) image or video sequence. The two basic premises of SR imaging algorithms are:

1. *Variability*: As SR imaging fuses information from multiple LR images, it is important that the information supplied by each image be different. In the case of images, this premise translates into small spatial (sub-pixel) differences in the multiple acquired images. In the case of video sequences, differences in temporal acquisition are also added to this premise.

2. *Registration*: While variability ensures that the spatio-temporal information of the images are different, often these differences are so vast that the images need to be *registered* to a common frame of reference. If the registration computed between images (or sequences) is inaccurate, reconstruction results are poor.

This thesis is a collection of projects that attempt to improve the performance and efficiency of spatio-temporal SR imaging. In pursuit of this goal we have conducted four studies that provide new theory and methods to further advance the field of SR imaging.

In this chapter, we will briefly review the chronological progression of imaging systems and the development of SR imaging. We will then discuss the limitations of current systems to provide the motivation behind this work, followed by the objectives and organization of this thesis.

1.1 Super-Resolution Imaging

From as early as 1984, researchers have been working on fusing information from multiple digital images to enhance the spatial details in images. Some of these techniques are based on the assumption that high frequency details are available in a LR image as aliased frequencies; and algorithms that estimate and filter the aliased information were developed. Other techniques assume that the image acquisition system is diffraction limited, for example, the camera lens itself is filtering out

high frequency signals and no aliasing occurs. Yet another set of techniques rely on extrapolation of available information by assuming that image information is analytic in nature and can be modeled as a mathematical function. A review of various techniques of SR imaging is presented in Chapter 2. Irrespective of the assumptions that these techniques make, the common goal of SR imaging remains – to combine the information in multiple low resolution images to generate an image that has higher resolution.

It is important to understand that a single image can never (not with current technology) achieve the same level of detail as a real world scene. This is because the spatial resolution of a camera is limited by the number of detectors on the camera and the blur induced by the detector technology itself. The temporal resolution on the other hand is limited by the exposure time and the frame rate of the camera. These factors limit the spatial detail that can be captured as well as the maximum speed of the dynamic events that can be acquired. SR traditionally has been the reconstruction of *static* or *still* HR images from a set of LR images captured at sub-pixel differences from each other [74][77]. The central idea behind SR reconstruction is that at sub-pixel displacements, each LR image acquires slightly different information about the scene, and if the blurring and sampling induced by the camera can be modeled and reversed, then these multiple LR images can be fused together to generate a HR image. Lately the domain of SR has widened to incorporate videos sequences as well, where a video sequence with high spatio-temporal resolution is created by fusing a set of low (spatio-temporal) resolution (LR) videos or combination of such LR videos and HR images [55][71].

SR reconstruction is focal to many applications such as generating HR still images from video [49] (where it is desirable to enlarge a single frame with more detail), converting NTSC/PAL video to a high definition video (HDTV), artificial zoom (where a section of the video stream is enlarged with more detail), in remote

sensing and astronomy [65] (where images are often blurred) and medical imaging [23][33] (such as positron emission tomography - PET scans where SR algorithms have been used to increase the spatial resolution of the scan by combining multiple scans offset with spatial shifts).

1.2 Motivation

The major motivation behind this research work comes from the fact that although a substantive amount of work has been conducted in SR imaging of still images, corresponding work in SR imaging of video sequences is still in its infancy.

While some ideas and concepts from image-SR translate well into video-SR, the temporal aspect of video sequences adds some unique challenges. These unique challenges need to be addressed before video-SR gains the same popularity that image-SR enjoys. In the author's opinion there are several areas that have not undergone sufficient investigation in current studies on Spatio-Temporal (ST) Super-resolution imaging, and some of these areas are highlighted below:

1. Contemporary SR imaging algorithms only support relatively high frame rate (30 fps) video sequences, and traditionally the sequences used to test these algorithms have very simple motion such as a large ball bouncing on the floor or a car driving into a parking lot. When compared to the frame rates of the video cameras the motion in the sequences is at much lower frequencies. In such cases, sub-frame registration between the sequences is achieved by linearly interpolating the motion between frames. For simple motion, acquired at high frame rates, linear interpolation does not deviate significantly from the underlying motion. While linear interpolation works for high frame rates, it results in erroneous ST registration for low frame rate sequences. Thus, for low frame rate video sequences, or sequences whose acquisition frame rate is lower than the motion in the sequence, an alternate approach is needed.

2. Another limitation of current studies on Spatio-Temporal SR imaging is that the scope of the problem being addressed is limited to multiple acquisitions of a single scene. Other assumptions such as planar motion, limited view point variation and fast frame rates also restrict the scope of the problem. In some real life events it is not possible to acquire multiple video sequences over the same duration of time. It is possible however, to repeat the event and acquire multiple sequences corresponding to each repetition. In some cases, the repetition of an event can result in subtle spatial and temporal changes since the repetition need not be identical in every respect. A generalized SR algorithm should be able to deal with these changes. To the best knowledge of the author, there has been no prior work that deals with Spatio-Temporal SR imaging from event repetitions.

3. Limited view point variation is a significant issue in SR imaging because it forces the low resolution images to have a large degree of overlap in the scene that is acquired. This implies that images acquired from view points that have large angular differences, cannot be registered and reconstructed as a high resolution image. The reason view point is a significant issue is because the feature points in images often change when viewed from different angles and different distances, thus making it difficult to relate feature points in one image to another. The larger the difference in viewpoint and the more complex the object or scene being viewed, the more difficult it is to relate image features to each other. The same issue translates from static images to video sequences acquired from varying view points, and consequently there is a need to register video sequences that have considerable variation in view points.

4. SR imaging community has also not addressed the issue that when multiple

input sequences are available, we need to choose which combination of sequences will result in the best reconstruction. This is a crucial choice, since poorly registered sequences will result in poor reconstruction. Currently, there are two ways that the community addresses multiplicity of sequences – (i) they use all the sequences that are available or (ii) they determine the best combination of sequences by subjectively viewing reconstructed SR videos from all possible input combinations. The first approach can often lead to ghosting errors that indicate poor SR reconstruction and the second approach is computationally very expensive since all input combinations are reconstructed. Hence, there is clearly a need for a method to choose optimal input sequences.

The justifications above highlight the necessity for further research in Spatio-Temporal SR imaging and in developing methods to improve the efficiency of SR reconstruction. In this thesis we will attempt to address the four problems discussed above.

1.3 Problem Statement

In this thesis we investigate: (i) SR imaging schemes for low frame rate videos, (ii) SR imaging schemes for video sequences of related scenes (event repetitions with limited variation in view points), (iii) SR imaging schemes for video sequences with orthogonal view points and (iv) methods to choose sequence combinations for SR imaging.

1.3.1 SR Imaging at Low Frame Rates

In our first study we look at a simplified problem that the video sequences are of the same scene, same viewpoint and are acquired at a fixed (unknown) temporal offset. These restrictive assumptions are removed in the later studies. We investi-

gate the effect of low frame rates on current state-of-the-art SR imaging algorithms. Our work has shown that linear interpolation approach for sub-frame registration of video sequences is not very accurate when frame rates of the video sequences are low.

1.3.2 SR Imaging from Related Scenes – Limited Variation in View Points

In our second study we investigate a new paradigm of combining multiple video sequences from repeated instances of activities or related scenes. This is a challenging problem since the temporal scale and spatial viewpoint of the sequences can be different. We examine a state-of-the-art algorithm for synchronization of related video sequences and show that the algorithm performance drops with increase in complexity of motion in the sequences. Also, this contemporary algorithm cannot compute sub-frame synchronization. Sub-frame synchronization is critical in order to increase the temporal resolution in Spatio-Temporal SR imaging.

1.3.3 SR Imaging from Related Scenes – Orthogonal View Points

In our third study we investigate the effect of orthogonal viewpoints in Spatio-temporal SR imaging, with particular emphasis on dynamic MRI sequences. Orthogonal viewpoints are addressed as a separate study since the algorithms for viewpoint reconciliation fail when the acquisition is orthogonal (infact in most practical cases, a viewpoint change greater than 30 degrees is irreconcilable). Another interesting problem associated with this study is that the MRI frames are not static, rather there is significant anatomical movement in the sequences.

1.3.4 Choosing Sequence Combinations for SR Imaging

Improving the efficiency of SR imaging is an integral part of this thesis work. In our fourth study we investigate the factors that can help determine which sequence com-

binations should be used for SR reconstruction. The sequences individually have a fixed frame rate, however they are non-uniformly distributed in time (with respect to each other). Our studies reveal that the order in which the sequences are fused as well as the accuracy with which they are registered also affect the reconstruction result. It is therefore important to investigate factors such as non-uniformity, registration accuracy and ranking of the sequences to improve SR imaging.

1.4 Thesis Contributions

The primary contribution of this thesis is the design of novel methods for efficient spatio-temporal super-resolution imaging. The major contributions are :

1. The development of event models [57][64][58] for sub-frame registration of sequences for SR imaging (presented in Section 3.2).

2. An algorithm [60] to align video sequences of related scenes that is invariant to changes in viewpoint and temporal scale (presented in Section 4.2).

3. An algorithm [59] to align video sequences acquired from orthogonal viewpoints (presented in Section 5.2).

4. A confidence measure and iterative ranking method [61] [62] to choose LR sequences that result in the best SR reconstruction (presented in Section 6.3).

Our first study advances the state-of-the-art in sequence alignment by developing a method to compute the dynamics of events from video sequences. A variational framework is developed that builds on the event models to compute the temporal registration or alignment between sequences. The dynamics of events (or event models) offer significant performance improvement in sub-frame sequence alignment over existing local alignment methods. Our second study involves the development of a novel method to register video sequences of related events that

have varying temporal scale. Our proposed method allows us to combine data from sequences of multiple repetitions of activities which is particulary useful when the number of acquisition sources is limited, either due to prohibitive costs or technological limitations. Our third study deals with a method to register and reconstruct video data that has been acquired in orthogonal planes. This study is a significant development in being able to generate and visualize medical data in four dimensions (3D+time). Our fourth study involves an in-depth analysis of non-uniform sampling, in particular recurrent non-uniform sampling (RNUS) and reconstruction. Concepts learned from RNUS are then applied to construct a confidence measure that allows us to compare and choose between LR sequences so that only those LR sequences which have been accurately registered and which contribute useful information to the reconstruction process are used for SR reconstruction; and LR sequences that will deteriorate the reconstruction process are discarded.

1.5 Thesis Outline

This thesis contains materials previously published at various international conferences and journals. In **Chapter 2**, we review the development of SR imaging, state-of-the-art SR imaging techniques and discuss their limitations. We also review registration, video synchronization and sequence alignment algorithms that can be (or have been) used as precursors to reconstruction in SR imaging. Some fundamentals of dynamic time warping, linear regression, recurrent non-uniform sampling, computer vision and image processing techniques that are used in this thesis work are also briefly discussed in Chapter 2.

The main body of work done in this PhD thesis is presented in four chapters from Chapter 3 - Chapter 6. In **Chapter 3** we address a simplified problem of registering low frame rate sequences of the same scene. We present the formulation of event models and a variational framework to compute sub-frame accurate sequence

registration. In this chapter, we compare and contrast event model based registration to another commonly used state of the art method for sequence alignment. We demonstrate the efficacy of our method with experiments on real and synthetic data sequences.

In **Chapter 4** we present an algorithm to register LR sequences that are acquired over varying temporal scales, for example, two repetitions of the same or similar activity. This extends the problem scope from Chapter 3 to varied view points and varied temporal duration. In this second study we develop a symmetric transfer error (STE) which is minimized in a dynamic programming framework. The minimization of a symmetric error results in synchronization that is not biased by the choice of reference sequence. The regularized nature of the STE significantly reduces the occurrence of singularities and results in sub-frame synchronization. Comparative analysis with a rank-constraint based method is also presented in Chapter 4 to demonstrate the marked improvement in video synchronization with our method.

In **Chapter 5** we extend the problem scope to address SR imaging from sequences acquired from orthogonal viewpoints. We present a method to register 2D magnetic resonance imaging sequences acquired in bidirectional orthogonal planes. This method allows the visualization of dynamic events in 4D space.

During the course of our first study on registration and reconstruction based on event models, we realized that not all the low resolution sequences were contributing equal information to the reconstruction process. Some combinations of LR sequences resulted in superior reconstruction while others resulted in unacceptable reconstruction results. This motivated our fourth study on developing a metric that could help choose sequences for efficient reconstruction. In **Chapter 6** we introduce a novel concept in SR imaging of automatically selecting only those LR sequences that have a positive impact on reconstruction. This selection of sequences

is achieved via a confidence measure, which is formulated in Chapter 6. This automatic pre-screening of LR sequences has two significant aspects: (i) It allows us to use a smaller number of LR sequences to achieve a good SR reconstruction and (ii) It weeds out sequences that would deteriorate SR reconstruction. The underlying concepts behind the confidence measure and various factors that affect it are also presented. We also develop an iterative greedy ranking algorithm that uses the confidence measure to efficiently reconstruct higher resolution data.

Chapter 7 presents the conclusions of this thesis and future research directions.

Chapter 2

Review

In this chapter we present a literature review of various methods and algorithms that are relevant to this thesis. The review is divided into two parts. In the first part of this chapter we review various techniques for image and video super-resolution imaging. We begin by presenting the imaging model of a camera which results in the generation of low resolution (LR) images (Section 2.1). We then discuss several super-resolution (SR) approaches that are used to fuse these LR images together. We categorize these approaches into frequency domain (Section 2.3) and spatial domain (Section 2.4) approaches and discuss these in some detail. Closest contemporary work dealing with spatio-temporal SR and the need for accurate alignment and registration between the video sequences is highlighted in Section 2.5. In the latter part of the chapter we review related work in video alignment and registration and also briefly introduce various image processing and computer vision algorithms that are used in this thesis.

2.1 Imaging Model and Low Resolution Image Generation

Super-resolution (SR) is the process of combining multiple low resolution (LR) images or videos to form a higher resolution (HR) image. Nearly all SR algorithms are

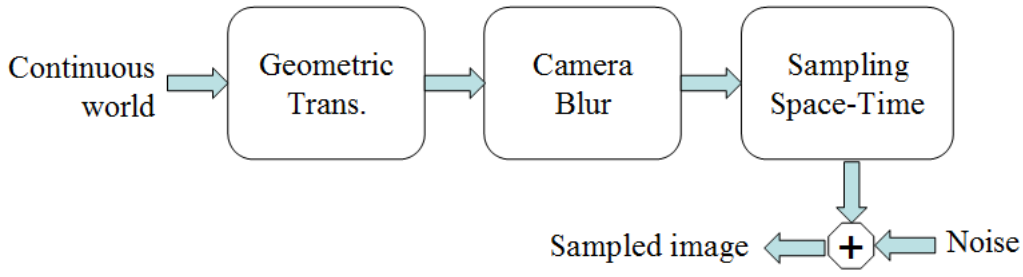


Figure 2.1: General imaging model of the generation of low resolution images. (Trans.=Transformation)

based on the premise that the LR images are blurred and down-sampled versions of the SR images. A general model of the generation of LR images is shown in Fig. 2.1, where a continuous HR scene undergoes a geometric transformation from the world coordinates to the camera coordinates and looses spatial resolution as the camera's point spread function leads to some blurring. The scene is also sampled in both space and time and we can assume that through this entire process of acquisition some noise gets added to the image, leading to a LR representation of the real scene.

2.2 Categorization of Super-resolution Imaging Techniques

Past research in SR imaging can be categorized in two ways. The first categorization looks at the problem domain that is addressed – namely, *image* super-resolution or *video* super-resolution. We call this categorization 'image versus video SR'. The second categorization looks at the techniques or methods for SR imaging – namely, spatial versus frequency domain SR techniques. These two categorizations are discussed in further detail in the following sections.

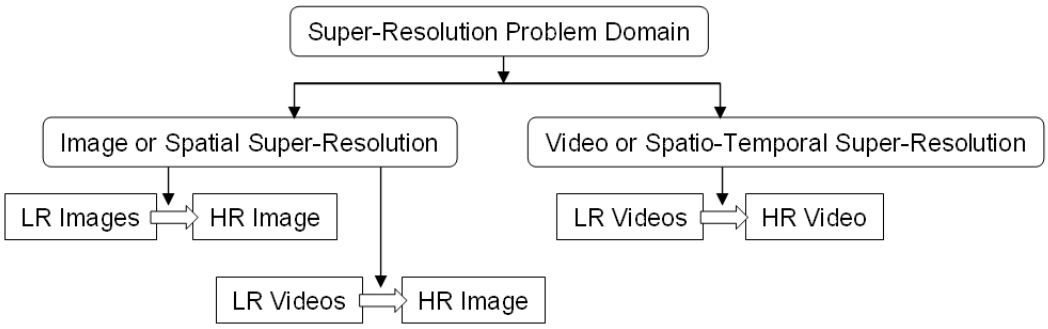


Figure 2.2: Super-resolution categories based on problem domain - Image SR versus Video SR.

2.2.1 Image vs. Video SR

Image SR (or Spatial SR) deals only with the spatial aspect of the imaging model and combines LR images (static images or frames from a video sequence) to generate a high resolution image, as depicted in Fig. 2.2. Note that, even though some image SR techniques use video sequences as their input, they do not improve the temporal resolution of the sequence. Instead, the image SR techniques assume that adjacent frames in the video sequence have some spatial shifts due to camera or object motion, and use this variability in the individual frames to generate a single high resolution image [71].

Video SR (or Spatio-Temporal SR) on the other hand also incorporates the temporal aspect of the imaging model and combines LR videos to generate a high spatio-temporal resolution video [55][5][6].

2.2.2 Spatial vs. Frequency Domain SR Techniques

In this categorization, emphasis is placed on the solution domain of SR techniques – whether the technique operates in image space (pixel values and locations) or in image frequency, see Fig. 2.3. The earliest work in SR perhaps dates back to Huang and Tsai's [74] frequency domain approach, where the relation between the LR video sequences was developed in the frequency domain and the reconstructed

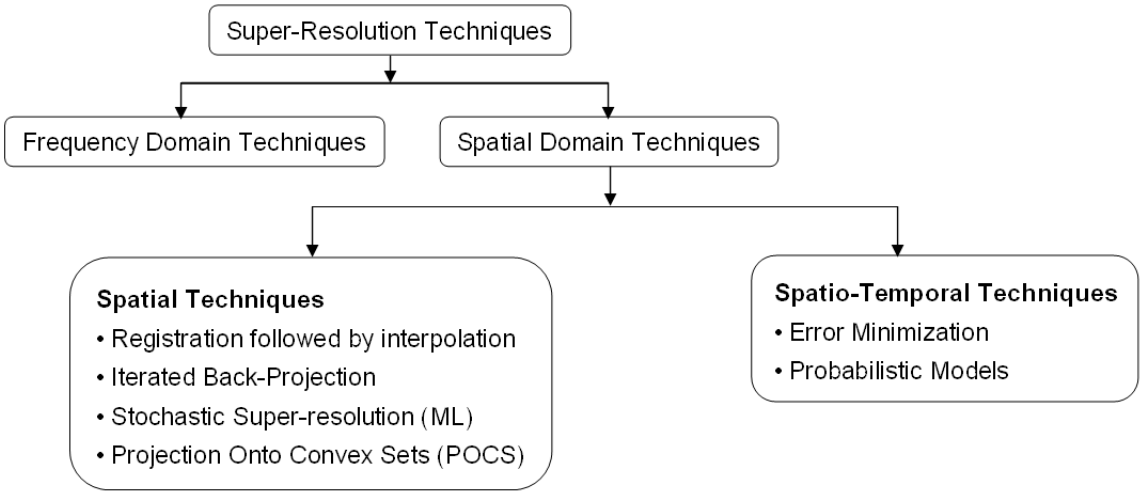


Figure 2.3: Classification of SR approaches into frequency domain and spatial domain. ML-Maximum Likelihood.

HR image is also computed in the frequency domain. Since then numerous SR algorithms have been proposed in the literature, mostly in the spatial domain where the imaging model and reconstruction is computed in the image space itself. The methods proposed in this thesis are in the spatial domain.

2.3 Image SR – Frequency domain Techniques

Frequency domain approaches [74][77] are based on two properties of the Fourier Transform: (i) A spatial shift in an image represents a phase shift in the Fourier transform (FT), (ii) the continuous FT (CFT) is related to the discrete FT (DFT) by an aliasing relationship. An assumption that is made in these techniques is that the original scene (image) is band-limited (does not have infinite spatial frequencies). Generally only planar motion parallel to the image plane is allowed. We explain the frequency domain technique proposed by Tsai *et al.* [74] in the following paragraph.

Let $\{\mathbf{f}_k\}_{k=1,..,p}$, be p LR frames acquired with some shift δ_k. A LR frame, $\mathbf{f}_k$ can be considered to be sampled from a continuous HR signal $f(x + \delta_k)$, where $f(x)$ is the ideal continuous image. If $F_k(\omega)$ and $F(\omega)$ are the Fourier transforms

of $f(x+\delta_k)$ and $f(x)$, respectively, then the FTs are related to each other as follows:

$$F_k(\omega) = e^{i\delta_k\omega}F(\omega). \tag{2.1}$$

The Discrete Fourier transforms (of the discrete LR images and the desired HR image) can be expressed as follows:

$$\mathbf{f}_k = \Phi\mathbf{F}_k, \tag{2.2}$$

where $\Phi = (exp(-\frac{2\pi i}{N}jn))$ is the factor that represents the shift, $j = 0..N-1$, and N is the number of pixels in the image. Thus, the relation between the known DFT coefficients of LR images **f**, the shift Φ and the unknown HR DFT coefficients **F** can be expressed as as a system of linear equations.

During implementation, one of the LR images is considered to have zero offset from the desired HR image, and is therefore the reference LR image. The DFTs of the remaining LR images are then compared to the DFT of the reference LR image to find Φ_k, or the offset between each LR image and the desired HR image. These multiple DFTs and offset values (Φs) are stacked in a matrix form and solved using numerical techniques to obtain **F**. The inverse DFT is applied to **F** to obtain the reconstructed HR image.

While the frequency domain approaches are theoretically simple and have low computational complexity, they are disadvantaged by the fact that the transformation relating the LR images has to be a *global* translation, rotation or scaling, *i.e.* all the pixels in an image undergo the same transformation. In the case of video sequences, global offsets imply that all frames in a video sequences are related by a single offset value to corresponding frames in a second sequence. The problem we are addressing in this work is not restricted to global transformations between images. Rather, we are trying to solve for the multiple offsets existing between frames of two (or more) video sequences. Hence frequency domain approaches can not be applied in our proposed work.

2.4 Image SR – Spatial Domain Techniques

In this category of SR techniques, the relation between the observed LR images and the unknown HR image is formulated in the spatial domain. The reconstruction of the HR image is also performed in the spatial domain. The spatial domain approaches can model complex transformations between the LR images, and even account for motion, and hence are a more global solution to the SR reconstruction problem. We now review the prominent categories of SR reconstruction algorithms in the spatial domain.

2.4.1 Registration and Interpolation of Non-uniformly Spaced Samples

Registration of multiple LR images based on motion compensation (subpixel displacement between the LR images) leads to a dense composite image with non-uniformly spaced samples. The reconstruction process used to interpolate between these LR images' set can be visualized as follows: the LR images are piled on top of each other based on the sub-pixel displacement. Each pixel in the SR image is then computed by averaging the supporting pixels in the LR pile [69]. Such interpolation methods are overly simplistic and incapable of reconstructing missing frequencies from spatially averaged areas and under-sampled data.

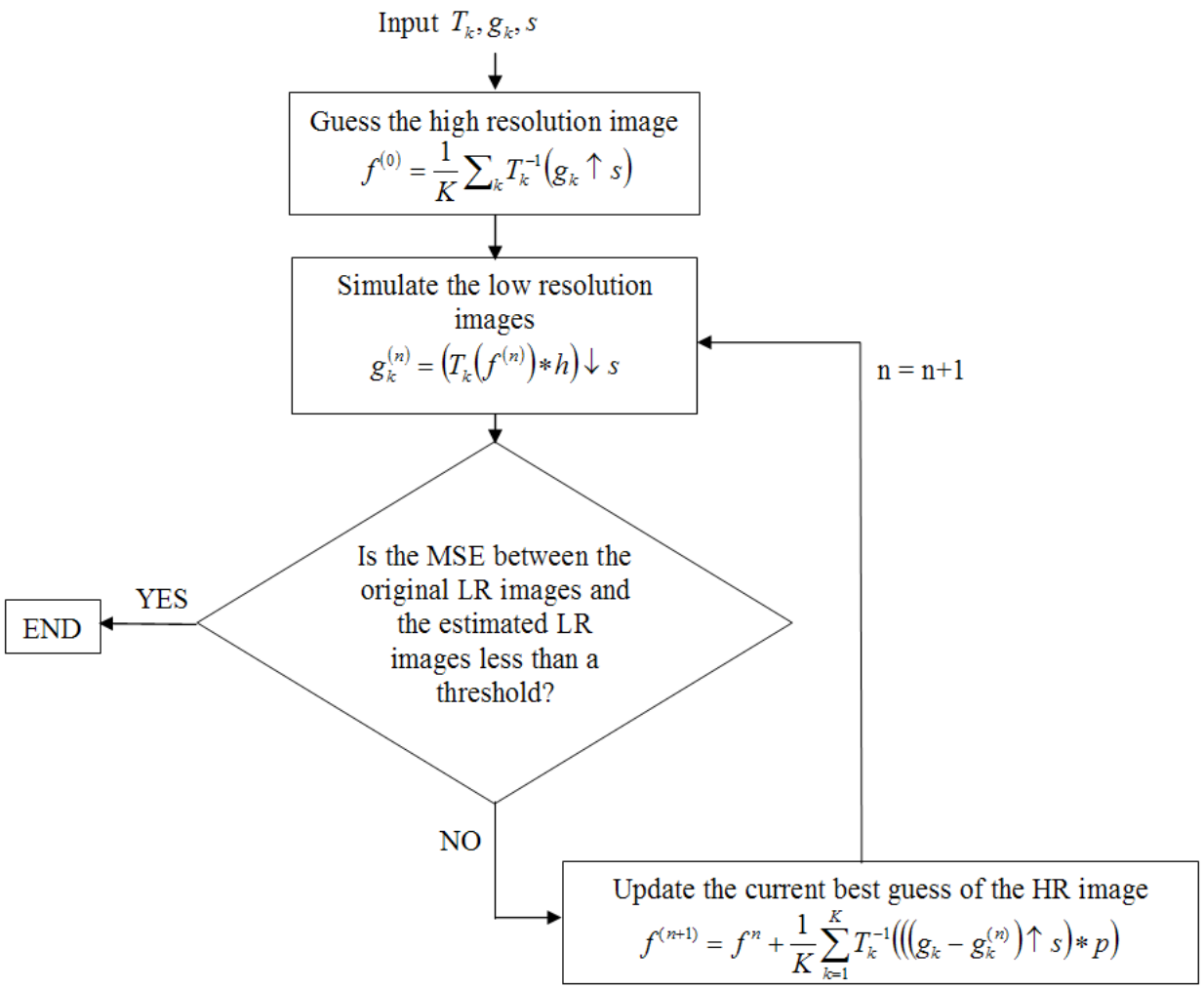


Figure 2.4: Irani-Peleg Iterated Back Projection Algorithm for image SR (adapted from [28]). MSE- Mean Square Error.

2.4.2 Iterated Back-Projection

The iterated back-projection algorithm introduced by Irani and Peleg [28] is perhaps the most widely used SR algorithm [23][33][10]. The general idea of the iterated back-projection algorithm is illustrated in Fig. 2.4, where the following terminology has been used.

- g_k is the k^{th} observed LR image.
- f is the HR image of the object (desired).
- T_k is the 2D geometric transformation from f to g_k which is determined by the motion parameters of the tracked object and is assumed to be invertible.
- h is a blurring operator which represents the point spread function of the camera. When unknown, this is assumed to a Gaussian.
- s is the sampling factor.
- η is the observation noise.
- $\uparrow$ is the upsampling operator.
- $\downarrow$ is the downsampling operator.
- k refers to the k^{th} LR image.
- p is the back-projection operator. The back-projection operator can be used to control the influence of each LR image on the reconstructed image. If all LR images are considered to be of equal significance then p can be set to 1.

The imaging or generative model can be represented as follows:

$$g_k = (T_k(f) * h) \downarrow s + \eta. \tag{2.3}$$

The geometric transforms T_k are computed between pairs of images by extracting image feature points and using the corresponding features to derive a spatial relationship between the image pair. Various methods that can be used to derive these transforms are presented in Section 2.7.

An initial guess of the HR image f is taken by simply upsampling the K LR images by the sampling factor s, applying the inverse geometric transformation T_k^{-1} that created the LR images and summing over all the pixels in K. Then, the LR images are estimated by simulating the imaging process. The mean square error (MSE) of the estimated and originally acquired LR images is 'backprojected' on to the initial guess to yield an improved HR image f^1. This process is repeated iteratively to minimize the following error function:

$$e^{(n)} = \sqrt{\frac{1}{K}\sum_{k=1}^{K} \|g_k - g_k^{(n)}\|_2^2}, \tag{2.4}$$

where n represents the n^{th} iteration. Intuitively, Eq. 2.4 computes the difference between the actual LR images g_k and the LR images that are generated by applying a generative model to the current estimate of the HR reconstructed image, $g_k^{(n)}$. When the most accurate HR image is reconstructed, then the difference between the real LR images and the generated LR images will be minimum.

Unfortunately, iterated back-projection methods do not lead to a unique SR reconstruction as the SR update is an ill-posed problem. Also, inclusion of *a-priori* constraints such as smoothness and edges are not easily achieved in the iterated back-projection method.

2.4.3 Projection onto Convex Sets

Patti *et al.* [49] and Stark *et al.* [65] address the problem of solving the generalized imaging model for an unknown SR image, (given multiple LR images captured at sub-pixel displacements), by the method of projection onto convex sets (POCS). In a vector space, a set C is convex if $\forall \overrightarrow{x} \in C$ and $\forall \overrightarrow{y} \in C$ the following relationship

is satisfied:

$$\lambda \overrightarrow{x} + (1-\lambda)\overrightarrow{y} \in C \quad \forall 0 \leq \lambda \leq 1. \tag{2.5}$$

SR reconstruction problem can be described in the form of convex set restrictions, where all the available LR images impose restrictions on the HR solution. The HR solution will satisfy all the imposed restrictions. Therefore, finding the HR image consists of finding the intersection of all the LR images. If the LR images are considered to be closed and convex and the HR solution exists, then successive projections of the solution vector onto the LR image convex sets will converge to the intersection of the LR images.

Let $g(m_1, m_2, k)$ be a sequence of LR images (where m_1, m_2 and k refer to the image coordinates and time coordinate respectively). Then for any arbitrary member y the convex constraints set can be defined as follows:

$$C_{tr}(m_1, m_2, k) = \{y(n_1, n_2, t_r) : \quad |r^{(y)}(m_1, m_2, k)| \leq \delta_0(m_1, m_2, k)\}, \tag{2.6}$$

where $r^{(y)}(m_1, m_2, k) = g(m_1, m_2, k) - \sum_{n_1,n_2} y(n_1, n_2, t_r) \cdot h_{tr}$ is the residual error associated with y. h_{tr} is the camera blurring and sampling model at arbitrary time t_r. This equation implies that for a vector to be part of a convex set, the difference between the LR image vector (is a known vector) and the blurred and downsampled version of the arbitrary vector y, must be less than δ_0. This sets a threshold around the LR image values so that a solution vector that falls below the threshold is accepted as a viable result. In other words, $\delta_0(m_1, m_2, k)$ is a bound representing the confidence that the actual image is a member of the convex set C_{tr}.

The projection P_{tr} of an arbitrary HR solution vector $x(n_1, n_2, t_r)$ on the convex constrained set $C_{tr}(m_1, m_2, k)$ is defined as follows:

$$P_{tr}[x(n_1, n_2, t_r)] = x(n_1, n_2, t_r) + \begin{bmatrix} (r^{(x)} - \delta_0) \cdot h_{tr}, & r^{(x)} > \delta_0 \\ 0, & -\delta_0 \leq r^{(x)} \leq \delta_0 \\ (r^{(x)} + \delta_0) \cdot h_{tr}, & r^{(x)} < \delta_0 \end{bmatrix}. \tag{2.7}$$

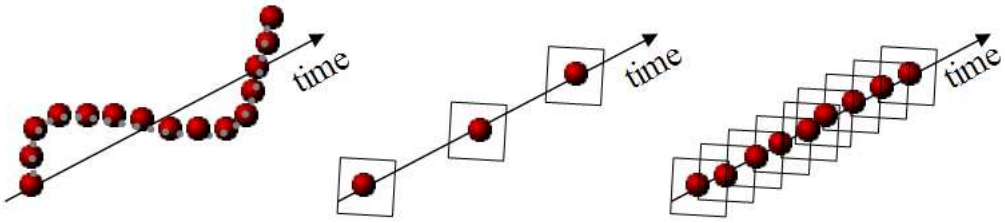
This equation implies that an arbitrary vector x (which is the initial guess of the HR image) is iteratively updated based on the residual (difference) between the

blurred and downsampled version of x and the LR images g. Additional constraints on the amplitude, such as members of the convex set can only take values from 0 to 255, can also be applied. Given the above projections, the composition of the projectors onto the family of sets is iteratively computed. The initial guess of the SR image is computed as a bilinear interpolation of one of the low-resolution images. The iterations of the POCS algorithm stop when either the intersection of all the constraint sets is reached or if by visual inspection (or a metric) the update on the SR image (change between successive estimates) is lower than a preset threshold.

POCS is an interesting approach to SR reconstruction based on a spatial domain observation model. The constraint sets allow for the inclusion of *a priori* information to the solution. However, the rate of convergence and the reconstructed SR image is dependent on the initial guess of the image. Also, the solution is non-unique as the POCS iteration stops when a point in the intersection of all constraint sets is found or the change between successive iterations decreases below a threshold.

2.5 Video SR – Spatio-temporal Superresolution

Spatio-temporal SR is distinct from spatial SR in terms of the temporal effects of the acquisition model. There are two primary temporal effects observed in video acquisitions which are not encountered in spatial SR. The first relates to dynamic events happening at rates faster than the frame rate of the camera. In such cases, either the event is not captured at all or is captured incorrectly. Shechtman *et al.* [55] refer to this as temporal aliasing. The second temporal effect relates to the exposure time of the camera, and though it is visible as a spatial artifact, the cause of this motion blur lies in the temporal domain.



(a) (b) (c)

Figure 2.5: Temporal aliasing (adapted from [55]). (a) Trajectory of a ball over time. (b) Trajectory sampled over time by a low frame rate camera. Perceived trajectory is along a straight line. (c) Illustration that even with perfect temporal interpolation between the sub-sampled frames of (b) the true motion trajectory cannot be recovered.

2.5.1 Temporal Effects in Video SR

- **Temporal aliasing**: As illustrated in Fig. 2.5, if the frame rate of the camera is below the Nyquist frequency of the trajectory of motion, then the true motion of the object cannot be recovered, even by performing ideal temporal interpolation. Figure 2.5, also illustrates the requirement of multiple sequences for temporal SR since, interpolation within a single sequence cannot recover the true motion.

- **Motion blur**: Motion blurring can be understood as the integration of the light received at each pixel during the exposure time of the camera, see Fig. 2.6.

2.5.2 Video SR Categorization

Current research in spatio-temporal SR can be divided into three categories: (i) True spatio-temporal SR, (ii) Spatio-temporal resolution enhancement and (iii) Spatio-temporal interpolation, see Fig. 2.7. The original idea of spatio-temporal alignment of sequences was introduced by Caspi and Irani [6] in 2002. They (along with Shechtman [55]) extended their work, to space-time SR by incorporating the temporal alignment algorithm from [5].

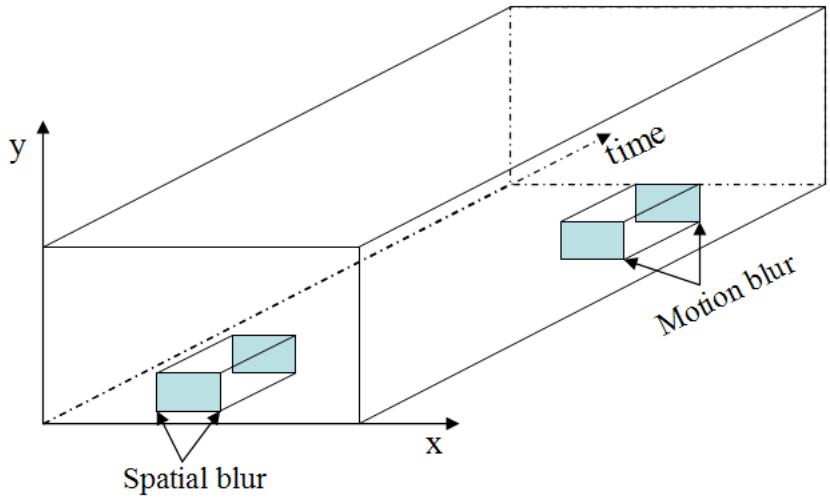


Figure 2.6: Illustration of Motion blur and Spatial blur caused due to exposure time and the point spread function of the camera respectively (adapted from [55]). The rectangle indicated by spatial blur illustrates the spatial area that will be averaged to a single pixel value in the discretized image. The temporal blur rectangle indicates the time-period over which temporal information will be averaged to a single frame in the video sequences.

Others have adapted the algorithm proposed by Shechtman *et al.* to enhance a low frame rate video based on a high frame rate video of the same scene [26] — spatio-temporal resolution enhancement. In spatio-temporal resolution enhancement, one video stream is available at high resolution, and scene information from the HR video is added to the LR video to improve the LR video quality. This enhancement is limited primarily to improving the resolution of static background objects. Spatio-temporal interpolation techniques are the standard methods used for frame rate up-conversion or for increasing the temporal resolution of videos. We will discuss the Caspi algorithm and the space-time SR algorithm in detail next. Other resolution enhancement and interpolation techniques are also briefly reviewed.

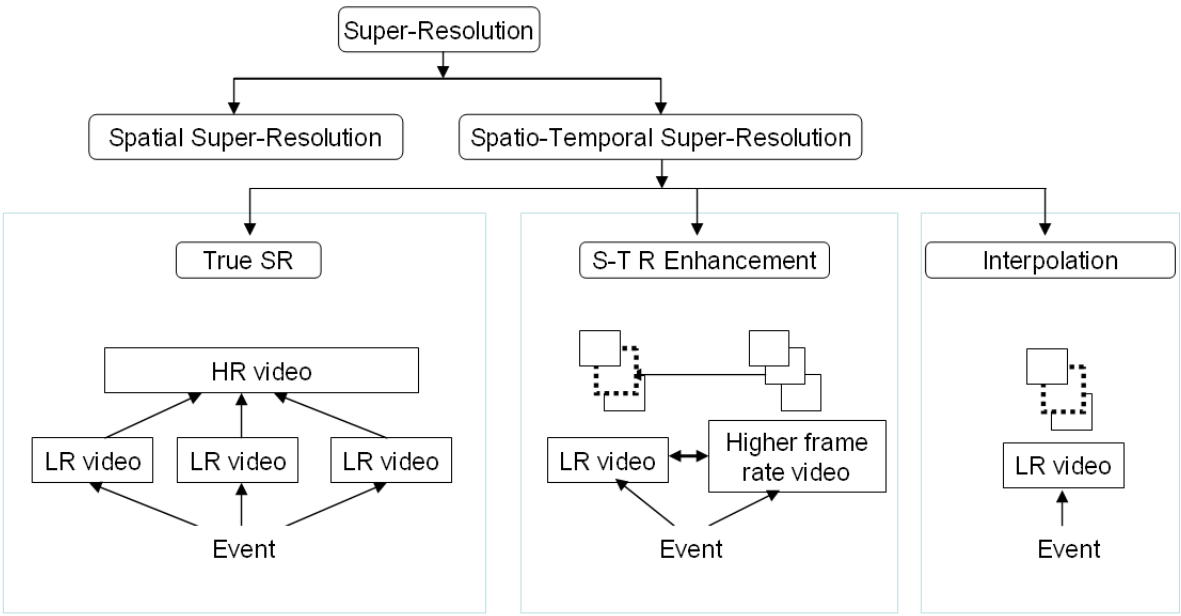


Figure 2.7: Different categories of spatio-temporal SR techniques.

2.5.3 Space-time SR

This section is a detailed review of the Shechtman, Caspi and Irani algorithm [55][5]. Let S be a dynamic space-time scene which is captured by n different video cameras as sequences $\{s_i\}_{i=1}^{n}$ of limited spatial and temporal resolution. The imaging process is modeled as blurring followed by sampling both in time and space. The blurring effect can be explained as the point spread function (PSF) of the camera (spatial) and exposure time or aperture time of the camera (temporal). The sampling effect also has both time and space components. Temporal sampling occurs because of the limited frame rates of the cameras and spatial sampling occurs because of limited detectors on the camera.

A general dynamic scene can be represented in 4D as (x, y, z, t). However if the scene is planar and the distances between the cameras is small compared to their distance from the scene, then the scene can be represented as a 3D space-time volume (x, y, t). Let one of the LR sequences be the reference LR sequence (any sequence can be chosen without loss of generality), say s_1. S^h is the HR dis-

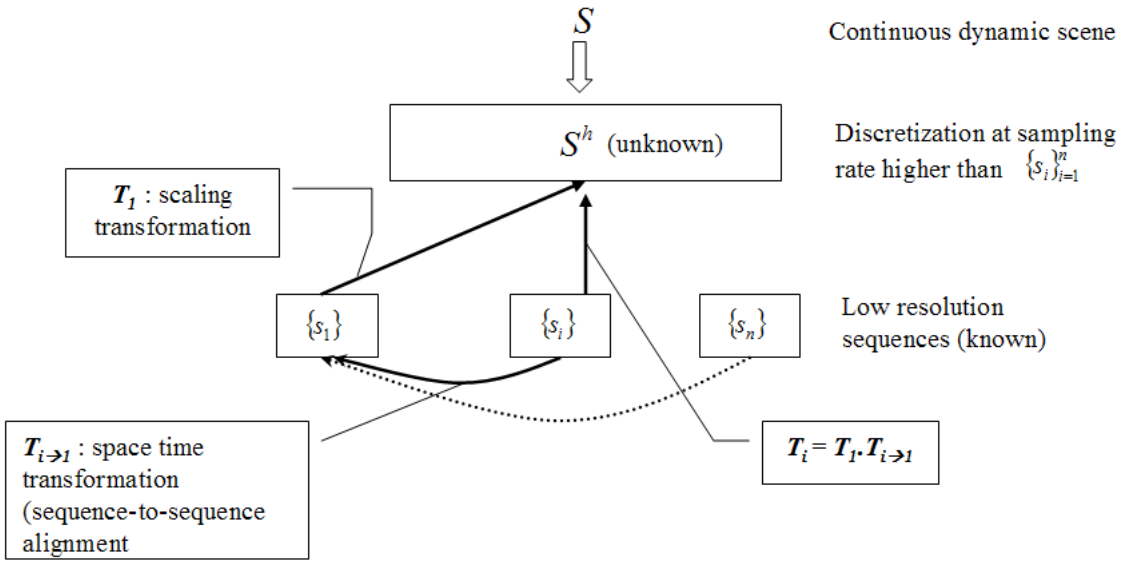


Figure 2.8: Relation between various transformations of the Shetchman model[55].

cretization of the continuous scene (at a higher sampling rate than $\{s_i\}_{i=1}^n$). The scaling transformation between s_1 and S^h can be represented as T_1. If $T_{i\to 1}$ is the space-time coordinate transformation from the i^{th} LR image to the reference LR sequence s_1, then T_i (transformation from i^{th} LR sequence to S^h) can be represented as $T_i = T_1 \cdot T_{i\to 1}$. Refer to Fig. 2.8 for an illustration of this relation.

Spatio-temporal Alignment of Sequences

Caspi *et al.* [5] model $T_{i\to 1}$ (sequence to sequence alignment) as follows. Temporal misalignment between the sequences arises because of different frame rates of the cameras and offset in the activation time. This temporal misalignment is modeled as a 1D affine transform. The spatial misalignment resulting from different camera calibration parameters is modeled as an inter-camera homography. Let s and s' be two LR input image sequences. Let $\mathbf{x} = x, y, t$ be a space time point in sequence s. Suppose this space time point is captured as a pixel at location (x, y) and frame number n. Similarly, let $\mathbf{x}' = x', y', t'$ be a space time point in sequence s', which is captured as a pixel at location (x', y') and frame number n'. In order to find the

spatial relationship between (x, y) and (x', y'), we need to represent these image points as homogenous coordinates. Homogeneous coordinates make it easier to represent and apply transformations (such as affine transform) as matrices. Let p and p' be the homogeneous representation of the image coordinates (x, y) and (x', y') respectively. If we assume a planar scene, then p and p' are related to each other by a homography (described in Section 2.7.2).

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{2.8}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{2.9}$$

The temporal misalignment between the two sequences, s and s', is modeled as:

$$t' = r.t + \Delta t, \tag{2.10}$$

where r is the ratio of frame rates of the cameras and Δt is a translation. The algorithm is implemented by extracting features of interest such as corners and maintaining a feature trajectory. Spatio-temporal alignment is implemented by recovering the alignment between the feature trajectories by minimizing the following function:

$$P = arg \min_{H, \Delta t} \sum_{trajectories} \left(\sum_{t \in trajectory} ||p'(r.t + \Delta t) - H(p(t))||^2 \right), \tag{2.11}$$

where H represents the homography computed between the feature points in frame n in sequence 1 and frame n' in sequence 2. Note that the same matrix H will apply to all frames in a sequence. Equation 2.11 is implemented as follows. An initial value of H and Δt are chosen. The homography is applied to one of the trajectories, say $p(t)$. Then, the difference between the feature point coordinates is computed between frame 1 of sequence 2 and frame $(1 + \Delta t)$ from sequence 2. Similarly, differences for all other corresponding feature coordinates is computed (for an offset

Δt and homography H), the differences are squared and summed. This operation represents the first summation in Eq. 2.11. As a video sequence is not limited to a single trajectory, the above operations are performed for all trajectories in the two sequences, and the squared differences for all trajectories are also summed. This is represented by the second summation in Eq. 2.11. Since t' is not necessarily an integer value, it is linearly interpolated from its neighboring values. The minimization is performed by first fixing Δt and solving for H, and then fixing H and refining Δt. The iterations are stopped when the residual error does not change. The values of H and Δt that result in the minimum error P (in Eq. 2.11), are the solution.

Spatio-temporal Reconstruction

Once the sequence alignments T_i are computed using the above method (Caspi algorithm) the next step is to reconstruct the SR video. In reconstructing the HR sequences, spatial blurring caused by the camera's spatial point spread function and temporal blurring caused by the cameras exposure time, need to be accounted for. Let the combined space-time blur of the i^{th} camera be represented as $B_i = B(\sigma_i, \tau_i, p_i^l)$, where $p_i^l = (x_i^l, y_i^l, t_i^l)$ is the LR ('l') space-time point, σ and τ are the standard deviations of the PSF in space and in time respectively. Let $p_i^h = (x_i^h, y_i^h, t_i^h)$ be the corresponding HR point (which is to be computed). Also, let $S(p)$ represent the pixel value corresponding to point p. Thus, the known LR images $S_i^l(p_i^l)$ are related to the unknown HR $S(p^h)$ by the following relation:

$$S_i^l(p_i^l) = B_i^h * S(p^h), \tag{2.12}$$

where $B_i^h = T_i \cdot B(\sigma_i, \tau_i, p_i^l)$ is the blur kernel in the HR coordinate system. Equation 2.12 models the blurring and downsampling that a HR image pixel undergoes before being captured as a LR image pixel. In this equation, the values of $S_i^l(p_i^h)$ are available, the values of B_i^h are approximated and $S(p^h)$ are the values that are to

be computed. If we stack all the LR images that are available into a large system of linear equations, then all the LR images are related to the same HR image by different space-time blur parameters. This large system of linear equations is represented as:

$$\overrightarrow{l} = A\overrightarrow{h}, \tag{2.13}$$

where $\overrightarrow{l}$ is a vector containing ordered LR pixels, A is the sampling and blur PSF (stacked B_i^h)matrices and $\overrightarrow{h}$ is a vector containing the unknown HR pixels. This system of linear equations is solved for h using conjugate gradients.

2.5.4 Epitomes

Another probabilistic approach that has been used for SR is termed as Epitomes. Epitomes [31][7] are patch based probability models that are generated by combining together a large number of example patches from input images. These example patches represent some of the high-order statistics in images and video data such as texture, shape and optic flow. Epitomes are learned and can be considered to be condensed versions of image patches in video sequences. For example, a low resolution input video sequence is broken down into small sets of 3D volumes (space-time). A probabilistic generative model is then used to learn the video epitome that could best describe the low resolution image patches. Epitomes have been demonstrated to be fairly successful in video inpainting, compression and mosaicking.

2.5.5 Frame Rate Up-conversion and Temporal Interpolation

Frame rate up-conversion is required for applications such as NTSC– PAL conversion and display on HDTV, where high frame rates are desired [72] [32]. The standard frame rate up-conversion methods are frame repetition, linear interpolation, motion compensated interpolation. Motion compensation is usually bi-directional in order to take into account frames on both sides of the up-converted frame. Motion

compensation and interpolation however, cannot deal with temporal aliasing, as described in Section 2.5. We mention interpolation in this thesis because it is currently the method of choice for most temporal resolution enhancement approaches.

It can be inferred from the discussion on SR reconstruction that finding a correct alignment between the LR sequences is paramount; hence in the following section we will review some approaches to achieve sequence-to-sequence alignment. Note that in video processing literature, both the terms – 'alignment' and 'synchronization', have been used to refer to the same concept.

2.6 Video Synchronization

Video Synchronization techniques can be classified based on the the source of the synchronization information. In some scenarios, this synchronization information is acquired from an external source, while in other scenarios this information is acquired from the video sequences themselves. We look at techniques in both these categories next.

2.6.1 External Gating Signal based Synchronization

Currently, most synchronization algorithms (especially in medical imaging) use an external timestamp to align multiple datasets together. For example, in cardiac imaging [16][63], an ECG signal and MRI images from the heart are captured concurrently. The unique cardiac peaks in ECG are used as landmark points to align corresponding cardiac images. It is however not necessary that a direct reading from the imaging region be used for time stamping. Yang *et al.* [86] records audio data of subjects reciting consonants and vowels while simultaneously imaging the subjects via ultrasound. The speech patterns in the audio data are then used as timestamps to align the ultrasound data in order to visualize tongue movement. Others have also tried multiple camera acquisitions with a controlled triggering of the onset of cap-

ture, such that the sequences are offset by known subframe displacements. Then, by simply exploiting the offset and the frame rates of the cameras, multiple video sequences can be combined.

2.6.2 Image based Synchronization

A more generalized form of temporal synchronization is based on information derived from the images themselves, such as, change in region properties such as brightness [70][72] (in cardiac images) or motion of regions of the image[21]. Listgarten *et al.* [40] use a Hidden Markov Model (HMM) based approach for alignment of continuous time series data from speech and mass spectrometry. They present a continuous profile model (CPM) which assumes that each acquired or observed time series is a noisy sub-sampled representation of a single true time series, which they call latent trace. The noisy time series are generated by moving through a sequence of Hidden Markov states. The CPM is trained using expectation maximization , and subsequently the latent trace of the model which represents a higher resolution series is obtained. However, the CPM algorithm only performs global alignments and a large number of replicated experimental data is required to train the HMM. Giese and Poggio [21] model biological motion patterns using linear combinations of prototypical sequences with the objective of synthesizing motion of objects. For example, given a sequence of walking, they warp the temporal duration of the 'walk' by making the first section of a step faster to synthesize a 'limping' sequence. They use a dynamic time warping approach to vary the time duration of activities. They modeled the temporal deformation as a non-parametric transformation as shown below.

$$t' = t + \tau(t) \tag{2.14}$$

However, the underlying problems of correspondence and warping are ill-posed. Since even in the absence of ambiguity in the features being compared, there are

infinitely many possible solutions that can bring the two trajectories into correspondence. Other such as Patti *et al.* [49] have also used DTW for alignment of curves, however they constrain their warping model to a linear warp.

Perperidis *et al.* [50], use the Caspi model [5] described in Section 2.5.3 for temporal registration of cardiac images and enhance it by incorporating local deformable transformations using 1D cubic B-splines. Their temporal transform is represented as follows:

$$T_{temporal}(t) = T^{global}_{temporal}(t) + T^{local}_{temporal}(t). \tag{2.15}$$

The global transform has been modeled like Caspi's temporal transformation model (as shown in Eq. 3.7), where α accounts for scaling differences and β accounts for translation differences.

$$T^{global}_{temporal} = \alpha t + \beta. \tag{2.16}$$

The local transform has been modeled as splines using the following equation:

$$T^{local}_{temporal} = \sum_{l=0}^{3} B_l(u)\phi_{t_{i+1}}, \tag{2.17}$$

where B_l represents the l^{th} basis function of the B-Spline and ϕ denotes a set of control points with a uniform temporal spacing. The optimal temporal transform is found by maximizing the normalized mutual information between the cardiac datasets. The local temporal transform is also computed separately by finding transitional landmarks and then searching through all possible local deformations of the splines while optimizing the normalized mutual information. They report that computing the optimal deformation by their approach is computationally expensive and sometimes takes over 24 hours to resolve. Their spline model is based on computing transitional landmarks such as start of the cardiac cycle, maximum contraction, end diastole etc., and depends greatly on the accuracy of this landmark recovery stage. For trajectories where such local landmarks are not as easily distinguishable, the

temporal transformation will reduce to a global transformation, and thus to Caspi's model.

Past literature in video synchronization and temporal registration can be broadly classified into two categories — video sequences of the same scene or video sequences of similar scenes; differing primarily on the assumptions made with respect to the temporal offset between sequences.

2.6.3 Video Synchronization of Same Scene

In synchronizing videos of the same scene, the temporal offset is considered to be an affine transform [5], as described in the previous section. Dai *et al.*[8] use 3D phase correlation between video sequences for synchronization. The 3D phase correlation is computed in the Fourier domain. First the Discrete Fourier Transform (DFT) of the frames is computed. The DFT returns a magnitude spectrum and a phase spectrum. Dai *et al.* shift the temporal position of one video sequence, and iteratively compute the spatial registration between individual frames in both sequences (assuming a homography) and compute the correlation of the phase spectra of the corresponding video frames. The time shift that results in the highest correlation value is used to synchronize the video sequences. However, their algorithm only works for 2D planar scene sequences and the temporal offset is limited to a translation.

Another method for synchronization of the same scene has been proposed by Tuytelaars *et al.*[76]. They compute synchronization by checking the rigidity of a set of *five* (or more) points. The rigidity is computed as a rank-constraint on the spatial homography or fundamental matrix. We discuss the rank-constraint based method in greater detail in Chapter 4, where we compare our work with the rank-constraint based method. The central idea behind the rank-constraint based algorithm is that for matched points moving non-rigidly in two sequences of the same

Table 2.1: Review of related work. Legend: D.S.-Dynamic time shift, V.I.-View Invariant, S.A.-Sub-frame Accuracy

Author	**Scene**	**D.S.**	**V.I**	**S.A.**
Caspi *et al.* [6] [5]	Same	X	$\surd$	$\surd$
Tresadern *et al.* [73]	Same	X	$\surd$	$\surd$
Lei *et al.* [37]	Same	X	$\surd$	X
Carceroni *et al.* [4]	Same	X	$\surd$	$\surd$
Wolf *et al.* [84]	Same	X	$\surd$	X
Pooley *et al.* [52]	Same	X	$\surd$	$\surd$
Dai *et al.* [8]	Same	X	$\surd$	$\surd$
Tuytelaars *et al.* [76]	Same	X	$\surd$	X
Giese *et al.* [21]	Diff	$\surd$	X	X
Perperidis *et al.* [50]	Diff	$\surd$	X	X
Rao *et al.* [54]	Diff	$\surd$	$\surd$	X

motion, the 4^{th} singular value in the case of a homography and 9^{th} singular value for fundamental matrix, will determine if the correct temporal alignment has been computed. A large singular value implies that the sequences have not been synchronized properly. Tresadern *et al.*[73] also follow a similar approach of computing a rank-constraint based rigidity measure between *four* non-rigidly moving feature points. They are able to reduce the rank number by 1 by translating the feature points such that the origin of the image lies at the centroid of the feature points. Caspi *et al.*[5] recover the spatial and temporal relation between two sequences by minimizing the SSD error over extracted trajectories that are visible in both the sequences. Carceroni *et al.*[4] extend [5] to align sequences based on scene points that need to be visible only in two consecutive frames. They approach the problem of synchronization by assuming that the pair wise correspondence between frame number of input sequence ($t_i = \alpha_i t_1 + \beta_i$) induces a global timeline relationship ($L = \{[\alpha_1....\alpha_N]^T t + [\beta_1...\beta_N]^T\}$) between the sequences. This global time line implies that even if we do not have knowledge of the temporal alignment of the entire sequence, we can construct the line from pair-wise correspondences from a few dynamic features that are visible in two or more frames of the scene.

2.6.4 Video Synchronization of Related Scenes

When aligning video sequences of different but related scenes, *i.e.* sequences correlated via motion, one has to factor in the dynamic temporal scale of activities in the video sequences. Giese and Poggio [21] approach alignment of activities of different people by computing a dynamic time warp between the feature trajectories. However, the problem when the activity sequences are from varying viewpoints is not addressed, and the approach computes a one-to-one frame correspondence. Perperidis *et al.* [50] have proposed an algorithm to locally warp cardiac MRI sequences. The algorithm extends Caspi's work [5] to incorporate spline based local alignment. Though this approach does lead to good alignment of time-varying sequences, it has two main drawbacks: (i) the computation space for spline based registration is quite large and the authors need to compute points of inflexion in the cardiac volume change; and (ii) the alignment is still a one-to-one frame correspondence and not sub-frame accurate. Others, such as Rao *et al.* [54] use rank constraint as the distance measure in a dynamic time warping algorithm to align multiple sequences. Note that this is the first work that can deal with video sequences of correlated activities.

In the previous discussion we reviewed various techniques for sequence-to-sequence alignment. In the following sections we will review techniques for spatial registration. Spatial registration is required to compute a pixel-by-pixel correspondence between images or video frames. While there are various methods for spatial registration, we will only review those methods are either directly used in this thesis or after suitably modifications.

2.7 Spatial Registration

When a 3D scene is acquired by a camera, a spatial point in 3D space is transformed into a 2D point on the image plane. When multiple cameras are used, different

camera acquisition matrices result in the same 3D point being transformed into multiple 2D image points. Spatial or Image registration involves transforming the 2D points in one image into the coordinate system of a second image, such that data between the two images can be compared or integrated.

Registration can be either rigid or elastic. In rigid registration a *single* transform matrix is applied to all the pixels in the image, whereas, in elastic (or non-rigid) registration the transformation for each pixel is independent of each other. Non-rigid registration is generally used when the object deforms between the multiple acquisitions.

Another classification of registration techniques is based on the properties of the images that are used for registration – area based and feature based techniques. In area based techniques, regions are mapped from one image to another, using metrics such as correlation and mutual information. In feature based techniques, features such as edges, corners and lines are extracted in each image independently, and correspondence (mapping) between the features is then computed by using standard feature matching algorithms such as Scale Invariant Feature Transform (SIFT) [42], Intensity Based Regions (IBR) [75] and Maximally Stable Extremal Regions (MSER) [46]. In this thesis we use rigid, feature based spatial registration techniques which are discussed next.

2.7.1 Feature Extraction

The Scale Invariant Feature Transform (SIFT) algorithm has been proposed by David Lowe [42] for feature extraction and feature description. The SIFT algorithm detects extrema in Scale-space and these extrema are called keypoints (features). The keypoints are detected as follows:

- The image is convolved with Gaussian filters at different scales. Gaussian filters are smoothing filters that blur the image. If $I(x, y)$ is the original image,

and $G(x, y, k\sigma)$ is the Gaussian blur at scale $k\sigma$, then the convolved Gaussian-blurred image can be represented as $L(x, y, k\sigma) = G(x, y, k\sigma) \otimes I(x, y)$, where $\otimes$ is the convolution operator.

- Successive Gaussian-blurred images are then subtracted to get a Difference of Gaussians (DoG) image. A DoG image can be represented as $D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma)$.

- Maxima and minima in a DoG image are captured as keypoints. These are computed by comparing each pixel in the DoG image to its eight neighbors on the same scale and eighteen neighboring pixels in the adjacent DoG images.

- Keypoints that have low contrast or are poorly localized around edges are then discarded. This is done to ensure that the keypoints that are chosen are stable across scales.

- In order to achieve rotation invariance, each keypoint is assigned one or more orientations based on local image gradient directions.

- For each keypoint a 128 dimensioned descriptor is generated that contains information about keypoint position, orientation and scale.

The SIFT feature vectors are invariant to minor affine (rotation, scaling, shearing and translation) changes and have been proven to be highly distinctive (*i.e* feature vectors from two different features are distinct). Features extracted in two images can be matched by computing the difference between their feature vectors. This matching is used to compute the spatial transform that relates one image to another. Of particular interest to us is the transform that relates planar scenes – homography, which is discussed next.

2.7.2 Homography

Suppose the scene being viewed by two cameras is planar (for example a wall with graffiti on it) and has limited depth, then the images acquired by the cameras are related by a homography, which is a 3×3 transformation matrix. A standard algorithm used to compute the homography is the Direct Linear Transformation (DLT) Algorithm [25]. Suppose $\mathbf{x}_i$ and $\mathbf{x}'_i$ are 2D point correspondences extracted in Image1 and Image2 respectively. The homography transformation is given by $\mathbf{x}'_i = H\mathbf{x}_i$, where $H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$.

The homography transformation can be expressed as the vector cross product $\mathbf{x}'_i \times H\mathbf{x}_i = 0$. For $n \geq 4$ feature point correspondences, the vector cross product can be solved for H. The homogenous equation for the vector cross product can be written as $Ah = 0$, with A as the following $2n \times 9$ matrix:

$$A = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 & -y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 & -x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 & -y'_2 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_nx'_n & -y_nx'_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_ny'_n & -y_ny'_n & -y'_n \end{bmatrix}, \tag{2.18}$$

and $h = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 & h_8 & h_9 \end{bmatrix}^T$. We compute the SVD (singular value decomposition) of the matrix A. The singular vector corresponding to the smallest singular value is considered to be the solution to h or H.

Note that in an image more than 4 corresponding feature points are extracted. Thus the question now becomes - how can we seek a homography that satisfies the most correspondences. This question is solved by using methods such as RANSAC, which is discussed next.

2.7.3 Random Sampling and Consensus

Random Sampling and Consensus (RANSAC) [18] is a mathematical strategy used to find a solution that agrees with the largest data. For example, once features have been extracted and tentatively matched based on some score, RANSAC strategy is used to randomly choose four feature correspondences to compute the homography. The algorithm then checks to see how many of the remaining feature correspondences agree with the computed homography, these are called 'inliers'. The goal of the algorithm is to maximize the number of inliers, and the homography that results in the maximum number of inliers is deemed to be the solution. RANSAC can result in an incorrect solution when incorrect feature matches are so numerous that they overwhelm the correct feature matches.

2.8 Motion Tracking

Motion in video sequences is a big clue that can be used for sequence alignment, and hence tracking of object motion is important. We use two basic image processing techniques for motion tracking in this thesis – adaptive background subtraction and template matching.

2.8.1 Adaptive Background Subtraction

In this method a statistical model of the background scene is built and moving regions in a video sequence are determined by subtracting each image frame from the background model [3]. The statistical model is built from video frames where no motion occurs. For each image point a measure of mean and variance in pixel intensity is computed. This statistical model accommodates small variations in light intensity which maybe caused by shadows and flicker of light source.

When a frame that has motion in it is subtracted from this adaptive background model, the difference in pixel intensity between the model and frame of interest

is compared to the variance of the background model. Only motion that results in pixel intensity changes greater than the model variance is considered as extracted motion.

2.8.2 Template Matching

Template Matching is another technique used for object tracking [22]. In template matching, a template image of the object being tracked is explicitly specified and the algorithm searches for this template image in the video sequence. The tracking of the template to subsequent video frames can be implemented as a neighborhood search around the last known position of the template match.

2.9 Motion Modeling

Motion modeling has been used in the past for a myriad of applications, such as video summarization [78], video compression [30], indexing and retrieval [12][2] [68], motion estimation [35][80], video segmentation [51] and action recognition [41][9] . These models can be loosely categorized into parametric and non-parametric (structural) models. This categorization is based on the premise that parametric models allow direct interpolation or extrapolation of motion trajectories based on the derived model parameters. Non-parametric or structural models allow segmentation and clustering of motion but cannot be used directly for interpolation or extrapolation of motion trajectories.

2.9.1 Parametric Motion Models

Parametric motion models [78]-[30] represent motion in terms of a parametric equation. For example, the x and y coordinates of a trajectory can be represented as a linear function of time. Other functions such as the Fourier basis, radial basis function, non-uniform B-splines, Wavelet basis, bilinear transformation and higher

order polynomial transforms, can also be used to represent motion. Once the basis vectors to be used in the parametric models have been decided, the coefficients of these basis vectors can be computed using methods such as Bayesian networks, learning theory, condensation algorithm and more popularly – linear least squares. Parametric models are scalable in the number of parameters that are required for the model; hence variations in motion type, computation cost and reshaping of regions can be dealt with. Also, appropriate approximations of the dominant or apparent motion can be made at considerably lower computation costs [10]. For example, Davis *et al.*[9] describe an approach to represent oscillatory motions by three parameterized sine-wave generators. Thus the basis vectors for their model are sinusoids, and a FFT analysis is computed to calculate the amplitude and phase parameters of the model.

2.9.2 Non-parametric Motion Models

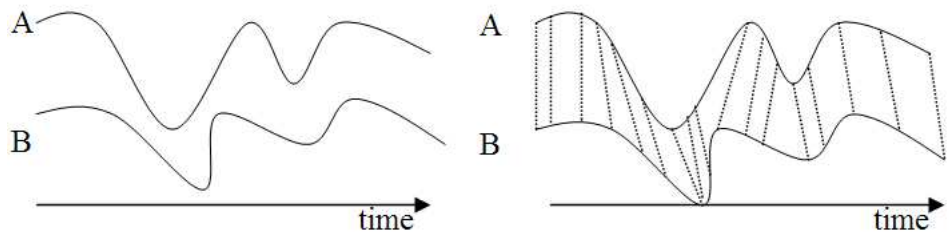
Non-parametric or structural motion models are based on analysis and recognition via templates, prototypes or training data [13]. Genc *et al.* [20] apply morphing, which is widely used in computer animation, to interpolate trajectories of motion. Some other non-parametric models include motion textures [44] (which use energy distribution of motion vectors to derive multidimensional vectors representing motion) and event based analysis [88] (which uses stochastic properties of events without prior knowledge of the events or their temporal scale to segment video sequences). Zelnik-Manor *et al.* [88] compute a statistical distance measure based on the histograms of the spatio-temporal gradients at various temporal scales to cluster and recognize video events. It is theorized that a temporal event is always characterized by the same temporal scales in all sequences. However, events when repeated by different or even the same individual may or may not be on the same temporal scale. For example, an individual throwing a ball repeatedly may do so at

different speeds, varying non-linearly with time, but the overall event still remains "throwing a ball". Another drawback of restricting the event model to a constant temporal scale is that the classes of actions that can be discriminated must be quite varied in terms of their speed and orientation. However, we use event properties for temporal registration and demonstrate why dominant motion descriptors are more suitable for sub-temporal registration than local warping strategies.

2.10 Dynamic Time Warping

Dynamic time warping (DTW) is a standard method used in speech recognition for adjusting the temporal duration of utterances. For example, if the word "dynamic" was uttered slowly and fast, the two speech patterns could be aligned based on DTW. Indeed any data which can be turned into a linear representation can be analyzed with DTW. In Fig. 2.9, we show two 1D signals (A and B) that have an overall similar shape but vary in the temporal alignment. DTW can be used to warp the time axis of these signals before a distance metric is used to compute the similarity between the signals. However, DTW has two major drawbacks. It can map a single point onto a large subsection of another time series thus leading to 'singularities'. In addition, it may fail to find an obvious alignment between two series because a feature in the first is slightly higher in magnitude than its corresponding feature in the other series.

DTW is implemented by computing the cost of mapping a point in one series to all the points in another series. Thus, for two series of length n and m, a $n \times m$ cost matrix (D) is computed. Then from the (n, m) cell of matrix D, a reverse path is traversed, such that the accumulated cost of the path is minimum. This traversed path represents the mapping computed between series 1 and series 2. DTW can be implemented in both symmetric and asymmetric form. If the implementation is symmetric, both the reference and test sequences are given equal weight in com-



(a) (b)

Figure 2.9: Illustration of dynamic time warping of two 1D signals A and B. (a) The two signals which have the same overall shape, but vary in the temporal duration of some sections of the signal. (b) The alignment computed by DTW.

puting the warping path. An illustrative equation of symmetric warping is shown below:

$$W(n,m) = min \begin{pmatrix} D(n,m) + W(n-1,m) \\ D(n,m) + W(n-1,m-1) \\ D(n,m) + W(n,m-1) \end{pmatrix}, \tag{2.19}$$

where n refers to Time-Series 1, m refers to Time-Series 2, W is the accumulated cost of warping and D is the cost of matching the current cell. However, if one sequence is considered more accurate than the other, then the warping can be unequally weighted towards it and such an implementation is asymmetric in nature. Compare Eq. 2.19 to one possible asymmetric implementation of DTW shown in Eq. 2.20, which is 'biased' towards time-series1 and is also penalizing a linear warping:

$$W(n,m) = min \begin{pmatrix} W(n-1,m) \\ 2D(n,m) + W(n-1,m-1) \\ D(n,m) + W(n,m-1) \end{pmatrix}. \tag{2.20}$$

Consider Eq. 2.20, in traversing the minimum accumulated cost path, when the next step to cell (n,m) is being evaluated, the path from $(n-1,m)$ cell does not add the cost $D(n,m)$ (the cost of mapping the n^{th} series-1 point to the m^{th} series-2 point) to the accumulated cost $W(n,m)$. However, the path from $(n-1,m-1)$, doubles the cost of $D(n,m)$ in addition to the accumulated cost till $(n-1,m-1)$. Thus the linear path from $(n-1,m-1)$ to (n,m) is penalized higher than the other paths.

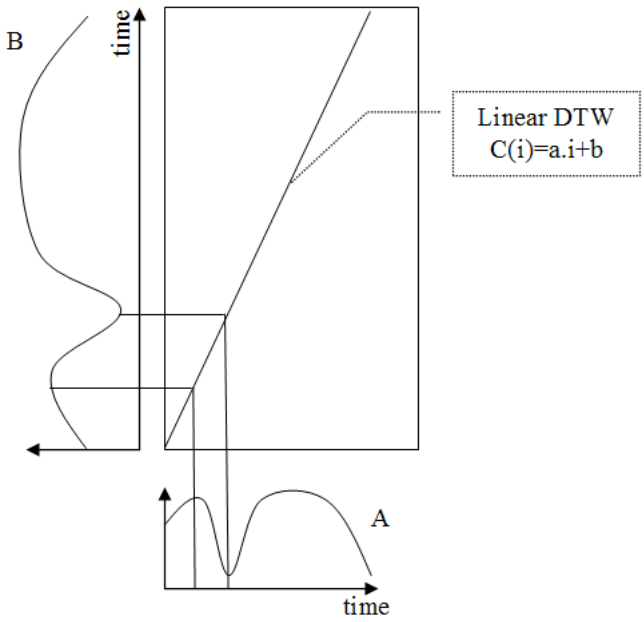


Figure 2.10: Dynamic Time Warping of two signals A and B. (Most algorithms assume a linear warp.)

2.11 Linear Regression

In this section we present a review of weighted least squares regression and three goodness of fit statistics that we have used in our work for modeling event dynamics. Regression methods such as least squares have been used in SR for estimating the value of HR pixels from Eq. 2.13. We use linear regression to estimate the event model as a function of the motion trajectory of the object in LR images. Regression is the method of estimating real valued functions from noisy samples. The estimated functions then provide a mapping between a systems input(s) and output(s). A regression model will select the best approximating function while minimizing a measure which is called (synonymously) risk or discrepancy or cost. A commonly used loss function is the squared error between the observed value and estimated value of the output. The weighted least squares regression minimizes the error estimate using the following equation:

$$SSE = \sum_{i=1}^{n} w_i (c_i - \hat{c}_i)^2, \tag{2.21}$$

where c_i is the i^{th} trajectory point of the n trajectory points available, $\hat{c}_i$ is the trajectory point computed based on the least squares fit and w_i are the weights assigned to each trajectory point.

In the bi-square weighted method, the weight given to each data point depends on how far the point is from the fitted line. Points near the line get full weight while points farther from the line get a reduced weight. This approach minimizes the effect of outliers on the results of the regression analysis. To assess if a regression model is appropriate, goodness of fit statistics such as sum of squared error (SSE), R-square and root mean square error (RMSE) are computed. The SSE measure is defined in Eq. 3.1; a value closer to 0 indicates a better fit. The R-square statistic measures how successful the fit is in explaining the variation in the data and it is the square of the correlation between the trajectory point and the predicted trajectory point. For sum of squares about the mean $\bar{c}$, SST (total sum of squares) is defined as:

$$SST = \sum_{i=1}^{n} w_i (c_i - \bar{c}_i)^2. \tag{2.22}$$

Then, the R-square statistic can be defined as:

$$Rsquare = 1 - \frac{SSE}{SST}. \tag{2.23}$$

An R-square value closer to 1 indicates that a greater proportion of the variation in the data has been accounted for. The root mean square error is then defined as:

$$RMSE = \sqrt{\frac{SSE}{\nu}}. \tag{2.24}$$

The RMSE statistic estimates the standard deviation of the randomness or error in the data. An RMSE value closer to zero indicates a better fit. If n is the number of trajectory points acquired and m is the number of fitted coefficients estimated from n, then the residual degrees of freedom are defined as $\nu = n - m$.

2.12 Recurrent Non-uniform Sampling

Recurrent non-uniform sampling (RNUS) is used to describe the sampling strategy where a signal is sampled below it's Nyquist rate, but multiple such sample sets offset by a time delay are available, *i.e.*, the sampling frequency is fixed, but the sampling time is randomly initialized. Figure 2.11 illustrates such recurrent non-uniform sampling, where $x(t)$ is a 1D continuous time signal which is sampled at a sampling rate of T, giving rise to samples at $T, 2T..MT$. Another sample set is also acquired at a sampling rate of T, however, this sample set is offset by a timing offset τ. Most RNUS algorithms assume that the value of τ is known *a priori* (*i.e.*, it is a controlled variable) or its exact value can be computed.

In RNUS reconstruction, a signal is reconstructed from multiple sample sets which are offset from each other by a *known* time interval [45]. The non-uniform sampling theorem [38] states that "a bandlimited function $x(t)$ can be uniquely reconstructed from a set of samples which are non-uniformly spaced but satisfy the condition that there be precisely N distinct samples to every interval of length NT, where N is some finite integer and T is the sampling period." An interpolation formula for reconstruction of the signal from its non-uniform samples is provided in [19]. In practice however, we neither encounter bandlimited signals nor do we have infinite samples. Therefore, the interpolation formula has been simplified and adapted for finite data reconstruction, with reduced complexity. A review of some of these approximations is presented in [45]. Another approach to reconstruction from RNUS was presented by Feichtinger *et al.* [17], where the sampling atoms or synthesis functions are computed using approximation operators such that every bandlimited function $x(t)$ has a stable summed expansion of the type shown below:

$$x = \sum_{n \in \mathcal{Z}} x(\tau_n) e_n, \tag{2.25}$$

where e_n are the sampling atoms. Unlike uniform sampling functions the sampling

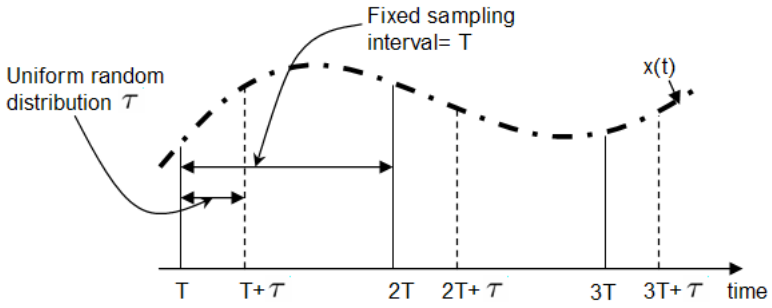


Figure 2.11: Illustration of recurrent non-uniform sampling with two sample sets.

atoms are not necessarily translations of a mother function and decay rapidly such that the reconstructed signal value is computed using adjacent local samples.

In the next sections we review how RNUS is formulated and how an approximate high resolution signal can be reconstructed from non-uniform samples.

2.12.1 Sampling formulation

Suppose $x(t)$, without loss of generality a 1D signal, is sampled at a rate $f_{sL}(T = 1/f_{sL})$, such that $f_{sL} < f_{Nyquist}$, to generate a sub-sampled signal S_i. If N such sample sets are available (*i.e.*, $1 \leq i \leq N$), each sampled at a rate f_{sL} , and sampling instances are placed such that the combined sampling rate equals Nf_{sL}, then if $Nf_{sL} > f_{Nyquist}$, $x(t)$ can be reconstructed from its non-uniform samples. This strategy is often used in interleaved analog to digital converters [67]. Let the n^{th} sample set acquired be represented as follows:

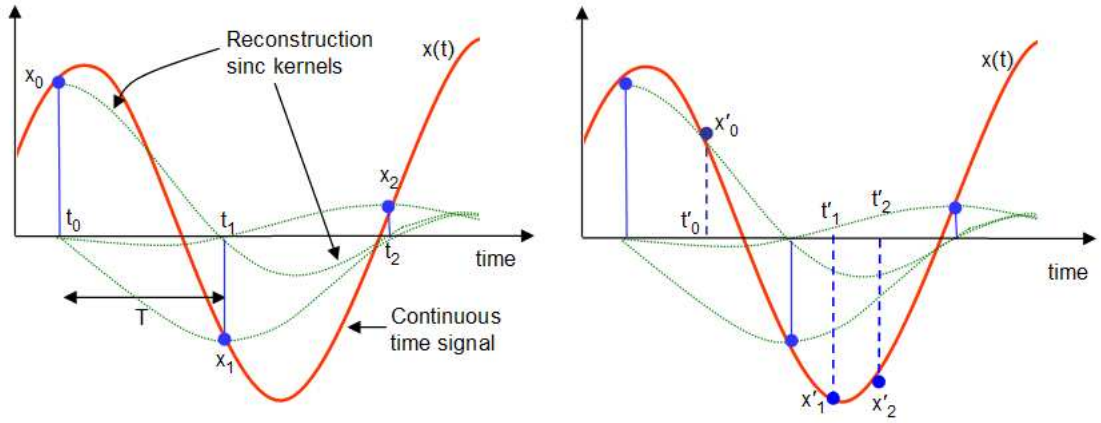
$$S_n = x(kT + \frac{(n-1)T}{N} + \tau_n), \tag{2.26}$$

where $1 \leq k \leq M$, and τ_n is the n^{th} random temporal offset.

2.12.2 Reconstruction formulation

Direct reconstruction of a continuous signal from its N non-uniformly sampled sequences [66] can be done as follows:

$$x(t) = \sum_{n=1}^{N} \sum_{k \in \mathcal{Z}} x(kT + \triangle_n)\phi_n(t - (kT + \triangle_n)), \tag{2.27}$$



(a) (b)

Figure 2.12: (a) Reconstruction from uniform samples using sinc kernels. (b) Illustration of non-uniform samples which can be expressed as linear combinations of samples from (a).

where $\triangle_n = \frac{(n-1)T}{M} + \tau_n$ and ϕ_n represents the reconstruction kernels such as splines, Lagrange's polynomials and the cardinal series. An indirect approach to reconstruction from RNUS is to derive uniformly separated samples from the non-uniform signal instances, and then reconstruct using the standard interpolation formula in Eq. 2.28. Suppose a bandlimited signal $x(t)$, is sampled at the Nyquist rate to obtain uniform samples $x(kT)$, $x(t)$ can be reconstructed from the samples using the interpolation formula:

$$x(t) = \sum_{k=-\infty}^{+\infty} x(kT) \frac{sin\Omega(t-kT)/2}{\Omega(t-kT)/2}, T = 2\pi/\Omega. \tag{2.28}$$

Let x_0, x_1 and x_2 correspond to three discrete samples of $x(t)$ taken with a uniform time interval at time t_0, t_1, and t_2 (see Fig. 2.12(a)). Assuming a finite window of reconstruction (instead of the infinite samples in Eq. 2.28), an approximate reconstructed signal can be computed as:

$$\hat{x}(t) = x_0 sinc(t-t_0) + x_1 sinc(t-t_1) + x_2 sinc(t-t_2). \tag{2.29}$$

If $x(t)$ was also sampled at non-uniform time instances t'_0, t'_1 and t'_2, as shown in Fig. 2.12(b), by substituting t with t'_i $(0 \leq i \leq 2)$ in Eq. 2.29 we can write the

following linear equations:

$$\begin{aligned} x(t'_i) &= x_0 sinc(t'_i - t_0) + x_1 sinc(t'_i - t_1) \\ &\quad + x_2 sinc(t'_i - t_2) : 0 \leq i \leq 2, \end{aligned} \tag{2.30}$$

where $x(t'_i)$ are the known non-uniform samples. Equation 2.30 can be expressed as a system of linear equations:

$$B = A \cdot \mathbf{x}, \tag{2.31}$$

where $A = \begin{bmatrix} sinc(t'_0 - t_0) & sinc(t'_0 - t_1) & sinc(t'_0 - t_2) \\ sinc(t'_1 - t_0) & sinc(t'_1 - t_1) & sinc(t'_1 - t_2) \\ sinc(t'_2 - t_0) & sinc(t'_2 - t_1) & sinc(t'_2 - t_2) \end{bmatrix}$, $B = \begin{bmatrix} x'_0, x'_1, x'_2 \end{bmatrix}^T$ and $\mathbf{x} = \begin{bmatrix} x_0, x_1, x_2 \end{bmatrix}^T$.

These linear equations can be solved using standard methods (such as LU decomposition [53]) to calculate the sample values at the uniform sampling instances ($\mathbf{x}$ in Eq. 2.31). By plugging the solution of $\mathbf{x}$ (sample values at uniform instances of time) into Eq. 2.29, approximate reconstruction of the original signal can be done.

2.13 Summary

In this chapter we reviewed current state-of-the-art in super-resolution imaging. We presented a chronological development of methods in this area and examined the limitations and drawbacks of these techniques. We reviewed methods in video synchronization and dynamic time warping. Algorithms for feature extraction, spatial registration and recurrent non-uniform sampling were also reviewed. We will refer back to these algorithms in subsequent chapters.

Chapter 3

Event Models as a Tool for SR Imaging at Low Frame Rates

In this chapter we present our first study on improving the temporal registration of low resolution (LR), low frame rate, video sequences for super-resolution (SR) imaging. We assume a simplified scenario where: (i) the sequences are acquired from the same scene, (ii) the view point or camera position does not change, and (iii) the video sequences have a fixed temporal offset between them. We investigate the effects of low frame rates on SR reconstruction from such video sequences. Contemporary approaches ignore the global dynamics of the scene and use a linear interpolation model for alignment of LR sequences. We have used event dynamics, a property that is inherent to an event and is thus common to all acquisitions of the event, to integrate multiple low-temporal resolution acquisitions to generate high-temporal resolution data.

This rest of this chapter is organized as follows. In Section 3.1 we briefly review some contemporary registration algorithms and introduce the problem statement. We present a systems overview of our proposed method in Section 3.2. In Section 3.2.1 we discuss our algorithm for matching video sequences based on event dynamics. We also present a comparative analysis of a contemporary algorithm proposed by Caspi *et al.* [5][6] and submit a case study where their algorithm gives erroneous results. In Section 3.3 we present our experimental setup and results

with real and synthetic data. The results are compared with those obtained with the Caspi algorithm on the same data. In Section 3.4 we present a novel application of our work in registering and generating high resolution MRI sequences for medical data visualization, as well as, the generation of high temporal resolution videos. We briefly discuss the relation between proposed work and the sampling theorem in Section 3.5. In Section 3.6 we put forward the conclusion of this study.

3.1 Introduction to Event Models

All digital data acquisition techniques essentially acquire discrete samples of a fundamentally continuous world. Often, limited by current technology, we can only collect and process a limited amount of information from the events that occur around us. For example, generic video cameras can capture information at frame rates of 30 frames per second (fps), whereas a fast magnetic resonance imaging (MRI) protocol can capture only 6-7 fps of soft tissue motion in the human body. In order to recreate the true event that occurred, most researchers use multiple acquisition sources and offset their capture so that correlated but discrete samples are acquired. In this chapter, we address the problem associated with a scenario where only a single acquisition source can be used at a given time. To solve this issue, one approach is to reconstruct information by interpolating between low-resolution discrete samples available in a single set of data. Interpolation is a poor solution when few samples are present, as the interpolated data may be completely inaccurate when compared to the real event. The unique approach we propose is to use repetitions of the same event (which may or may not be replicable each time) and use event dynamics to temporally register and reconstruct the samples in a multidimensional space. Temporal registration of the LR sequences is used to generate a high-resolution data set, which can be used further for applications such as robust object tracking, super-high resolution video generation or medical image visualiza-

tion.

In the past, temporal registration has been achieved by using either specific timestamp information from an external source or information derived directly from the video data by identifying key points. In both these cases, global and local misalignments are computed by minimizing a matching criterion. While global alignment allows the video sequences to be roughly registered, correction of local misalignment is essential as it allows sub-frame temporal registration. Current techniques for correcting local misalignment are based on linear interpolation or splines. Linear interpolation involves connecting discrete trajectory points with straight lines. As the line must pass through the trajectory points, linear interpolation leads to exact fitting. If the trajectory points were not accurately extracted, which can often be the case if the image is noisy or features change due to motion in the scene, the interpolated lines will also be an inaccurate representation of the trajectory motion. Thus, if feature extraction or coordinate selection is not robust, error is introduced into the registration. Splines on the other hand require accurate determination of control points and knots. These control points and knots are computed from a large solution space, making Splines computationally expensive. Also, more than one combination of control points and knots can be used to interpolate between the trajectory points, thus leading to a non-unique registration.

Video sequences are typically used when the aim is to capture event related motion in the scene. In this thesis we propose to use event dynamics, a property that is inherent to an event and is thus common to all acquisitions of the event, to integrate multiple low-temporal resolution acquisitions to generate high-temporal resolution data. First let us define an 'event'. An event can be defined as the occurrence of something important at a certain spatial location, over an interval of time. Thus, the two parameters important in defining an event are its spatial location and its temporal range. It is our hypothesis that any event in multidimensional

space will generate distinct spatiotemporal patterns, and while these patterns may not be identical over multiple repetitions, they will have a high degree of correlation. Space-time patterns of image elements (such as single pixels or regions of an image) can be represented as motion models of the elements. This will allow the parameterization of events occurring in spatio-temporal space. Low-resolution capture limits us to sample only a few representative space-time points of these motion patterns. However, if sufficient spatiotemporal patterns of repeated instances of the same event are available, then the low-resolution samples can be used to generate a high-resolution space-time pattern of the event.

3.2 System Overview

A system overview of the proposed technique for temporal alignment of LR videos of the same scene is shown in Fig. 3.1. In Fig. 3.1 we assume that (i) multiple LR videos are available as input to the spatio-temporal alignment module, (ii) the LR videos are from the same scene, and (iii) the spatial alignment between them can be represented as a linear conformal transformation[1]. Figure 3.1 shows three system modules: (i) Spatial alignment, (ii) Temporal Alignment, and (iii) Reconstruct. In the spatial alignment module, feature are extracted and tracked through the video sequences. These features are considered to be discrete samples from the continuous motion in the sequences. Standard algorithms are used to compute the spatial alignment between the sequences, these have been reviewed in Section 2.7. The temporal alignment module is implemented in two stages: (i) Generation of a weighted bi-square regression based parametric model of the dominant motion and (ii) Minimization of the alignment error of the parametric models. In this chapter we focus on temporal alignment, the weighted bi-square regression based

[1]For MRI data, this spatial misalignment is caused by slight movement of the subject between scans and for real video data used in this part of our work the LR videos are generated as subsampled versions of a high resolution video hence there is no spatial alignment required.

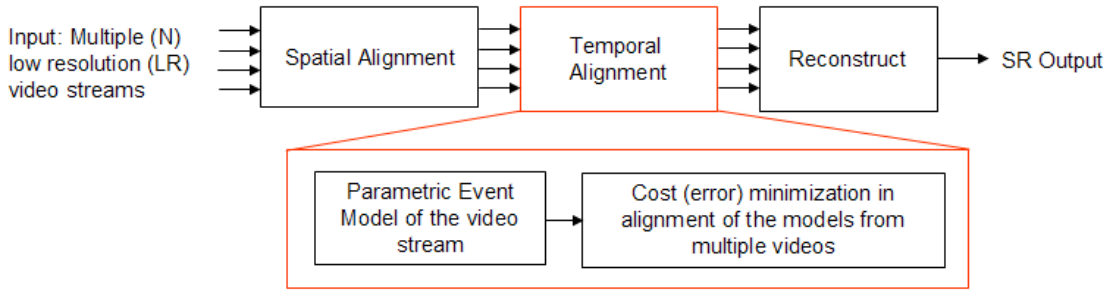


Figure 3.1: Systems overview of proposed technique for temporal alignment of videos of the same scene

parametric model is discussed in detail in Section 3.2.1 and the minimization of alignment error is discussed in Section 3.2.2. The last module in Fig. 3.1 represents the reconstruction algorithm used to generate the super-resolved data. Various SR reconstruction algorithms have been reviewed in detail in Chapter 2. For MRI data the reconstruction module of the system is implemented in the frequency domain, and differs from the methods reviewed in Chapter 2. The MRI reconstruction algorithm is presented in Chapter 6, Section 6.4.4.

3.2.1 Regression Model for Temporal Alignment

Regression models (reviewed in Section 2.11) have been used in the past for motion modeling in two contexts: (i) to predict the position of a feature point between two acquired frames (*i.e.* interpolate between two acquired feature locations), and (ii) to extrapolate the position of an feature point for a frame in the future, *i.e.* predict position of a feature in a frame that has not been acquired yet. In this work we are concerned with the first context – interpolating missing data between acquired frames. In our method, we do not consider individual trajectory points, but rather look at the entire dynamics that those points represent for alignment. Let the discrete points sampled on the two trajectories be represented as $c(x, y, n)$ and $c'(x', y', n')$, where x, y represent the spatial position and n represents the frame number or temporal position of the feature point. Since we are focussing on tem-

poral alignment only, we assume that the trajectories have been previously spatially aligned and can therefore be represented as $c(x, y, n)$ and $c'(x, y, n')$. Note that after spatial registration, the spatial coordinates x', y' in the second sequence can be represented as x, y.

We approximate the real world continuous event dynamics by applying a weighted bi-square linear regression model on the discrete samples. The weighted bi-square linear regression (WBLR) model is presented in detail in Appendix A. Intuitively, the WBLR model works as follows. Given a set of discrete feature points, the WBLR model attempts to compute model parameters which can be used to generate continuous trajectories. The algorithm assigns a weight to each discrete feature point. These weights are used to assign importance to the feature points. For example, if a feature in the trajectory was incorrectly extracted, it is desired that the impact of this feature on the continuous trajectory should be minimized. Hence it will be assigned a low or zero value. Since we cannot pre-determine if a feature has been inaccurately extracted, these weights are unknown. The WBLR algorithm initializes the weights to 1. In this work, we start with the assumption that the model is linear, and using the discrete feature points, we compute the parameters for a linear fit to the data. The difference between the actual feature points and the estimated feature points is then computed as follows:

$$SSE = \sum_{i=1}^{n} w_i (c_i - \hat{c}_i)^2 \tag{3.1}$$

where c_i is the i^{th} trajectory point of the n trajectory points available, $\hat{c}_i$ is the trajectory point computed based on the least squares fit and w_i are the weights assigned to each trajectory point. The WBLR algorithm then adjusts the values of the weights to try and minimize the sum of squared error (SSE) in Eq. 3.1. The algorithm iterates over computing model parameters, SSE and weight update till the SSE converges. We then increase the model parameters to a quadratic, higher

order polynomial and exponential fit and reapply the WBLR algorithm for each update to the model. Model parameters that result in the least residual (SSE) and best goodness of fit statistics is chosen to represent the trajectory. The following goodness of fit statistics have been used in this work: sum of squared error (SSE, Eq. 3.1), R-square and root mean square error (RMSE).

The R-square statistic measures how successful the fit is in explaining the variation in the data and it is the square of the correlation between the trajectory point and the predicted trajectory point. For sum of squares about the mean $\overline{c}$, SST (total sum of squares) is defined as:

$$SST = \sum_{i=1}^{n} w_i (c_i - \overline{c}_i)^2. \tag{3.2}$$

Then, the R-square statistic can be defined as:

$$Rsquare = 1 - \frac{SSE}{SST}. \tag{3.3}$$

An R-square value closer to 1 indicates that a greater proportion of the variation in the data has been accounted for. The root mean square error is then defined as:

$$RMSE = \sqrt{\frac{SSE}{\nu}}. \tag{3.4}$$

The RMSE statistic estimates the standard deviation of the randomness or error in the data. An RMSE value closer to zero indicates a better fit. If n is the number of trajectory points acquired and m is the number of fitted coefficients estimated from n, then the residual degrees of freedom are defined as $\nu = n - m$.

The events represented in our real sequence experiments are ball throwing and swinging a ball tied to a string. These events can be characterized in spatio-temporal volume by a sinusoidal pattern and spiral pattern respectively. A spiral pattern on decomposition can also be viewed as a sinusoid with slight variations in the structural parameters. Figure 3.2 illustrates the decomposition of a spiral pattern into a linear translatory and a sinusoidal pattern. Variations are also added to these

patterns by the linear translatory motion of the ball. A sinusoid pattern can be approximated by the series:

$$sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + ... \tag{3.5}$$

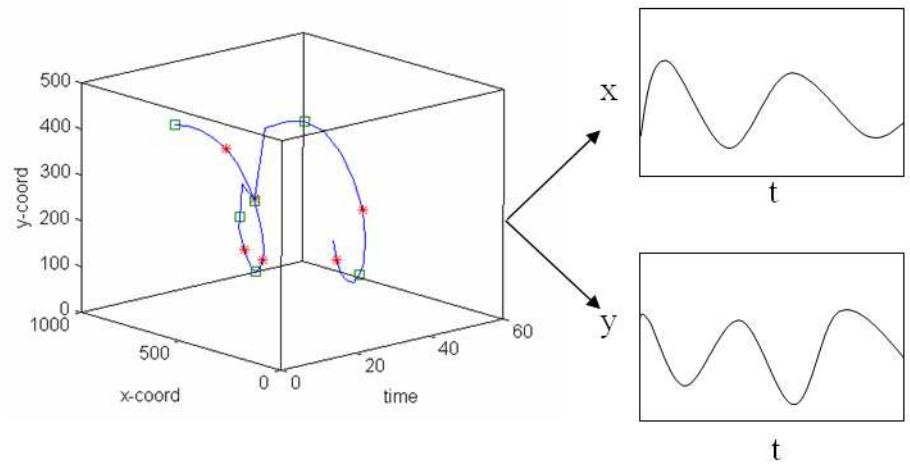
This series converges quickly, and a series order of 7∼8 is sufficient to represent the underlying trend of the motion of the ball. Thus the upper bound of our parametric model can be set to a polynomial of degree 8, which reduces our search space to a more manageable one. Note that this parametric model is global and we do not compute parametric models locally. For local parametric models the search space would be large, like the search space of splines.

3.2.2 Minimization of Alignment Error

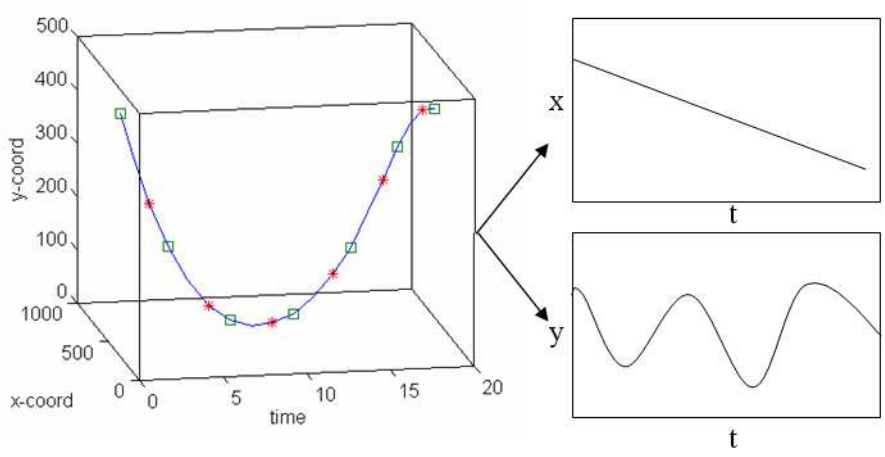
We chose the weighted (bi-square weights) least squares algorithm for our regression model since it optimizes the generation of a continuous trajectory by factoring in inaccuracies in feature extraction. Subsequent to linear regression computed by minimizing the residual error (Eq. 3.1), the trajectories can be represented as continuous curves $c(x, y, t)$ and $c'(x, y, t')$, where the temporal variable of the second sequence t' is related to the temporal variable of the first sequence t as: $t' = t + \Delta t$. Δt is a single offset value that relates all corresponding frames in one sequence to the other. Temporal registration of the two trajectories (thus the video sequences) involves finding a sub-frame displacement Δt that minimizes the distance between the coordinate positions in the two continuous event models as follows:

$$C = \arg\min_{\Delta t} \sum_{t} [(c(x,t) - c'(x,t')^2 + (c(y,t) - c'(y,t')^2]. \tag{3.6}$$

We did not use fitting with splines as they depend greatly on the accuracy of the control points. Since our trajectories are from multiple events, we want to reduce the dependence of the algorithm on local points or features, which may or may not



(a)



(b)

Figure 3.2: Decomposition of motion patterns into constituent x and y motion. (a) Spiral motion is decomposed into constituent x and y oscillatory patterns; (b) ball-throw motion is decomposed into linear motion along the horizontal coordinate and oscillatory motion along the vertical coordinate.

have been accurately obtained from the multiple trajectories. We believe that the overall trend of the event will be a better global representation of the trajectory.

3.2.3 Caspi Model for Temporal Alignment

In Chapter 2 we reviewed various approaches for computing temporal alignment between two sequences. Of the approaches reviewed, the algorithm proposed by Caspi *et al.* [5] is the most closely related to this work, since the problem addressed is sequence alignment for super-resolution reconstruction. We provide a quick overview of the Caspi algorithm in this section (Note that the equations from [5] have been adapted to reflect only a temporal misalignment). Temporal misalignment is defined to occur when two input sequences have a time-shift or offset between them, which could have been caused by different frame rates of the cameras or delay in activating the cameras. This temporal misalignment is modeled in [5] by a 1D affine transform as follows:

$$t' = s.t + \Delta t. \tag{3.7}$$

The optimum temporal alignment is computed by minimizing the following error function:

$$\overrightarrow{P} = \arg\min_{\Delta t} \sum_{trajectories} \left(\sum_{t \in Trajectory} ||p'(s.t + \Delta t) - p(t)||^2 \right), \tag{3.8}$$

where $p(t) = [x(t), y(t), t]$ is the spatial position of the feature point along the trajectory at point t in time. $p'(s.t + \Delta t)$ is the location of the corresponding feature point in the second sequence at time $t' = s.t + \Delta t$. Since t' is modeled as a subframe displacement, coordinate values at t' are linearly interpolated from the corresponding integer location $t_1 = \lfloor t' \rfloor$ and $t_2 = \lceil t' \rceil$. Error minimization is performed by computing Δt for the best linear interpolation value. The minimization is stopped when the residual error stops changing or when a given number of iterations are exceeded. A search is then performed for $\alpha = t' - t_1$ that minimizes the following

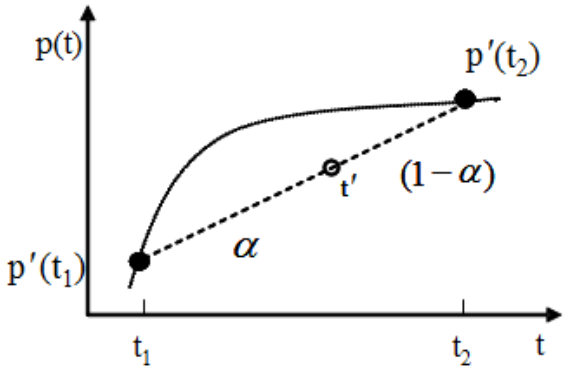


Figure 3.3: Pictorial representation of linear interpolation in Eq. 3.9 for sub-frame temporal alignment.

equation (see Fig. 3.3 for a pictorial representation):

$$\min_{\alpha} \sum_{t} ||p'(t_1).(1-\alpha) + p'(t_2).\alpha - p(t)|| : \alpha \in [0,1] \tag{3.9}$$

3.2.4 Comparative Analysis

Let us now consider a case of two trajectories $p(t)$ and $p'(t)$ as shown in Fig. 3.4, where $p'(t)$ is the same as trajectory $p(t)$, but offset by a subframe displacement 'Δt' These continuous trajectories are discretely sampled by an acquisition source at different points in time. We represent this by the circles on the trajectory. Using Caspi's algorithm $p'(t)$ will be interpolated for all subframe values from its value at the integer points, say $p'(t_1)$ and $p'(t_2)$ according to Eq. 3.9. Subsequent to linear interpolation, at some temporal displacement $\Delta\hat{t}$, $p'(t)$ will become equal to $p(t)$ and the algorithm will find a best match and stop (we have simplified the trajectory to that of a single coordinate, and only display two points on the trajectory). However, the true temporal displacement was 'Δt', which was not computed as the linear interpolation ignored the curvature of the trajectory. In cases where the sampling rate of the trajectory is high, linear interpolation will provide acceptable results. This is because, as the time between two samples begins to decrease, the trajectory becomes more linear. However, for poor temporal sampling rates, as

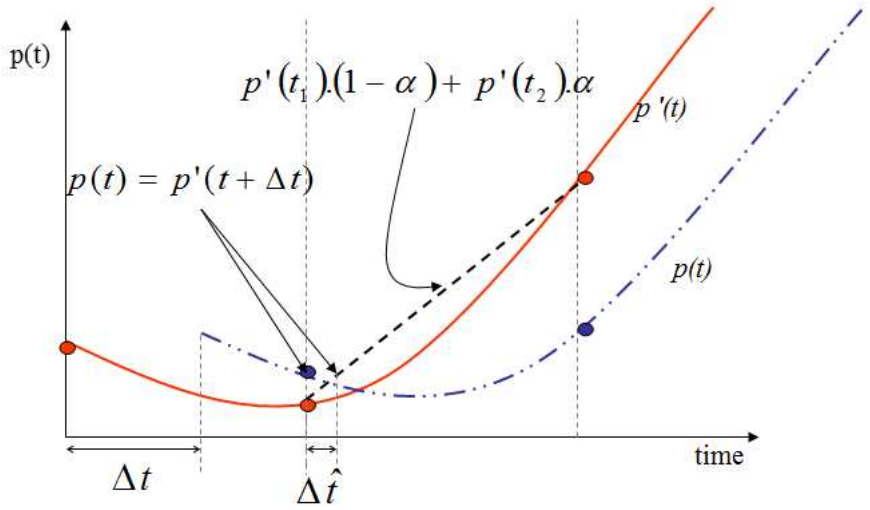


Figure 3.4: Case study of failure of linear interpolation for temporal alignment.

is the case with MRI, linear interpolation will often result in erroneous temporal offsets, as we show in our experiments.

3.3 Experimental Analysis

The system shown in Fig. 3.1 has been implemented in the MATLAB7.0 programming environment. We present our experimental analysis in the following two sections: synthetic trajectories and real video data.

3.3.1 Synthetic Data

Synthetic data was created by downsampling eight high temporal resolution trajectories of hyperbolic functions, such as $tanh$, $cosh$, $sinh$ and combinations of these functions. Some representative trajectories are shown in Fig. 3.5. Each high resolution trajectory was downsampled into two trajectories each. One of these two downsampled trajectories was also offset by a known temporal difference in order to create temporal misalignments. Hence, for these cases the true temporal alignment was also known a priori. Entire trajectories as well as sub-segments of trajectories were matched using both the methods. The results of the temporal alignment

Table 3.1: Results of Temporal Alignment (TA) with Synthetic Data for Linear Interpolation (LI) and Event Dynamics (ED) based matching. LI corresponds to Caspi Algorithm and ED corresponds to Proposed Algorithm. Unit of error is 'frames'.

Traj	Actual TA	TA by LI	Error with LI	TA with ED	Error with ED
Traj 1	13	6	7	12.25	0.75
	13	7.25	5.75	13.5	0.5
	13	6	7	13.5	0.5
Traj 2	13	6	7	16	3
	13	6	7	14.75	1.75
Traj 3	10	12	2	11	1
	10	6	4	11	1
Traj 4	10	12	2	11	1
	10	6	4	10	0
Traj 5	10	12	2	11	1
	10	6	4	10	0
Traj 6	10	6	4	9	1
	10	6	4	10	0
Traj 7	10	12	2	10	0
	10	13	3	11	1
	10	18	8	14	4
Traj 8	10	13	3	12	2
Average Error in frames			4.7142		0.8571

are shown in Table 3.1. It can be seen from Table 3.1 that the proposed method has an average error of 0.86 frames, compared to 4.7 frames for the Caspi model. The maximum error for the Caspi model was 9.25 frames, while the error with our method was 0.75 frames for the same trajectory. The maximum error in our model was 4 frames, while the corresponding error for the Caspi model was twice as high at 8 frames. An analysis of the relation between error and complexity of trajectory reveals that complex trajectory (or complex motion) will result in higher error in registration, see trajectory-7 in Fig. 3.5 and its corresponding error value in Table 3.1.

3.3.2 Noise Analysis

We also performed analysis on synthetic data to determine the robustness of both Linear Interpolation (Caspi Algorithm) and Event Dynamics (proposed) methods to

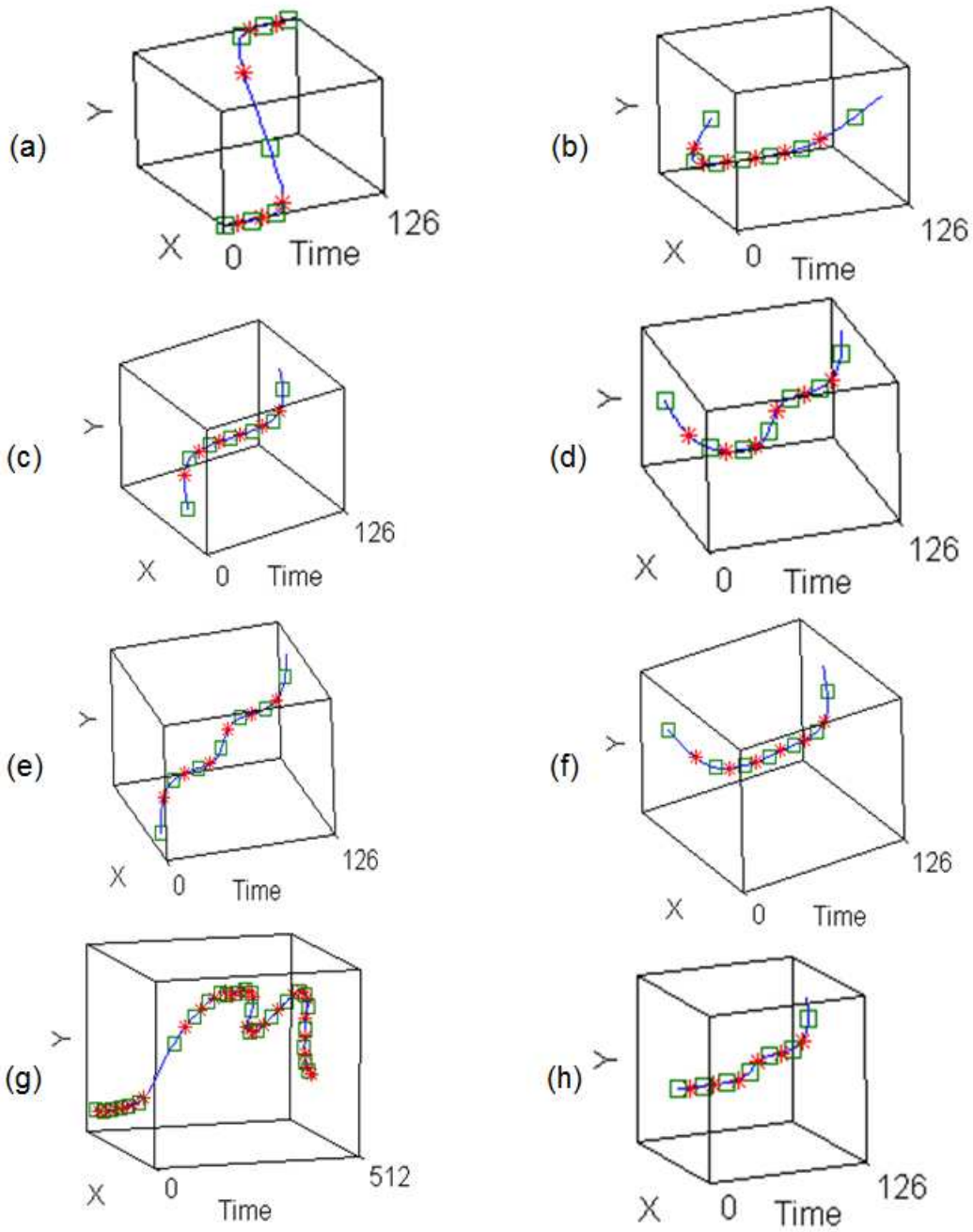


Figure 3.5: (a)-(h) Sample trajectories from synthetic data. (a) Traj 1, (b) Traj 2, (c) Traj 3, (d) Traj 4, (e) Traj 5, (f) Traj 6, (g) Traj 7, and (h) Traj 8.

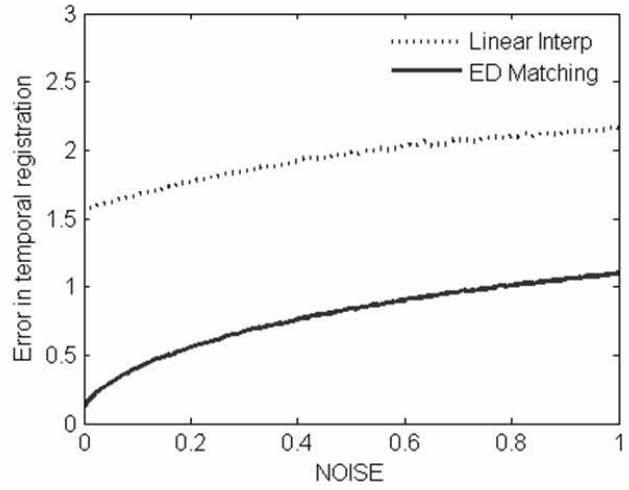


Figure 3.6: Plot of Error in temporal registration vs. variance of noise added to the 50 real data trajectories for Linear Interpolation (Linear Interp) and event dynamics based matching (ED Matching).

noise which was randomly added to the trajectory locations. The trajectories were normalized using the following equation:

$$c_n(x_i, y_i, t_i) = \frac{c(x_i, y_i, t_i) - \mu.c(x, y, t)}{\sigma.c(x, y, t)} \tag{3.10}$$

where μ is the mean, and σ is the standard deviation of the trajectory. Pseudo-random noise with zero mean and variance between 0 and 1 was then added to the trajectory points, and temporal registration was performed on these noisy trajectories using both the Linear Interpolation and proposed methods. The experiments were iterated 1000 times, and the average error computed is shown in Fig. 3.6. The addition of noise leads to an increase in the error in both the methods as can be seen in Fig. 3.6. However, event dynamics based temporal registration consistently resulted in lower temporal registration errors compared to linear interpolation at all levels of noise.

3.3.3 Real Data

We captured multiple video sequences of throwing a ball and swinging a ball tied to the end of a string. These sequences were captured at 30fps and were downsampled

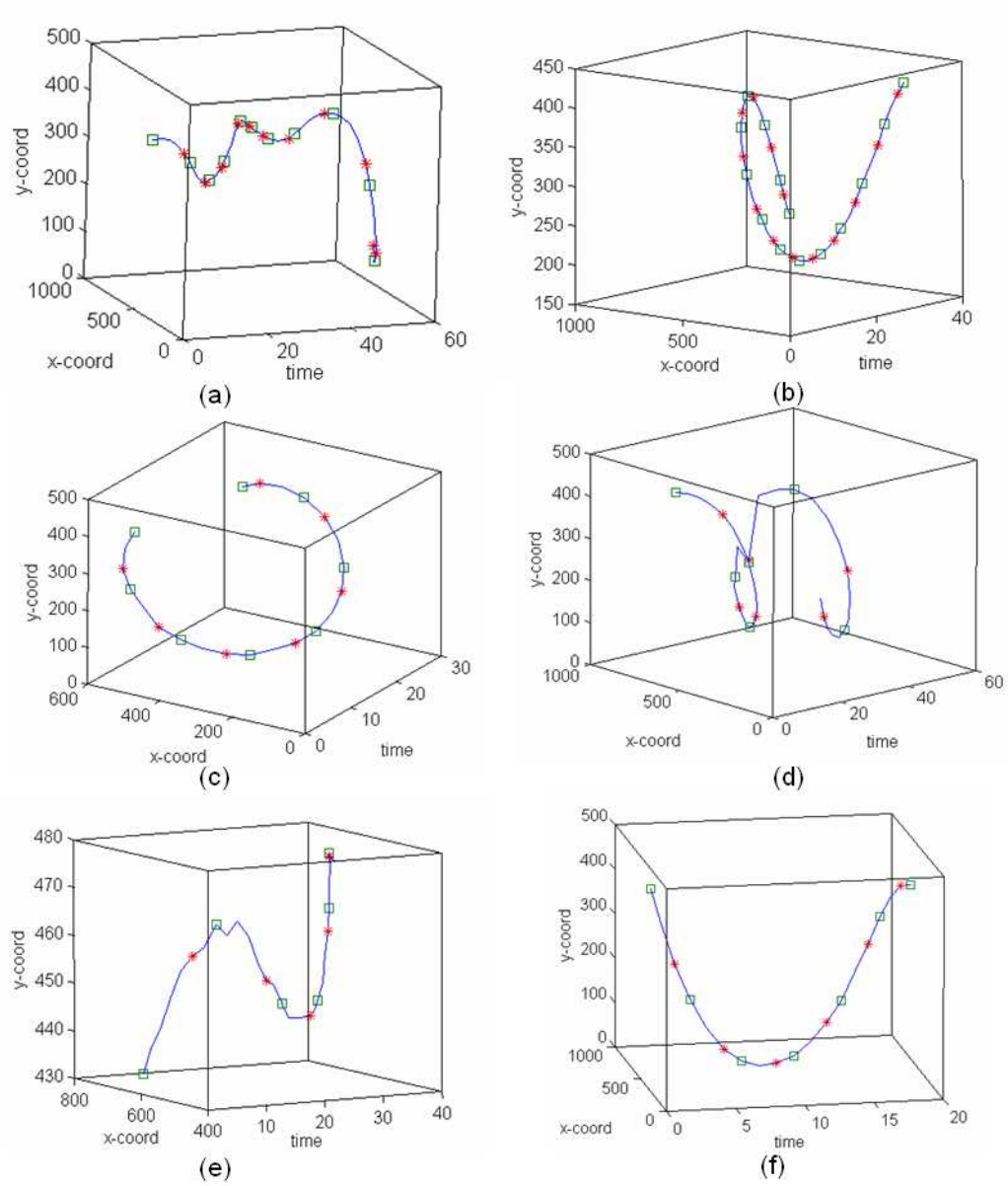


Figure 3.7: (a)-(f) Sample trajectories from the real data. (a) Throw 1, (b) Swing 1, (c) Swing 2, (d) Swing 3, (e) Throw 2 and (f) Throw 3.

into two sub-sequences each, with varying frame rates ranging from 15fps to 3.3fps. One of these two downsampled trajectories was also offset by a known temporal difference in order to create temporal misalignments. Hence, for these cases the true temporal alignment was known a priori. The moving ball was segmented using a color based scheme and the centroid of the segmented region was computed. Some representative centroid trajectories are shown in Fig. 3.7. Illustrative frames of these sequences have been compiled in Appendix B, Section B.1. These video sequences and the temporal registration result are also available online at: `www.ece.ualberta.ca/~meghna/ieeeMult/2006.html`.

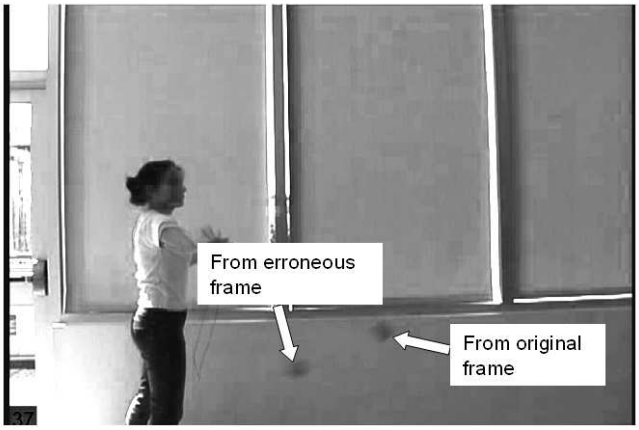
In Fig. 3.8 we show the extent of spatial error that can be caused by a few frames of error in the temporal registration. This figure consists of two frames transparently overlapped over each other. The position of the ball in the original sequence for that instant of time is indicated on the figure. It can be seen that linear interpolation has led to an incorrect alignment with a large spatial offset from the correct position. The results of temporal registration using both linear interpolation and proposed method are tabulated in Table 1. It can be seen that linear interpolation results in an average error of 2.27 frames and a maximum error of 4.95 frames in temporal offset calculation, while event dynamics based matching results in an average error of 0.105 frames and a maximum error of 0.5 frames. Also interesting to note is that the results for the 'throw' sequences are better than the results for the 'swing' sequences. This could be because the complexity of motion for swinging an object is much higher (3D motion) than that of a throw.

3.4 Applications

In this section we present two applications of our work, namely – medical visualization and generation of high temporal resolution videos. While our work can be applied to a myriad of regions of the body for medical visualization, in this thesis,

(a)

(b)



(c)

Figure 3.8: (a) Original frame at correct temporal alignment. (b) Incorrectly aligned frame corresponding to (a) and (c) Superimposed frames from the original sequence and temporally aligned sequence showing incorrect registration.

Table 3.2: Results of Temporal Alignment (TA) with Real Data for Linear Interpolation (LI) and Event Dynamics (ED) based matching. LI corresponds to Caspi Algorithm and ED corresponds to Proposed Algorithm. Unit of error is 'frames'.

Traj	Actual TA	TA by LI	Error with LI	TA with ED	Error with ED
	4	2.75	1.25	4	0
	4	2.85	1.15	4	0
Throw 1	5	3.8	1.2	4.85	0.15
	6	2.75	3.25	5.9	0.1
	7	2.05	4.95	6.95	0.05
	6	3.45	2.55	5.55	0.45
	7	2.05	4.95	6.95	0.05
Swing 1	4	2	2	4	0
	5	2	3	4.75	0.25
	4	2	2	4	0
	3	1.9	1.1	2.95	0.05
	3	1.9	1.1	2.8	0.2
Throw 2	4	2	2	4	0
	3	1.9	1.1	2.95	0.05
	4	2	2	4	0
	4	2.75	1.25	4.25	0.25
	5	1.75	3.25	5.5	0.5
Swing 2	5	2	3	5	0
	4	2	2	4	0
	4	2	2	4	0
	5	1.75	3.25	5.25	0.25
	5	1.75	3.25	5.25	0.25
Swing 3	4	2.5	1.5	4	0
	4	1.8	2.2	4.2	0.2
	4	2	2	4	0
	5	2	3	5	0
	5	1.75	3.25	5	0
Throw 3	4	2.75	1.25	3.75	0.25
	4	1.8	2.2	4	0
	3	1.9	1.1	3.1	0.1
		Average Error	2.27		0.105
		Max Error	4.95		0.5

we present one representative application in the visualization of the oral-pharyngeal region.

3.4.1 Medical Visualization

Current technology in medical visualization allows users to view 2D and 3D data. However, the temporal changes in the 2D images or 3D volumes are lost as no technique has high temporal resolution (2D+time/3D+time) acquisition capability. In order to obtain high temporal resolution, accuracy in the spatial domain has to be sacrificed, as there is a trade-off between spatial and temporal resolution. We applied the event dynamics technique to aid the diagnosis and treatment of dysphagia or swallowing disorders. [1].

Dysphagia can be caused by many factors such as a stroke, trauma to the cranio-facial region or tumors in the brain and oral-pharyngeal tract. The current method of assessing dysphagia is video-fluoroscopy, where the patient swallows a barium cookie and is subjected to X-rays during the swallow. This method of assessment has three major drawbacks: (i) the patient is subjected to harmful radiation, (ii) soft tissue information is lost, since X-rays pass through soft tissue, and (iii) 3D spatial information is lost, since all image planes that are perpendicular to the direction of the X-rays are accumulated together into a single image plane. Thus 3D depth information is lost. These limiting features of X-ray have prompted our work in using MRI to assess swallowing disorders. The central idea behind our work is to combine low resolution (LR) MRI swallow video sequences (acquired over multiple swallows) into a single high resolution swallow sequence. The event dynamics technique has been used to register the LR MRI sequences to sub-frame accuracy. Experimental details and performance results are presented below.

An overview of the method is presented in Fig. 3.9. In the Dysphagia experiment, MRI images of a patient are taken while swallowing small amounts of water.

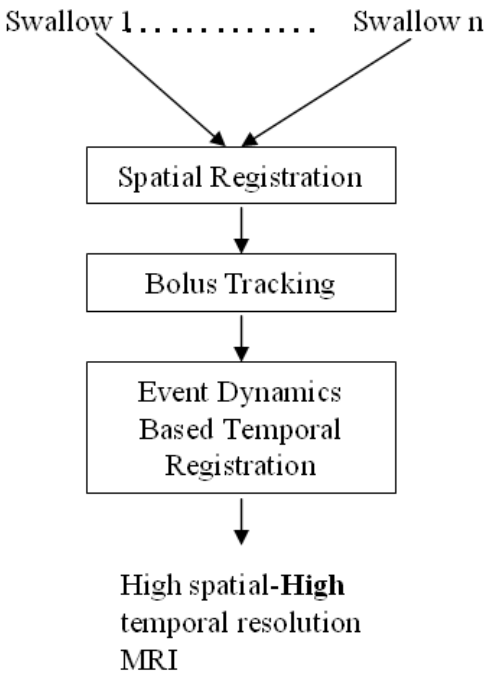


Figure 3.9: Overview of application of method to medical visualization.

MRI data is acquired using a radial acquisition method as 96 radial projections of 192 points and reconstructed to an image size of 384×384. Acquisition time for each image with the above configuration is 0.138 seconds, which computes to a frame rate of 7.2 fps. Some spatial movement of the head in between swallows is expected. To compensate for this movement we register the swallow images using linear conformal transformation by manually selecting control points in the two image sets and computing a transform based on these control points. Once the images have been spatially registered, the next step is to segment the bolus of water and track its path in the space-time volume of the swallow.

In order to segment the moving bolus we use a standard background separation technique [85]. A background template image is calculated as the average of MRI images where no bolus is present (images towards the end of swallow acquisition). From this background image we subtract the swallow MRI images. Regions of motion are thus segmented as shown in Fig. 3.10 and 3.11(a-b). We then compute a

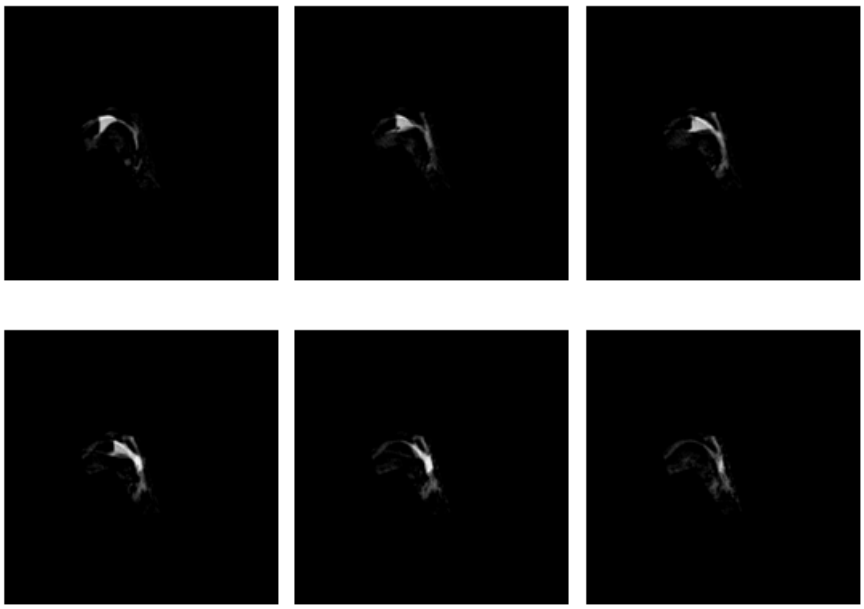


Figure 3.10: Background separated images from a swallow showing the bolus.

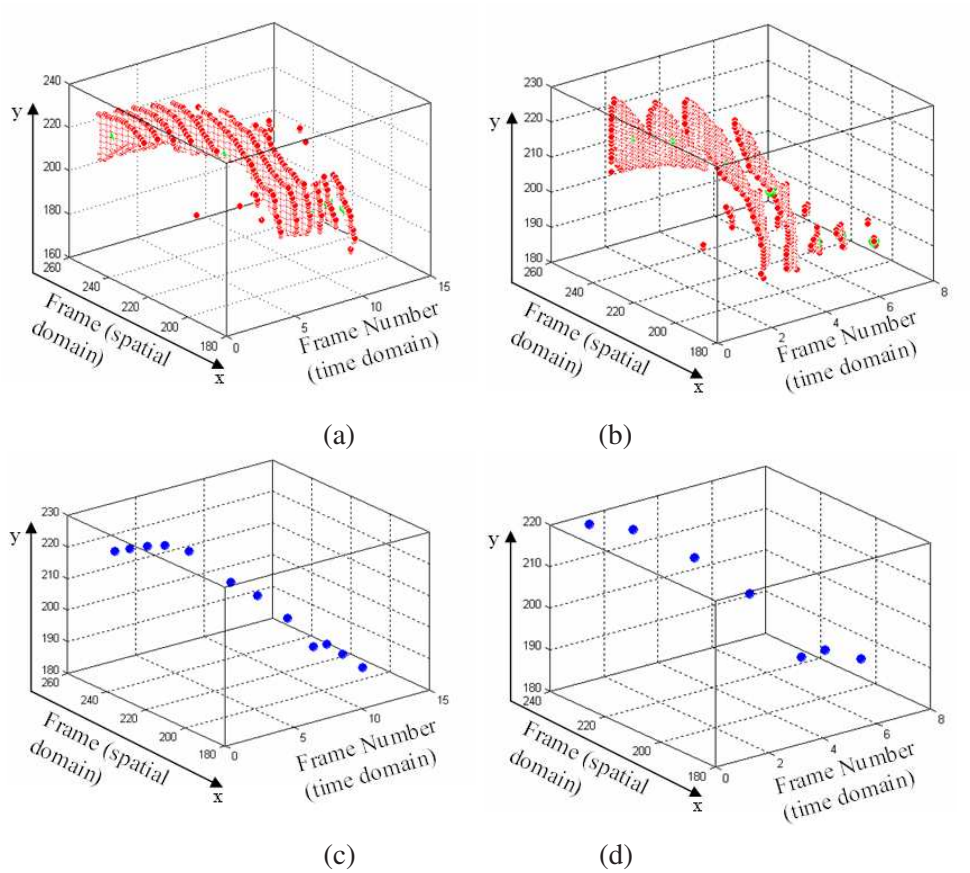


Figure 3.11: Bolus track in space-time volume of (a) swallow1 and (b) swallow2. Centroid path in space-time volume of (c) swallow1 and (d) swallow2.

weighted centroid of the bolus in all segmented images. The weights are computed in order to eliminate any residual noise. The centroid of the largest extracted region is assigned the highest weight and is retained, while other regions are considered as noise and removed from the space-time volume. A path of the centroid motion is then created as shown in Fig. 3.11 (c) and (d). The motion of the centroid is the trajectory path in this application.

In our experiments a polynomial of degree 6 was fitted on the coordinates of centroids with a residual error of 6.747 pixels, an R-square value of 0.9982 and a RMSE value of 1.29 pixels. The results of temporal alignment with the Caspi model and the Event Dynamics model were comparable. Caspi model found the alignment to be offset by 3.55 frames, while our method determined the offset to be 4.45 frames. The results with Event Dynamics model are shown in Fig. 3.12.

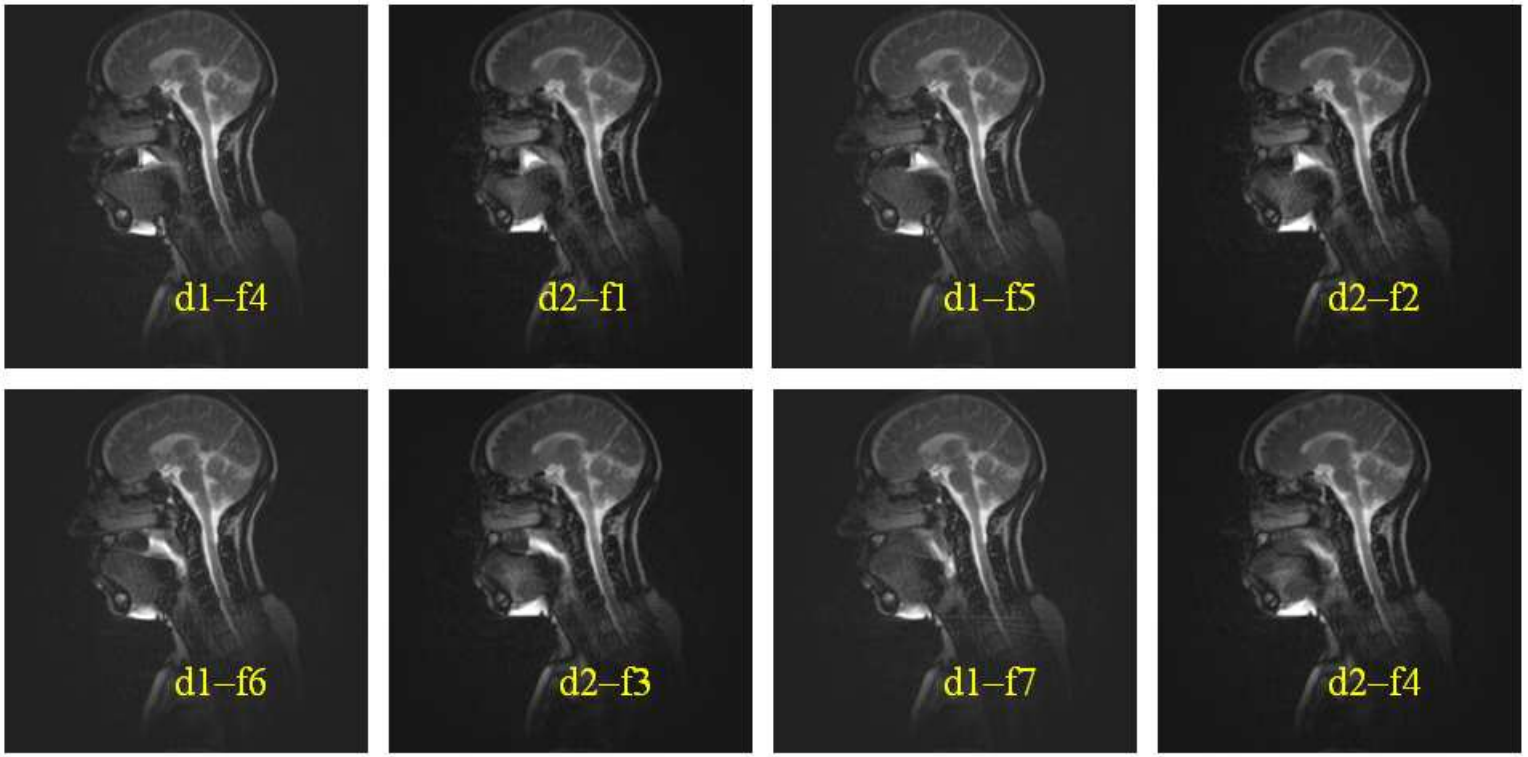


Figure 3.12: Two MRI datasets of swallowing aligned using offset determined by ED based matching. Legend: d <sequence number>-f <frame number>. Based on centroidal paths of the bolus of water, frames from sequence 2 were placed approximately midway between frames from sequence 1, but offset from the beginning of sequence 1 by 4 frames.

3.4.2 High Temporal Resolution Video Generation

The proposed Event Models technique can also be applied to the generation of high temporal resolution videos of events in the scenario where multiple cameras are acquiring a single event but at an unknown temporal offset from each other or when a single camera is acquiring multiple repetitions of an event. We validate this application by testing our algorithm on videos of spinning a ball on a thread and throwing a ball. Illustrative frames of these sequences have been compiled in Appendix B, Section B.1. These video sequences and the temporal registration results are available online at:

`www.ece.ualberta.ca/~meghna/ieeeMult/2006.html`.

Some representative frames are shown in Fig. 3.13. Figures (a)-(e) show consecutive frames of the original video sequence. This sequence was then subsampled by a factor of 5 with a time shift of 4 frames. Image (e) shows the frame that was retained in the subsampled sequences. Based on the temporal offset calculated by linear interpolation (g) has been incorrectly aligned in the series (f)-(j). Event Models on the other hand correctly aligned image (o) in the series (k)-(o). Using results from the real data experiments in Section 3.3, we can say that event dynamics based matching clearly performs better than linear interpolation and at worst performs as well as linear interpolation.

3.5 Event Dynamics Technique and Sampling Theorem

It would be interesting at this point to discuss the relation between the proposed Event Dynamics method and the Sampling theorem. The Sampling theorem states that in order for a continuous signal to be completely reconstructed from its discrete samples, the signal must be sampled at a sampling frequency higher than twice the maximum frequency of the signal or in other words twice the bandwidth of the

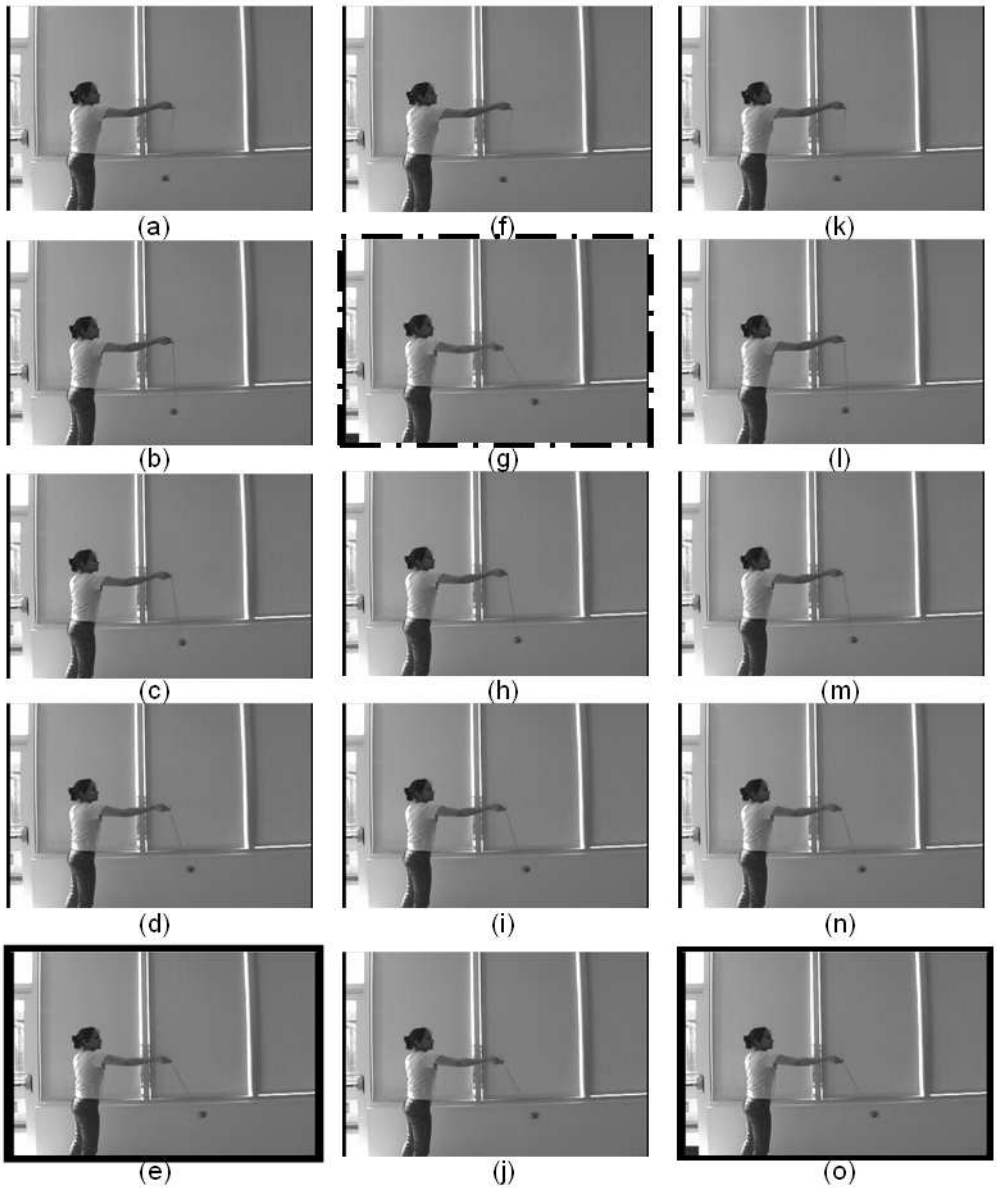


Figure 3.13: (a)-(e) Frame from the original sequence of ball swinging, (f)-(j) Temporal registration based on linear interpolation, (k)-(o) Temporal registration based on event dynamics. (e) is the frame in the original sequence that is to be aligned. (g) is the incorrectly aligned frame. (o) is the correct alignment of (e).

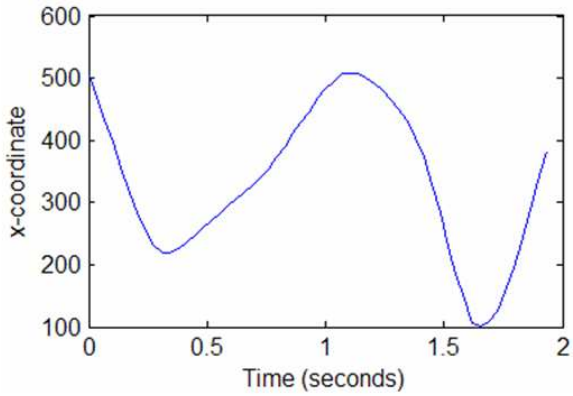


Figure 3.14: Trajectory of the centroid x-coordinate over 58 frames in a video taken at 30fps.

signal. If this sampling rate is violated then the reconstructed signal suffers from aliasing. In the temporal domain this results in temporal aliasing [5] in video sequences. With respect to our experiments in Section 3.3, let us assume that the video acquisition at 30fps (or 30Hz temporally) is higher than twice the maximum frequency of the motion in the video (swinging a ball). This assumption may not always be true, but is being made for the sake of discussion. A sample of the x-coordinate motion of the centroid of the ball is shown in Fig. 3.14. The Fourier spectrum of the x-coordinate motion is shown in Fig. 3.15. It can be seen from Fig. 3.15 that most the signal (x-coordinate motion) power is accumulated in the frequency below 5Hz. This implies that if we were to sample the acquired video further, we should sample it at a rate higher that 10Hz in order to reconstruct the motion of the x-coordinate accurately. However, in our experiments we sample the 30fps video at frequencies as low as 4Hz, which is much below the sampling frequency of the signal. It is to be noted that we are not reconstructing the signal from just a single sub-sampled video but rather we are trying to recover a better version of the signal from a temporally aligned set of samples. A detailed examination of the sampling theorem and its relation to this work is presented in the next chapter.

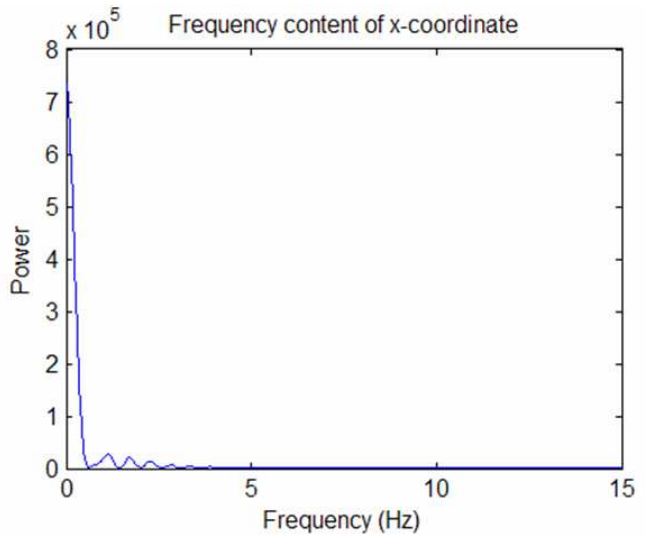


Figure 3.15: Power spectrum of the trajectory in Fig. 3.14 showing that bandwidth can be approximated to be below 5Hz

3.6 Conclusions

In this chapter we presented a method using event dynamics (ED) to temporally register multiple video sequences together. We also presented a comparative analysis of ED matching and linear interpolation for sub-frame temporal alignment of real sequences and synthetic trajectories. For fast occurring events with low acquisition rates, an ED based matching computes temporal offsets with higher accuracy than matching linearly interpolated values. It is to be noted that for such ED matching, no landmark points need to be calculated on the trajectory and the computation time is small as no brute force search is performed. However, a reasonably accurate polynomial curve fit does need to be computed. We analyzed the effect of noise on the accuracy of the ED based temporal alignment algorithm as well as the contemporary linear interpolation algorithm. We showed that the event dynamics based approach results in lower errors even at high noise levels. We also presented two applications of our work in medical visualization and high temporal resolution video generation.

In the next chapter we address a more complicated scenario where the sequences

are acquired from varied view-points and are of related events (and not the same event), thus the temporal scale of the activities in the sequence vary with time.

Chapter 4

Symmetric Transfer Error for Spatio-Temporal SR Imaging

In the previous chapter we presented event models and discussed how event models can be used to improve registration of low frame rate sequences for SR imaging. However, the problem scenarios used to compute and compare event models were restricted to multiple sequences of the same scene. In this chapter we expand on our previous work to address a more complex problem scenario where video sequences of related events are acquired via uncalibrated cameras with unknown and dynamically varying temporal offsets.

The rest of this chapter is organized as follows. In Section 4.1 we present the problem statement and outline related work in video synchronization, and justify the necessity for this work. We also review the closest related work – Rank Constraint based algorithm (RCB) proposed by Rao *et al.* [54]. In Section 4.2, we present our proposed symmetric transfer error and an optimization strategy to minimize it. In Section 4.3, we present the performance evaluation of the proposed technique. Application of our work in 4D MRI visualization is presented in Section 4.4. Finally, summary and conclusions are presented in Section 4.5.

4.1 Review of Synchronization Techniques

Synchronization of video sequences plays a crucial role in applications such as super-resolution imaging [5][6], 3D visualization [57], robust multi-view surveillance [36] and mosaicking [27]. Most video synchronization algorithms deal with video sequences of the same scene and hence assume that the temporal offset between the video sequences does not change over time, *i.e.*, a simple temporal translation is assumed. Video synchronization, however, is not limited to aligning sequences of the same scene. The synchronization techniques can be extended to find the spatio-temporal alignment between related scenes, for applications such as video search, video comparison and enhanced video generation. In such scenarios, the temporal offset between the video sequences changes dynamically, and cannot be estimated by a translational offset. For example, if $Frame_N$ in Sequence 1 matches $Frame_M$ in Sequence 2, then for dynamic offsets it is possible that $Frame_{N+1}$ may not match $Frame_{M+1}$. Rather $Frame_{N+1}$ could match $Frame_{M+k}$, where k is an arbitrary offset that is positive and changes with time.

The closest related work in synchronization of video sequences that are related by dynamically varying temporal offsets is limited to integer frame alignment of video sequences [54]. Rao *et al.* [54] use rank constraint as the distance measure in a dynamic time warping algorithm to align multiple sequences. In the Rank Constraint Based (RCB) algorithm proposed by Rao *et al.* [54], eight corresponding points between the first frames of two videos, say $(x'_1, y'_1), ..(x'_8, y'_8)$ and $(x_1, y_1), ..(x_8, y_8)$, are specified. Feature points in video sequences are tracked to acquire trajectories (u', v') and (u, v). The RCB algorithm uses the 9^{th} singular value of the matrix M defined in Eq. 4.1 as the distance function $d_{(i,j)} = \sigma_9(M)$

for warping.

$$M = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ x'_8x_8 & x'_8y_8 & x'_8 & y'_8x_8 & y'_8y_8 & y'_8 & x_8 & y_8 & 1 \\ u'_iu_j & u'_iv_j & u'_i & v'_iu_j & v'_iv_j & v'_i & u_j & v_j & 1 \end{bmatrix}$$

The warping matrix E represents the error in alignment upto frame i in sequence 1 and frame j in sequence 2.

$$E(i,j) = d_{(i,j)} + min(E(i-1,j), E(i-1,j-1), E(i,j-1)) \tag{4.1}$$

However, the RCB approach has the following limitations:

- The algorithm cannot compute synchronization with a sub-frame alignment.
- The dynamic time warping implementation and the distance measure used leads to singularities in video synchronization.
- If the eight corresponding points chosen in their rank matrix are from points that are not in temporal sync, then synchronizing using the 9^{th} singular value will not work.
- The authors also mention that when there are other points close to the correct match, then the matching algorithm results in ambiguities.
- The warping is computed in one direction, for example, Sequence 1 is warped towards Sequence 2. This uni-directional warping results in an alignment that is biased towards the reference sequence. Any error in feature extraction or noise in the reference sequence disproportionately affects the outcome of the synchronization.

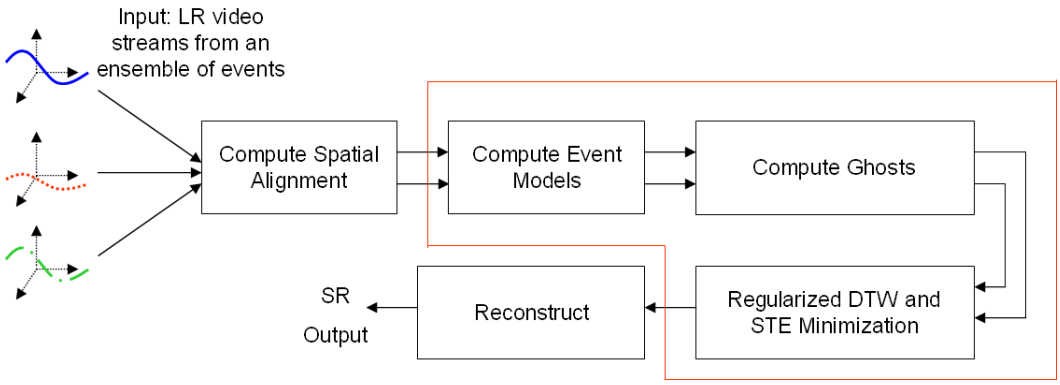


Figure 4.1: System Overview of Symmetric Transfer Error (STE) Approach. DTW – Dynamic Time Warping.

4.2 Proposed Approach

Our contribution lies in formulating the synchronization problem as the iterative minimization of a *symmetric transfer error* (STE), which allows us to compute *sub-frame accurate* synchronization of video sequences that have a *dynamically varying temporal offset* between them. In addition, the method of minimizing STE allows us to reduce the occurrence of singularities in the synchronization. Singularities [34] occur when multiple frames in the target sequence map to the same frame in the reference sequence. Such a situation can occur when the temporal speed of an event in one sequence is significantly slower than another sequence. However, in most cases this slowing down of the event should lead to a sub-frame mapping and not singularities.

A brief system overview of the proposed approach is shown in Fig. 4.1. The input to the system are low resolution video streams from an ensemble of events. The first system module computes the spatial alignment between these video streams. In this work we assume that the frame rates of the cameras are identical and fixed throughout the acquisition, the scene is planar and the backgrounds in both the scenes have sufficient static points that can be extracted using view-invariant feature detectors [47] to estimate the spatial relationship between the two scenes. These as-

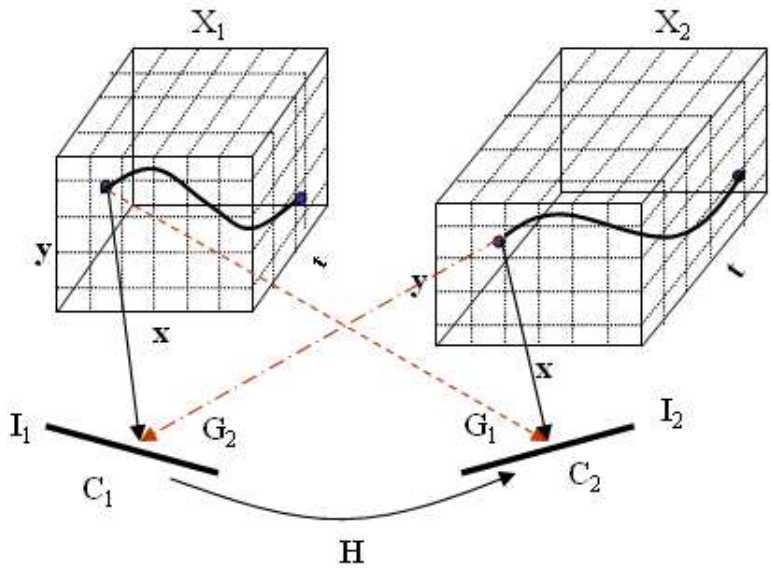


Figure 4.2: Illustration of two distinct scenes acquired using two distinct cameras. The projections (ghosts) of scenes onto the reciprocal cameras are also shown.

sumptions are not limiting since the synchronization algorithm can be adapted to account for epipolar geometry for non-planar scenes and robust correspondence algorithms can be used for wide-baseline cameras [46].

In the second module we extract feature trajectories from the sequences and compute event models (as described in Chapter 3). We assume that a single feature trajectory of interest is available to us such that the beginning and end points of the activity are marked in the trajectory, similar to the assumption made in [54]. In general videos, multiple object trajectories will be generated and an additional task of the synchronization algorithm will be to find corresponding feature trajectories in the multiple video sequences. This is an open problem in vision research and one that we will not address in this thesis.

In the third module (Fig. 4.1), we compute trajectory projections (ghosts) and the dynamic time warp between the event models. A symmetric transfer error is minimized in the fourth module before reconstructing the high resolution video sequence. These system modules are discussed in detail in the following sections.

4.2.1 Spatial Alignment

Spatial Alignment is the first module in the system overview shown in Fig. 4.1. Given that we have two cameras C1 and C2, as shown in Fig. 4.2, that view two independent scenes of similar activities; C1 views scene X_1 and acquires video I_1, and C2 views scene X_2 and acquires a video sequence I_2. Features, F_1 and F_2, are extracted and tracked in both the acquired video sequences. The spatial relationship (homography H) between the two scenes is computed by using the Random Sampling and Consensus algorithm (RANSAC) [18] and the Direct Linear Transform (DLT) algorithm [25]. We use these spatial alignment algorithms as follows. Edge and Corner feature points in the first frame of the input sequences are extracted. Each feature in one frame is matched to features in the second frame to find the correspondence between feature points. The RANSAC algorithm randomly chooses four corresponding features in the frames and computes a homography matrix H in the following manner.

Suppose $\mathbf{x}_i = (x_i, y_i)$ and $\mathbf{x}'_i = (x'_i, y'_i)$ represent a feature point correspondences extracted in frame 1 and frame 2 respectively. With four feature matches, we construct the matrix A as follows:

$$A = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 & -y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 & -x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 & -y'_2 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_nx'_n & -y_nx'_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_ny'_n & -y_ny'_n & -y'_n \end{bmatrix}. \quad (4.2)$$

The singular value decomposition of the matrix A is computed and the singular vector corresponding to the lowest singular value is the solved homography matrix H. The RANSAC algorithm applies this matrix H to all remaining feature points in the first frame, *i.e.* it projects these feature points onto the second frame. If the computed homography H is correct, then a majority of the projected features

will fall on their corresponding feature locations. The features that agree with the computed H are called 'inliers'. The algorithm runs a pre-determined number of iterations in order to maximize the number of inliers. The homography computed from Camera 1 to Camera 2 is denoted in this chapter as $H_{1\rightarrow 2}$ and from Camera 2 to Camera 1 is denoted as $H_{2\rightarrow 1}$.

4.2.2 Event Models

Feature extracted in Section 4.2.1 are tracked over all the frames in the sequence to generate feature trajectories. A single feature trajectory is used to illustrate this in Fig. 4.2. On their own these feature trajectories are discrete representations of the event in the scene, and we need to interpolate between the discrete representations. We use event models, proposed in Chapter 3, to generate continuous models from discrete points. The continuous feature trajectories are represented as $\mathcal{F}_1$ and $\mathcal{F}_2$. Note that we store these continuous models as discrete matrices in computer memory, but the temporal resolution of the continuous models is much higher than the discrete models. Hence, we index into $\mathcal{F}$ using integer numbers.

4.2.3 Compute Ghosts

As the feature trajectories are dynamically offset from each other (as illustrated in Fig. 4.2), we cannot directly optimize for both the homography and temporal offset. Direct optimization has been proposed by Caspi *et al.* [6], which is performed by using an estimated value of H and solving for a fixed temporal offset, and then using the computed offset as a fixed variable and solving for H. We cannot follow the same technique since the offset between two sequences in our problem scenario is not fixed. Instead, we project the trajectory from Scene 1 to Scene 2, as if Camera C2 did view Scene 1 (and vice versa). Note that the accuracy of this projection is subject to noise in the extracted trajectories as well as error in the homography computation. We call these projections the *ghosts* of the trajectories, represented as

$\mathcal{G}$.

$$\begin{aligned} \mathcal{G}_1 &= H_{1\to 2}.\mathcal{F}_1 \\ \mathcal{G}_2 &= H_{2\to 1}.\mathcal{F}_2 \end{aligned} \tag{4.3}$$

The ghost trajectory $\mathcal{G}_1$ is the projection of trajectory $\mathcal{F}_1$ (which was captured by the Camera C1) on to the imaging plane of the second Camera C2. The homography matrix $H_{1\to 2}$ is the projection matrix for $\mathcal{G}_1$. Similarly, the ghost trajectory $\mathcal{G}_2$ is the projection of trajectory $\mathcal{F}_2$ (which was captured by the Camera C2) on to the imaging plane of the first Camera C1. The homography matrix $H_{2\to 1}$ is the projection matrix for $\mathcal{G}_2$. These projections are illustrated in Fig. 4.2.

The aim of the synchronization algorithm is to temporally align a trajectory in Camera 1 ($\mathcal{F}_1$) to the ghost of the trajectory from Camera 2 ($\mathcal{G}_2$), and vice versa. Current dynamic offset synchronization algorithms, *e.g.* [54] and [21], synchronize discrete feature trajectories to a frame-by-frame correspondence by only computing a unidirectional alignment. For example, they assume Trajectory 1 to be the reference, and warp Trajectory 2 towards it. This unidirectional alignment biases the synchronization towards the reference sequence. Feature extraction and tracking errors in the reference sequence now propagate unchecked into the synchronization. We show in this work that a more symmetric approach will not only mitigate such an error propagation, but will also result in better sequence synchronization. Our symmetric optimization approach is presented in the following section.

4.2.4 Regularized Dynamic Time Warping and Symmetric Transfer Error

In the previous steps, feature trajectories and associated event models are computed independently for each sequence. This introduces a variability in the sequence alignment, such that frame alignment from Sequence 1 to Sequence 2 is not identical to frame alignment from Sequence 2 to Sequence 1. This asymmetry

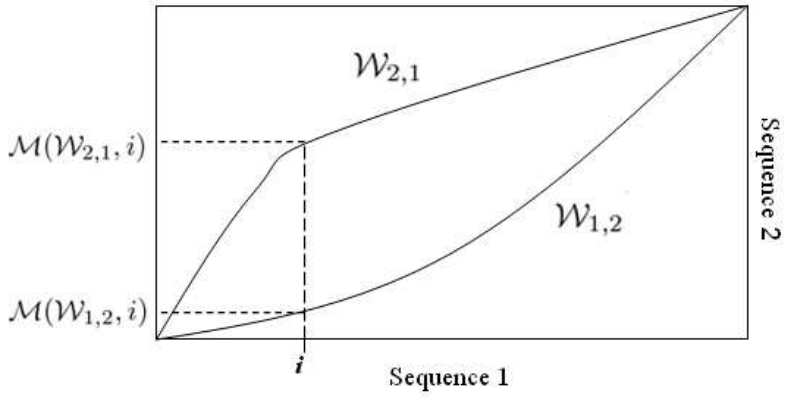


Figure 4.3: Computing symmetric transfer error for a single frame 'i' in Sequence 1.

is illustrated hypothetically in Fig. 4.3. In Fig. 4.3 the horizontal axis represents frames in Sequence 1 and the vertical axis represents frames in Sequence 2. The two reciprocal alignments computed between the sequences are shown as $\mathcal{W}_{1,2}$ and $\mathcal{W}_{2,1}$. *I.e.*, $\mathcal{W}_{1,2}$ is the mapping from trajectory $\mathcal{F}_1$ in Sequence 1 to the ghost $\mathcal{G}_2$ of Sequence 2, and $\mathcal{W}_{2,1}$ is the mapping from trajectory $\mathcal{F}_2$ in Sequence 2 to the ghost $\mathcal{G}_1$ of Sequence 1. Since the trajectories and event models are computed independent of each other, the reciprocal mappings are not identical, $\mathcal{W}_{1,2} \neq \mathcal{W}_{2,1}$. Asymmetry in mappings for real video sequences is shown in Fig. 4.13. If indexing function $\mathcal{M}(\mathcal{W}_{1,2}, i)$ indexes the mapping $\mathcal{W}_{1,2}$ and returns the frame in Sequence 2 corresponding to the i^{th} frame in Sequence 1 (similarly $\mathcal{M}(\mathcal{W}_{2,1}, i)$ indexes mapping $\mathcal{W}_{2,1}$ and returns the corresponding frame in this mapping), then the symmetric transfer error (STE) for the i^{th} frame (where $i \in \mathbb{R}$) is defined as follows:

$$\mathcal{E}(i) = |\mathcal{M}(\mathcal{W}_{1,2}, i) - \mathcal{M}(\mathcal{W}_{2,1}, i)| \tag{4.4}$$

Let us illustrate Eq. 4.4 with an example. Suppose in Fig. 4.3, for the i^{th} frame in Sequence 1, mapping $\mathcal{W}_{1,2}$ reports the corresponding frame in Sequence 2 to be the 7^{th} frame, and mapping $\mathcal{W}_{2,1}$ reports the corresponding frame to be the 10^{th} frame. Then, STE for the i^{th} frame is $\mathcal{E}(i) = 3$. STE for the entire sequence is then

calculated as follows:

$$\mathcal{E} = \sum_{i=1..max(N,M)} |\mathcal{M}(\mathcal{W}_{1,2}, i) - \mathcal{M}(\mathcal{W}_{2,1}, i)|, \tag{4.5}$$

where N is the length of $\mathcal{F}_1$ and $\mathcal{G}_1$ of Sequence 1 and M is the length of $\mathcal{F}_2$ and $\mathcal{G}_2$ of Sequence 2. Intuitively, minimizing STE for synchronization using Eq. 4.5 is akin to computing an optimal compromise between the reciprocal mappings $\mathcal{W}_{1,2}$ and $\mathcal{W}_{2,1}$.

The mapping functions $\mathcal{W}_{1,2}$ and $\mathcal{W}_{2,1}$ are computed using a regularized dynamic time warping (DTW) approach. The implementation of DTW via dynamic programming, factors in boundary conditions, continuity and monotonicity of the mapping function. We build a cost matrix $\mathcal{D}(n, m)$ $\forall n \in [1..N]$ and $\forall m \in [1..M]$ (Eq. 4.6) as follows:

$$\mathcal{D}(n, m) = \|(\mathcal{F}_1(n) - \mathcal{G}_2(m))\|^2 + w(\|\partial\mathcal{F}_1(n) - \partial\mathcal{G}_2(m)\|^2), \tag{4.6}$$

where the term $w(\|\partial\mathcal{F}_1(n) - \partial\mathcal{G}_2(m)\|^2)$ is the regularization function and the operator ∂ is defined as follows:

$$\partial\mathcal{F}(k) = \frac{\mathcal{F}(k) - \mathcal{F}(k-1) + [\mathcal{F}(k+1) - \mathcal{F}(k-1)]/2}{2} \tag{4.7}$$

In Eq. 4.6, w is the weight assigned to the regularization function. The motivation behind regularization of the cost function is two fold: (i) it allows us to factor in a smoothness constraint on the warping and (ii) it also reduces the occurrence of singularities in the mapping. The mapping function $\mathcal{W}$, is computed by traversing the path of minimum cost in the cost matrix $\mathcal{D}$, similar to the DTW algorithm reviewed in 2.10 and illustrated in Fig. 4.4, with i and j initialized to N and M respectively, as follows:

$$\mathcal{W}_{1,2}(n, m) = \mathcal{D}(n, m) + \min(\phi) \tag{4.8}$$

$$\phi = [\mathcal{W}_{1,2}(n-1, m), \mathcal{W}_{1,2}(n-1, m-1), \mathcal{W}_{1,2}(n, m-1)] \tag{4.9}$$

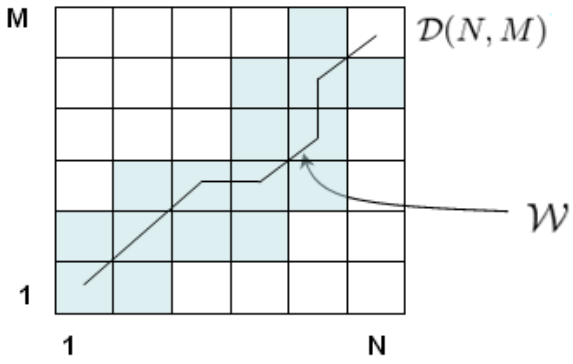


Figure 4.4: Illustration of computing the mapping function $\mathcal{W}$ from the cost matrix $\mathcal{D}$.

In Eq. 4.9 we consider a neighborhood of three frames (similar to [54]), however, this neighborhood can be extended. While Eqs. 4.6-4.9 detail how $\mathcal{W}_{1,2}$ is computed; the same apply to $\mathcal{W}_{2,1}$ with suitable substitutions made for $\mathcal{F}_2$ and $\mathcal{G}_1$. It can be seen from Eqs. 4.5-4.8 that STE is dependent on the regularization weight w, and the minimization of the STE with respect to w optimizes the sequence synchronization. An example of STE values computed for varying values of w for two synthetic trajectories are shown in Fig. 4.5, where the minimum symmetric error is achieved at a regularization weight of $w = 175$.

$$\mathcal{W}_{opt} = arg\min_{w} \sum_{i=1..max(N,M)} |\mathcal{M}(\mathcal{W}_{1,2}, i) - \mathcal{M}(\mathcal{W}_{2,1}, i)| \tag{4.10}$$

Equation 4.10 enforces the minimization of the difference between the two mappings $\mathcal{W}_{1,2}$ and $\mathcal{W}_{2,1}$ in Fig. 4.3. The advantage of optimizing a symmetric measure, as opposed to an asymmetric measure, is validated experimentally on real video sequences.

4.2.5 Algorithm Pseudocode

The pseudocode of the proposed algorithm is as follows:

- ***Pre-preocessing***

    1. Extract feature trajectories F_1 and F_2

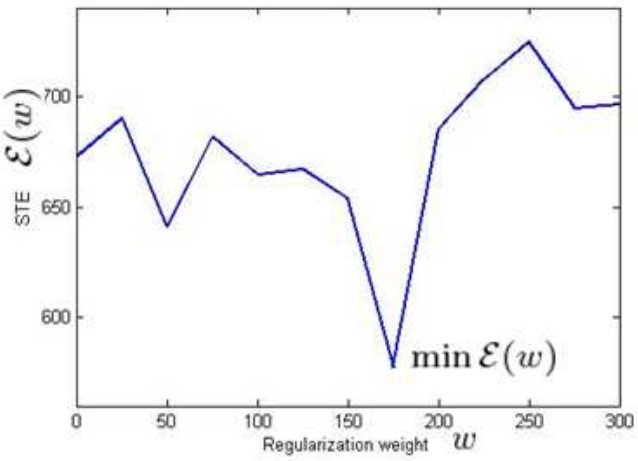


Figure 4.5: (Plot of STE $\mathcal{E}(w)$ versus regularization weight w for two synthetic sequences of length 100 and 140 frames. $min(\mathcal{E})$ is also indicated.

2. Compute Event models $\mathcal{F}_1$ and $\mathcal{F}_2$
3. Compute Homography H
4. Project Event models to derive ghosts $\mathcal{G}_1$ and $\mathcal{G}_2$
5. Set max_w, min_w, iter_step

- ***Iterate to minimize STE***

- **For** w=min_w : iter_step : max_w

1. Compute regularized warp $\mathcal{W}_{1,2}$
   (a) Compute cost metric $\mathcal{D}(n,m) = \|(\mathcal{F}_1(n) - \mathcal{G}_2(m))\|^2 + w(\|\partial\mathcal{F}_1(n) - \partial\mathcal{G}_2(m)\|^2)$
   (b) Compute mapping as the path of minimum cost $\mathcal{W}_{1,2}(n,m) = \mathcal{D}(n,m) + \min(\phi)$, $\phi = [\mathcal{W}_{1,2}(n-1,m), \mathcal{W}_{1,2}(n-1,m-1), \mathcal{W}_{1,2}(n,m-1)]$
2. Compute regularized warp $\mathcal{W}_{2,1}$
   (a) Compute cost metric $\mathcal{D}(n,m) = \|(\mathcal{G}_1(n) - \mathcal{F}_2(m))\|^2 + w(\|\partial\mathcal{G}_1(n) - \partial\mathcal{F}_2(m)\|^2)$

(b) Compute mapping as the path of minimum cost $\mathcal{W}_{2,1}(n,m) = \mathcal{D}(n,m)+\min(\phi)$, $\phi = [\mathcal{W}_{2,1}(n-1,m), \mathcal{W}_{2,1}(n-1,m-1), \mathcal{W}_{2,1}(n,m-1)]$

3. Compute Symmetric Transfer Error

(a) $\mathcal{E} = \sum_{i=1..max(N,M)} |\mathcal{M}(\mathcal{W}_{1,2}, i) - \mathcal{M}(\mathcal{W}_{2,1}, i)|$

(b) $\mathcal{M}$ is a simple indexing function.

- **End**
- Find $\min \mathcal{E}(w)$
- Report $\mathcal{W}_{opt}$ corresponding to $\min \mathcal{E}(w)$

For our experiments the variables in the pseudocode min_w, max_w, and $iter\_step$, were empirically determined and set to 0, 300 and 25 respectively.

4.3 Experiments and Comparative Analysis

We evaluated our method using both synthetic and real image sequences. We also implemented the rank-constraint based (RCB) algorithm as described in [54] that deals with aligning videos of similar events. While the RCB method cannot compute sequence alignment to sub-frame accuracy, we still compare our method with it for integer alignment. Our test cases can be divided into two sections - synthetic sequences and real sequences. For synthetic sequences, we evaluate and compare the STE algorithm to the RCB algorithm for integer alignment accuracy. For real sequences, we evaluate and compare the results upto sub-frame accuracy. These experiments and results are presented next. All experiments were run on a 3.2GHz Intel Pentium IV processor with 1GB RAM using MATLAB 7.04. Excluding the preprocessing time, the STE algorithm took 1.79 seconds to run through all iterations, to compute the optimal alignment between two synthetic sequences of length

100 and 140. The processing time for real sequences of length 84 and 174 was 1.93 seconds. The RCB algorithm was faster since it only computes the alignment once and not iteratively. The average run time for the RCB algorithm was 0.2 seconds for synthetic sequences and 0.25 seconds for real sequences. Even though the computation time for the proposed algorithm is higher than the contemporary method, it is still in the order of a few seconds and the increased run time is compensated by the increased accuracy of the alignment.

4.3.1 Synthetic Sequences

In synthetic tests, we generate planar trajectories (shown as the bottom plots in Fig. 4.6 and 4.7), 100 frames long, using a pseudo-random number generator ('rand' function in MATLAB). These trajectories are then projected onto two image planes using user defined camera projection matrices. An illustrative projection matrix is shown in the Appendix B, Section B.2.

The camera matrices are designed so that the acquisition emulates a homography, and are used only for generation purpose and not thereafter. A time warp is then applied to a section of one of the trajectory projections, such that its length now becomes 140 frames. Both the RCB and the STE methods are then applied to the synthetic trajectories to compute the alignment between them. This process is repeated on 100 synthetic trajectories. Figure 4.6 shows some synchronization results with simple synthetic trajectories. In Fig. 4.6 the horizontal axis of the chart presents frame numbers from Sequence 1 and the vertical axis represents frame numbers from Sequence 2. The lines drawn across the chart represent the mappings between the two Sequences. It can be in Fig. 4.6 that for simple trajectories, the STE and RCB methods result in comparable synchronization, close to the correct alignment. However, as the complexity of the synthetic trajectory begins to increase, the RCB method starts producing erroneous alignments while the STE

Table 4.1: Synchronization Errors for RCB and STE methods for Noisy Trajectories. Unit of measurement is the sum of absolute differences between the actual and computed frame correspondence.

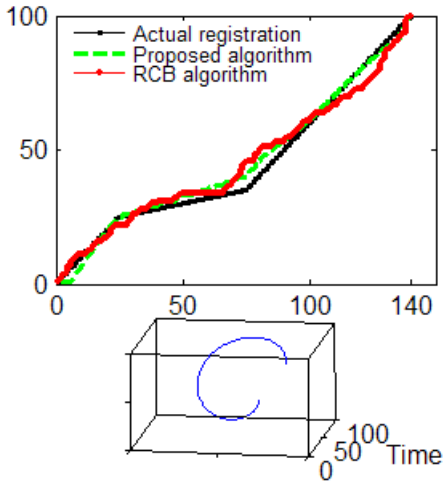
Noise σ^2	STE	RCB
0.0001	281	384
0.001	326	470
0.01	400	755
0.1	623	1199

method continues to compute synchronization that closely matches the actual synchronization, as can be seen in Fig. 4.7. Let u represent the errors made by the RCB algorithm and v represent the errors made by the proposed algorithm, then the percentage improvement in the synchronization results, w, is computed as follows:

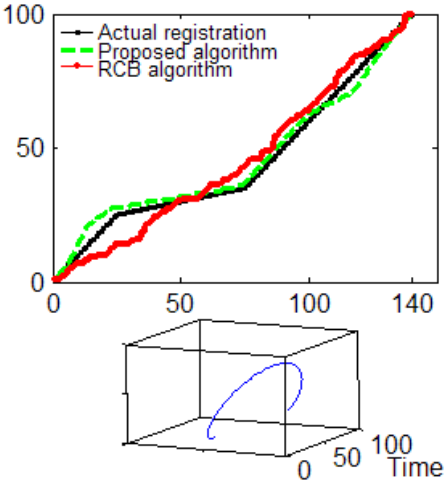
$$w = \frac{u - v}{u} \times 100. \tag{4.11}$$

On average, the proposed STE algorithm made 34% less errors in computing the synchronization when compared to the RCB method.

We also tested the effect of noisy trajectories on our synchronization approach. Normally distributed and zero mean noise with various values of variance (σ^2) was added to the synthetic feature trajectories. Two illustrative noisy trajectories are shown in Fig. 4.8. The results of synchronization of noisy trajectories with both the RCB and STE approach are shown in Table 4.1, where the sum of absolute differences between the actual and computed frame correspondence is reported as the synchronization error. When the noise variance is low ($\sigma^2 = 0.0001$), the proposed STE approach has 26% higher accuracy than the RCB algorithm, where the relative accuracy is determined using Eq. 4.11. As the noise variance increases, the STE approach is affected slightly by the addition of noise, however, the performance of the RCB method degrades dramatically. When the noise variance is ($\sigma^2 = 0.1$), the proposed STE approach has 48% higher accuracy than the RCB algorithm. Some representative alignment results are shown in Fig. 4.9.

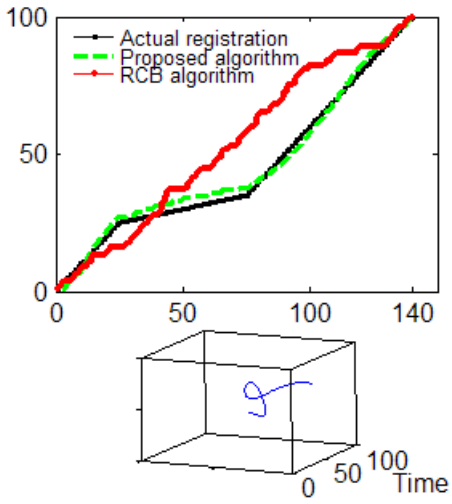


(a)

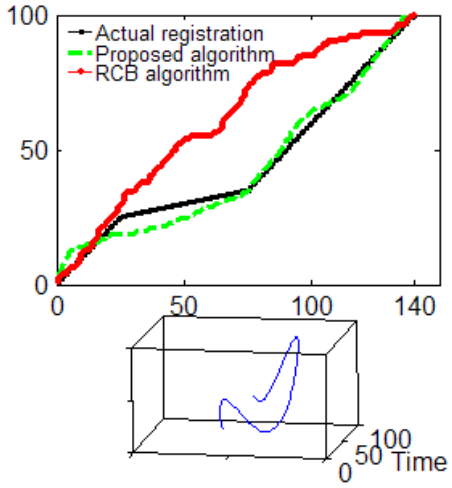


(b)

Figure 4.6: Results of synchronization of synthetic trajectories using proposed Symmetric Transfer Error (STE) approach and rank-constraint based (RCB) approach. (a)-(b) Simple trajectories result in comparable synchronization between both approaches.



(a)



(b)

Figure 4.7: Results of synchronization of synthetic trajectories using proposed Symmetric Transfer Error (STE) approach and rank-constraint based (RCB) approach. (a)-(b) More complex trajectories demonstrate the efficacy of the symmetric minimization approach.

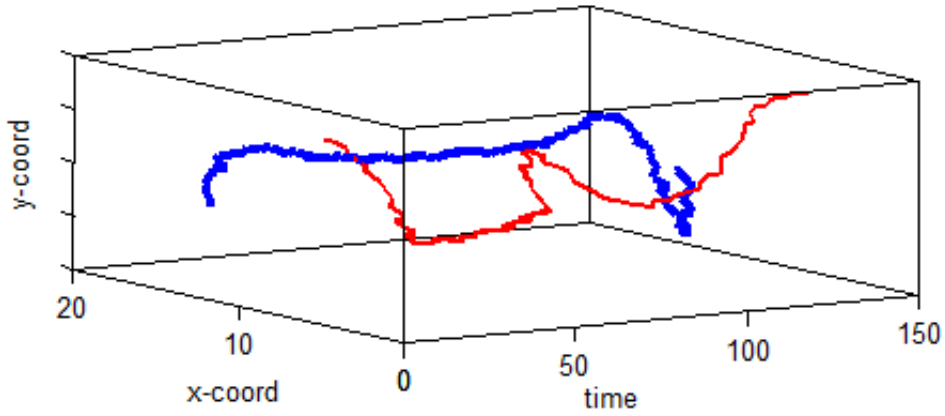
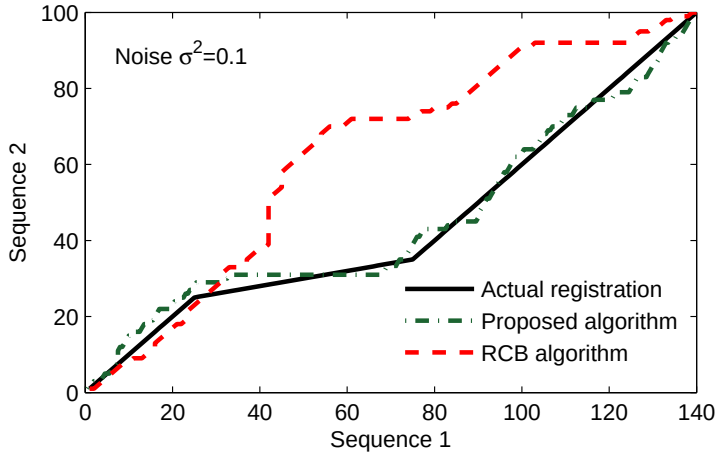
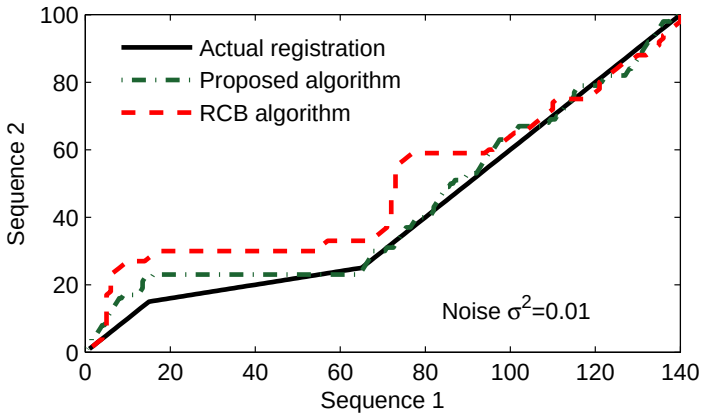


Figure 4.8: Two illustrative noisy trajectories with noise variance =0.1.



(a)



(b)

Figure 4.9: Performance of the STE and RCB methods for noisy trajectories with noise variance (a) 0.1 and (b) 0.01.

4.3.2 Real Sequences

We used video sequences provided by Rao *et al.* at http://server.cs.ucf. edu/∼vision/ and also acquired our own video sequences of activities similar to their data. Feature trajectories were available for the UCF video files. For our test video sequences, we provided an input template image of a coffee cup that was tracked in the video sequences to generate feature trajectories. The input sequences are shown in Appendix B, Section B.2.2. Both the proposed and RCB synchronization methods were then applied to the real video data. In the real data tests, ground truth information is not available. However, a tentative ground truth alignment is computed by visual determination.

Figure 4.10 shows the synchronization computed using the RCB algorithm. The top row of Fig. 4.10 (a)-(d), consists of frames from Sequence 1 that were matched to frames (shown in the bottom row (e)-(h)) from Sequence 2. The position of the object being tracked has been enclosed in a green rectangle to highlight the incorrect synchronization computed by the RCB algorithm. Fig. 4.11 shows the synchronization computed by the STE algorithm, where the frames in the top row are matched to frames in the bottom row. It can be visually determined that the alignment computed by the proposed method is a close temporal match.

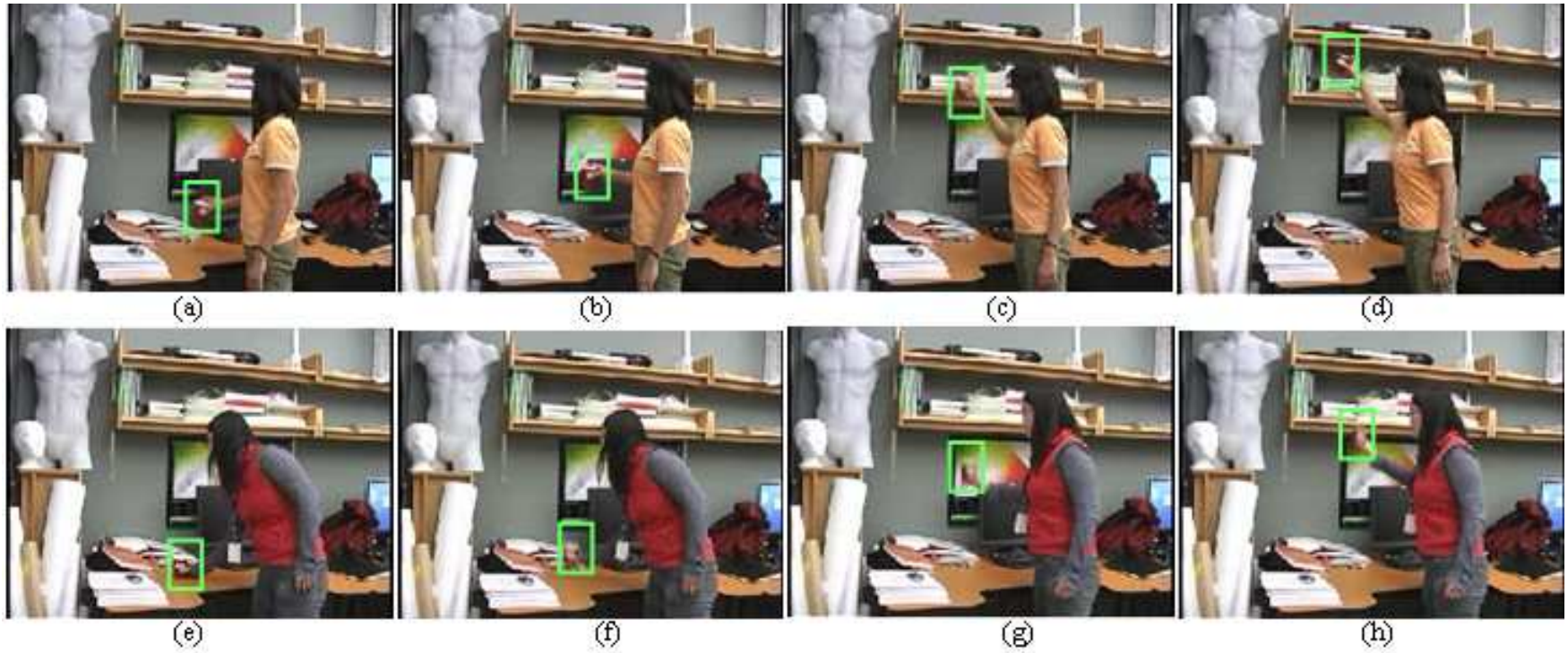


Figure 4.10: Results of synchronization using a rank-constraint based RCB method. The object being tracked is enclosed in a green rectangle. (a)-(d) Frames from Sequence 1, (e)-(h) Corresponding synchronized frames from Sequence 2.

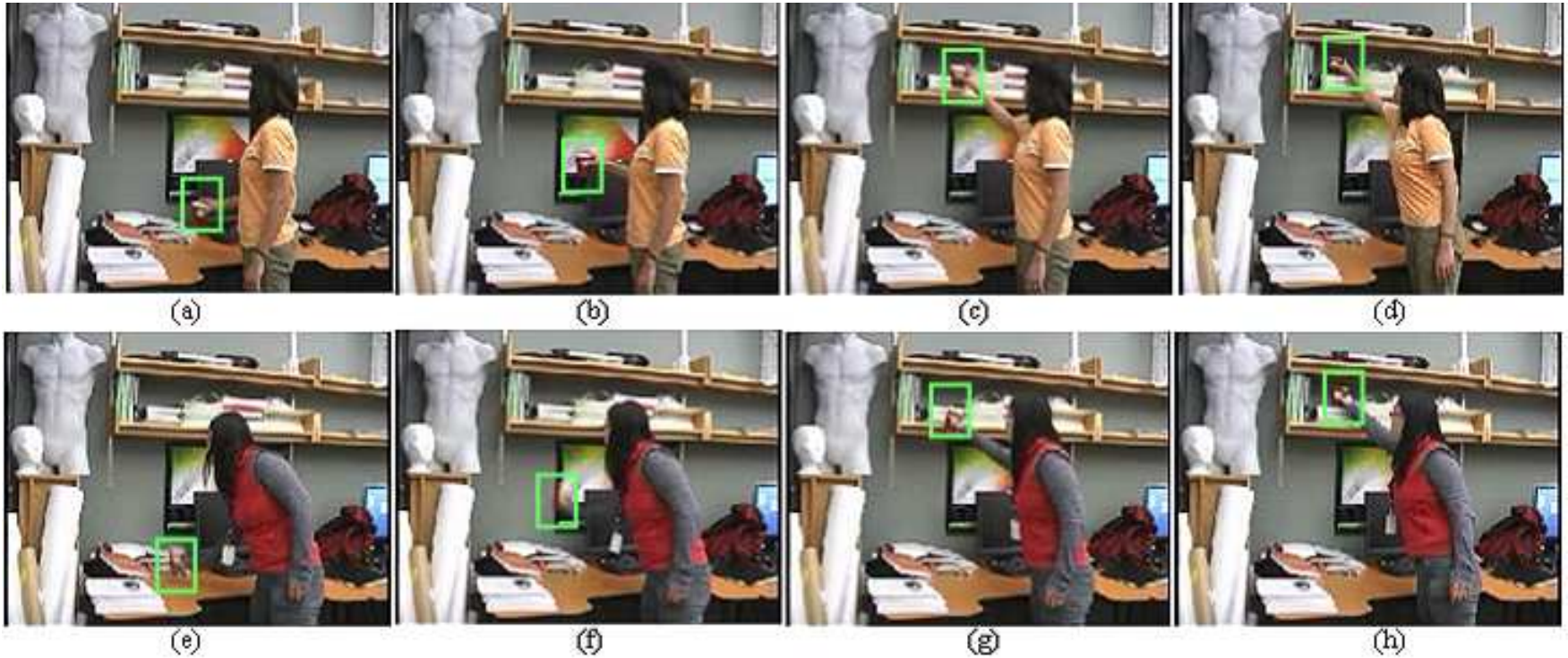


Figure 4.11: Results of synchronization using Proposed method. The object being tracked is enclosed in a green rectangle. (a)-(d) Frames from Sequence 1, (e)-(h) Corresponding synchronized frames from Sequence 2.

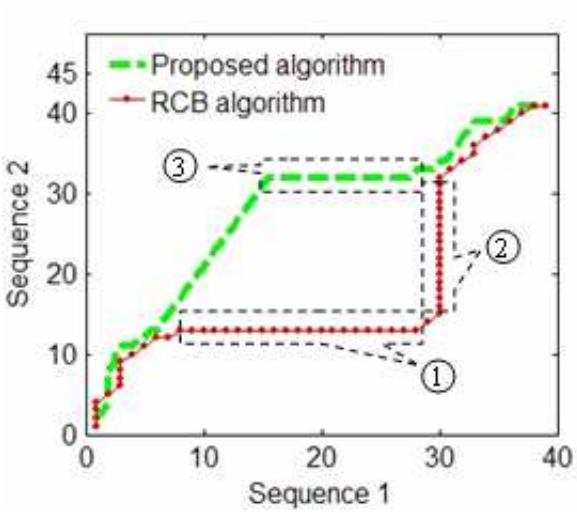


Figure 4.12: Warping computed between Sequence 1 and Sequence 2 of realUofA.avi test files. Point(1)-(3) are singularities marked on the warp.

The warping path computed between the two sequences by both methods is shown in Fig. 4.12. Points (1)-(3) marked on Fig. 4.12 highlight the regions in the warp where multiple frames in Sequence 1 were warped to a single frame in Sequence 2 and vice versa —*i.e.*, singularities. It can be seen from these highlighted regions that the proposed method reduces the number and length of singularities.

4.3.3 Symmetric vs. Asymmetric Synchronization

In the previous section we compared our STE based synchronization against the asymmetric synchronization proposed in [54]. The advantages of our approach in terms of sub-frame synchronization capability and reduction in singularities has been highlighted in the previously mentioned experiments. However, our approach and the RCB approach differ not only in the symmetry aspect, but also in the dynamic aspect of the cost function (Eq. 4.6). In order to highlight the advantage of the dynamic optimization of the STE, we validate it against unidirectional asymmetric synchronization using a fixed warping cost function. The mappings computed using both the symmetric and asymmetric approach for realUCF.avi test video are shown in Fig. 4.13. In Fig. 4.13, difference between the asymmetric mappings can

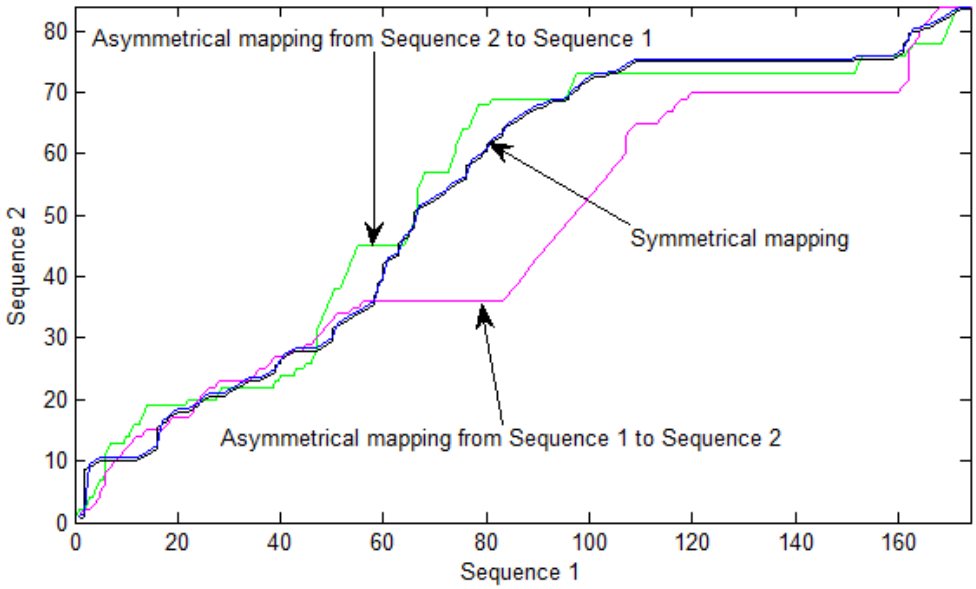


Figure 4.13: Symmetric vs. Asymmetric Synchronization of realUCF video files.

be clearly seen. This implies that depending on the video sequence that is chosen as the reference, the synchronization computed by the RCB algorithm will be different. However, note that the proposed STE method reduces the difference between the mapping from Sequence 1 to Sequence 2 and the mapping from Sequence 2 to Sequence 1.

4.4 Application – 4D MRI Registration

One of our motivations behind this work is to build a 4D (volume+time) representation of functional events in the human body using 2D planar acquisitions. Specifically we are investigating swallowing disorders. In Chapter 3, we presented the application of Event Models in registering MRI video sequences in the same imaging plane – mid sagittal plane. In this chapter, we investigate the scenario that the mutliple MRI videos are not acquired in the same imaging plane, as shown in Fig. 4.14. As swallow 'repetitions' are acquired in different imaging planes, the video sequences are related to each other by dynamic temporal offsets. We apply the RCB and proposed STE algorithm to compute the temporal offset to sub-frame accuracy.

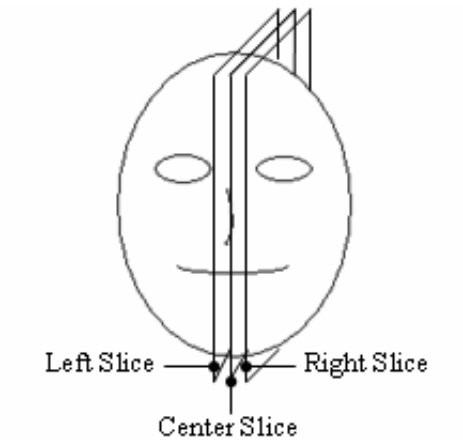


Figure 4.14: Illustration of spatial alignment of MRI slices.

Details of the input MRI sequences and method of acquisition are discussed in Appendix B. We acquire three video sequences corresponding to left, right and center MRI slice planes, as illustrated in Fig. 4.14. The MRI video sequences are subjected to dynamic temporal offsets in the motion of the bolus. Since the acquisitions are at very low frame rates, limited by the technology to 4-7 fps, it is crucial to align the sequences to sub-frame accuracy. Our approach shows promising results in aligning the MRI sequences to generate a 4D representation. Implementation details of this application are discussed next.

The trailing and leading edges of the bolus are extracted from the MRI sequences using standard background separation techniques [3]. The center of the trailing bolus is extracted using horizontal and vertical profiles, and is used to generate feature trajectories in the three sequences. After suitable event models have been computed for the trajectories, both the RCB and the STE algorithms are applied to the video sequences to compute synchronization. The results of synchronization with the STE and RCB algorithms are shown in Fig. 4.15, Fig. 4.16 and Fig. 4.17 respectively. Figure 4.15 shows a few frames from the synchronization computed between the center and right MRI slices that demonstrate sub-frame alignment. Frame 7 of the right MRI sequence is mapped to frame 6.5 of the center

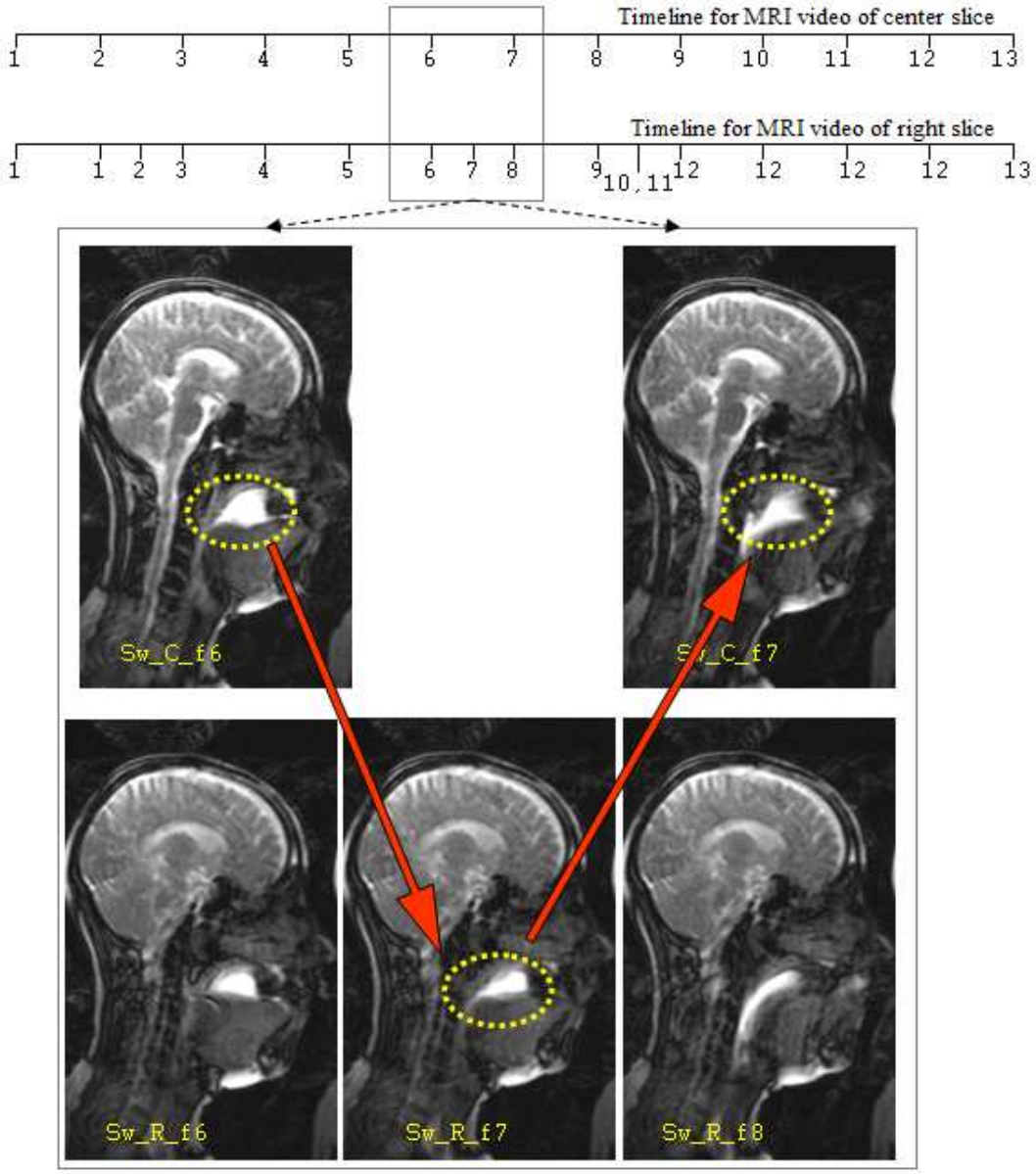


Figure 4.15: Synchronization computed between Center-Right MRI sequences. Frame 7 of the right MRI sequence is mapped to frame 6.5 of the center MRI sequence.

MRI sequence. Visually it can be seen that this sub-frame alignment is quite accurate and the results are much better than those produced with the RCB algorithm shown in Fig. 4.17.

Once the synchronization has been computed, 4D visualization of the MRI data is carried out with a 4D model as shown in Fig. 4.18. The 4D model depicts a sagittal volumetric section of the subject's body (from the shoulder to mid forehead). The multiple registered MRI frames are simultaneously rendered with varying levels of transparency. The colormap of the 4D model is set to show regions of high intensity as red and low intensity as blue. We have enhanced the images such that the bolus can be easily seen as the red section close to the nasal region.

4.5 Summary

We proposed and successfully tested a novel method to synchronize video sequences that are related by varying temporal offsets. Our formulation of synchronization as the minimization of a symmetric transfer error (STE) resulted in synchronization that was not biased by the choice of the reference sequence. The regularized nature of the STE significantly reduced the occurrence of singularities and resulted in sub-frame synchronization. Comparative analysis with a rank-constraint based method demonstrated a marked improvement in video synchronization with our method. An application of the proposed method in 4D MRI visualization was also presented.

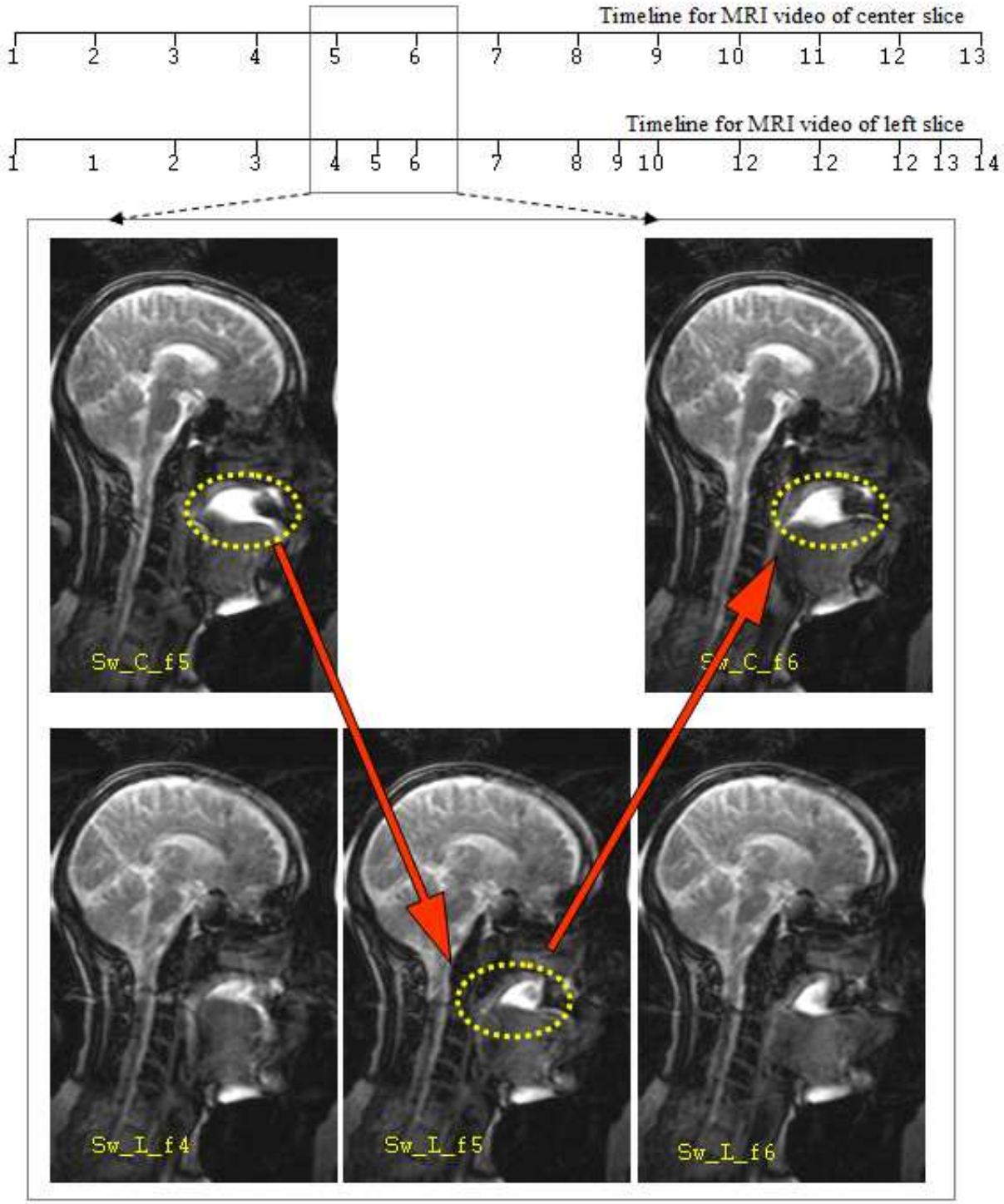


Figure 4.16: Synchronization computed between Center-Left MRI sequences by proposed algorithm. Frame 5 of the left MRI sequence is mapped to frame 5.5 of the center MRI sequence.

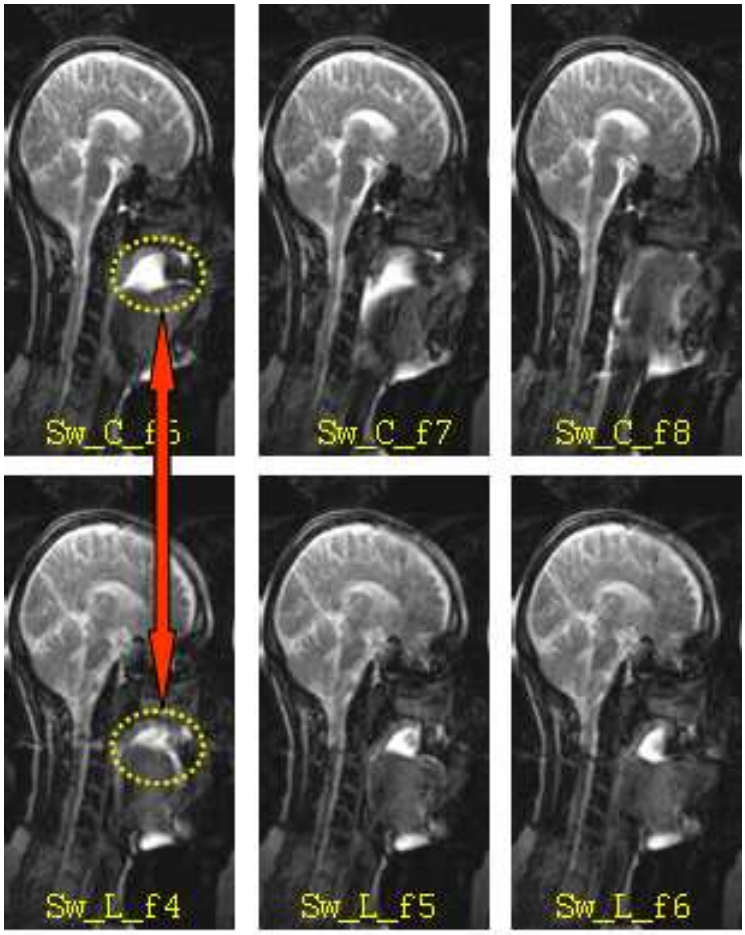


Figure 4.17: Synchronization computed between Center-Left MRI sequences by RCB algorithm. RCB algorithm is unidirectional and limited to integer frame alignment.

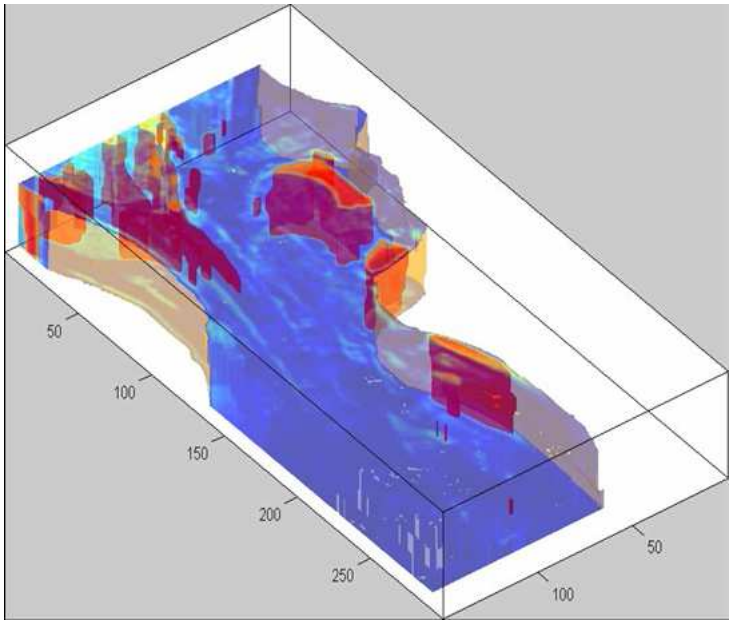


Figure 4.18: Static visualization (one time instant) of 4D synchronized MRI data. The colormap of the 4D model is set to show regions of high intensity as red and low intensity as blue. The bolus can be easily seen as the red section close to the nasal region.

Chapter 5

Spatio-Temporal SR Imaging from Orthogonal Viewpoints

In Chapter 4 we presented a novel method, based on symmetric transfer error, to align multiple video sequences of related events. It was assumed that even though the viewpoints of the multiple video sequences are different, there are sufficient corresponding points (atleast 4 in the case of a homography and 8 in the case of a fundamental matrix) that can be established in order to derive the spatial relationship between the sequences. However, in some applications, such as MRI, sequences from orthogonal view points are captured. In this scenario, no correspondence will be found between orthogonal sequences. In this chapter we present an approach to obtain super-resolution video from orthogonal view points.

The rest of this chapter is organized as follows. In Section 5.1 we present a brief review of related work in this area and define the problem that is addressed in this chapter. In Section 5.2 we present a strategy for alignment of orthogonal dynamic sequences. Experimental results and verification are presented in Section 5.3, and a summary is presented in Section 5.4.

5.1 Review of Dynamic MRI Registration

The ability to visualize patient data in 4D using Magnetic Resonance Imaging (MRI) offers medical practitioners many advantages over other imaging modalities

(CT/Ultrasound), including (i) visualization of soft tissue motion, (ii) short inter-exam time intervals and (iii) not subjecting patients to harmful ionizing radiation. While MRI was initially developed as a static imaging technique, new advances in protocols for MRI acquisition have seen the technology evolve into more dynamic applications such as imaging cardiac rhythms and blood flow. However, these dynamic acquisitions are still limited to the 2D image plane with very poor temporal resolution. In order to combat the tradeoff between spatial and temporal resolution in dynamic MRI, researchers align multiple acquisitions of a functional event (such as a cardiac cycle) to increase the temporal resolution of the data. Thompson *et al.*[70] use the sum of pixel intensities of complex-difference images in Fourier space as the gating signal to register multiple MRI sequences acquired in a single imaging plane. Siebenthal [79] *et al.* acquire a large dataset (180-240 images per slice) of MR images of the abdominal region in the sagittal plane and align the images by sorting them based on navigator frames (frame acquired in a fixed plane). This approach is suitable for generating 4D volumes of slow movements such as breathing, however, like Thompson *et al.* [70], the approach cannot deal with orthogonal image planes.

In Chapter 3 we addressed the problem of aligning MRI sequences by segmenting and tracking the centroid of the bolus (water that was swallowed) and using the centroid trajectories for alignment [64]. The MRI sequences were acquired in a single (mid-sagittal) plane. In this chapter, we extend the problem domain and propose an approach to align video sequences acquired in bidirectional planes. An illustration of the bidirectional acquisition is shown in Fig. 5.1, where we acquire MRI videos in three sagittal planes (center, left and right) and one video in the coronal plane. An interesting consequence of using bidirectional acquisition planes is that video sequences in the same acquisition view (*e.g* all sagittal sequences) automatically get aligned with respect to each other.

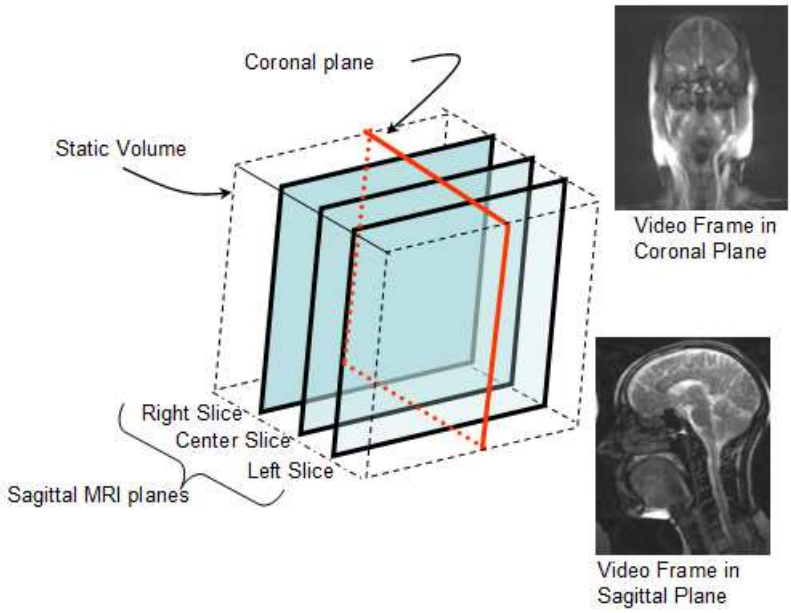


Figure 5.1: Illustration of MRI data acquisition planes.

We consider a representative application of visualizing swallowing in 4D. Current MRI research, that pertains to this application, involves breath-hold imaging [14][87] where the subject is asked to hold the position of the tongue for the duration of a breath while MR images are acquired. Thus, unlike our approach, contemporary methods cannot incorporate the temporal aspect of activities. Other assessment techniques (endoscopy and videofluoroscopy) for swallowing disorders are either invasive or require radiation exposure. Our goal is to develop methods that will allow diagnosis of swallowing disorders using non-invasive, non-ionizing MRI. Due to technological restrictions in MR imaging, we are limited to capturing the 4D swallowing process as a series of 2D images (acquisition is limited to about 7 dynamic frames during a swallow), *i.e.*, we can acquire video sequences of the swallow, but only in a single acquisition plane. As a major portion of swallowing is an involuntary task, we can capture nearly identical swallows by controlling the volume of the bolus. Thus, by changing acquisition plane over repeated swallows, and fusing these video sequences together we can generate 4D data. The task of the

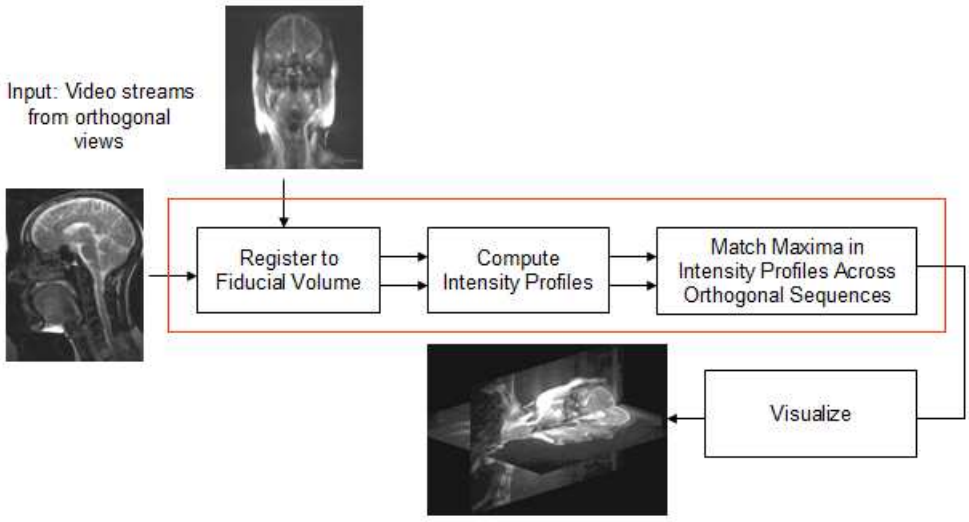


Figure 5.2: System overview of Bi-directional, Orthogonal Dynamic MRI Registration.

4D alignment problem is to compute the temporal alignment between these bidirectional, orthogonal video sequences using information from the image domain (no external gating is used).

5.2 Proposed Method

A system overview of the proposed method is presented in Fig. 5.2. Orthogonal MRI video sequences are acquired and registered to a static 3D reference volume, termed as the fiducial volume. Within each of the video sequences, intensity profiles are computed such that the variation in pixel intensity in select regions is tracked over time. These intensity profiles are matched to each other to register the orthogonal sequences and rendered for visualization. An intuitive explanation of the proposed method is to imagine that the moving bolus traces a 3D path in space, illustrated as the gray volume in Fig. 5.3. This 3D path is captured in multiple 2D planes as indicated by the sides of the cube in Fig. 5.3. The central idea behind our 4D alignment strategy is to find the volume or region in the bolus path that is common to the bidirectional acquisitions. The regions of interest in the coronal

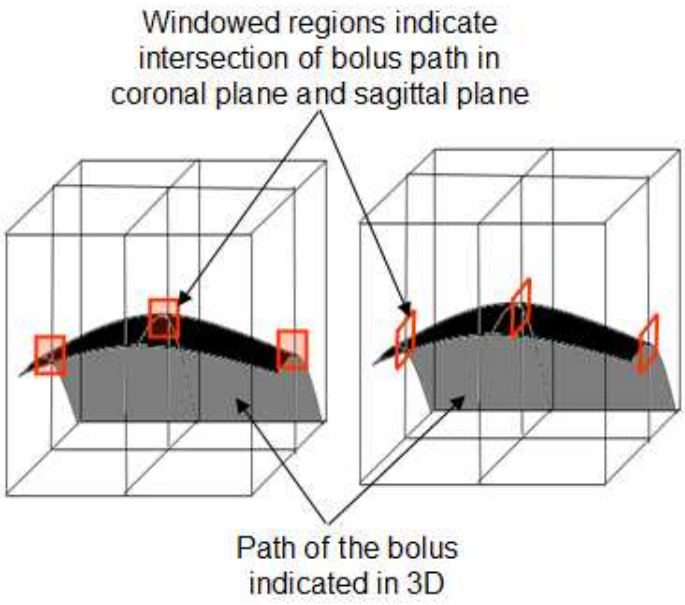


Figure 5.3: 3D path traced by the moving bolus and corresponding regions in the bidirectional planes that need to be identified.

plane are represented as red squares in the left cube and regions of interest in the sagittal planes are represented as red parallelograms in the right cube in Fig. 5.3. Once these regions are identified, we compute temporal profiles of pixel intensities to identify the corresponding frames at which the maximum volume of bolus traversed through these regions. Details of the proposed system are presented in the following sections.

5.2.1 MRI Acquisition

Swallowing images are acquired as even and odd radial acquisitions in k-space and re-gridded using a sliding window approach to video sequences with a frame rate of 7 frames per second. The sagittal dynamic MRI sequences are acquired in the left, right and center planes as shown in Fig. 5.1. The coronal swallowing sequence is acquired such that imaging plane bisects the oropharyngeal tract and the epiglottis, these anatomical regions have been annotated on the MR images in the Appendix B, Fig. B.9. Unlike contemporary methods, we do not place strict restrictions on the

acquisitions, and the MRI sequences are acquired at varying resolutions to simulate data acquired over a number of sittings and differing protocols. A static volume is also acquired as a series of images over 19 sagittal planes ($\sim$1cm apart). This volume is used as a fiducial volume to find corresponding regions in the bidirectional MRI sequences.

5.2.2 Registration to Fiducial Volume

As the protocol for acquiring the coronal and sagittal sequences can be different, we first register these sequences to a common fiducial volume. The spatial relationship between the respective planes of the fiducial volume and dynamic MRI sequences is approximated by an affine transform. An operator manually selects anatomically relevant control points to register the first frames in the dynamic sequences to the sagittal and coronal slices of the fiducial volume. We found that automatic registration methods such as SIFT[42] and RANSAC[18] (reviewed in Section 2.7.3) did not find any inliers to compute image registration as the dynamic images and the fiducial volume are noisy and do not have sharp edges or corners. The computed transforms are then applied to all the frames in the sequence. Once the dynamic sequences are registered to the fiducial volume, regions identified in one dynamic sequence can be projected onto the orthogonal sequence.

5.2.3 Computing Intensity Profiles

At the time of acquisition the coronal image plane is chosen such that it intersects the sagittal image plane close to midway (intersection is approximately determined based on the anatomy of the pharyngeal region). A slight variance in determining this intersection does not significantly affect the algorithm as we compute the intensity profiles over a windowed region. We use a columnar region of width 'w' (w = 7 pixels) around the plane of intersection as the region of interest, as shown in Fig. 5.4. The pixel intensities in this region are summed along the width of the

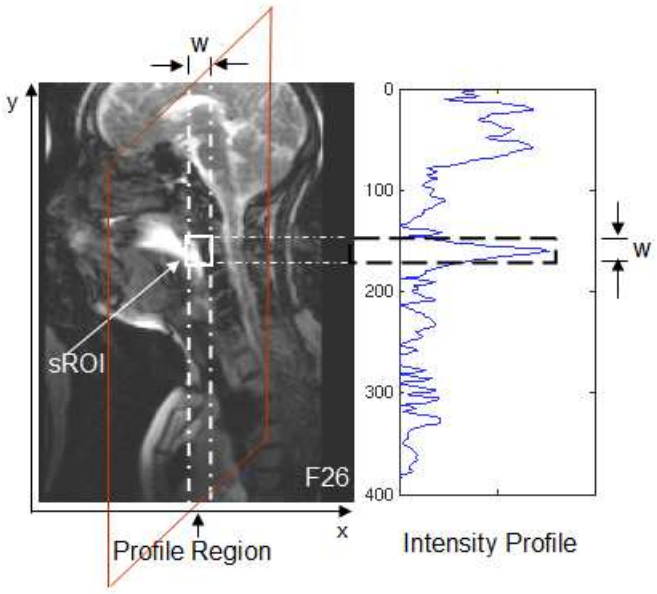


Figure 5.4: Intensity profile computed for frame-26 of the center sagittal sequence. The sagittal region of interest (sROI) is also indicated.

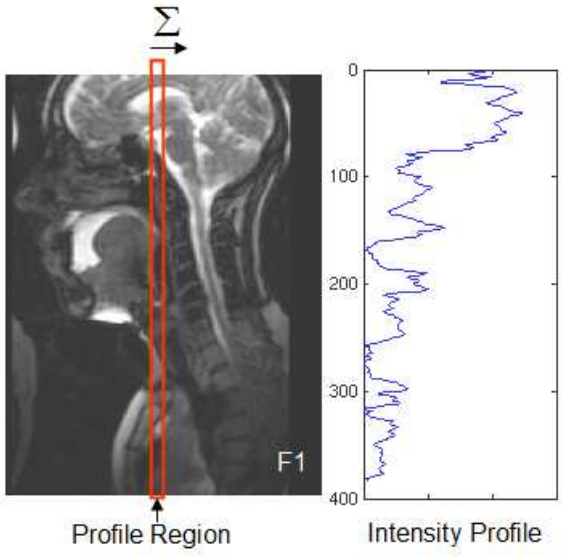


Figure 5.5: Intensity profile computed for frame-1 of the center sagittal sequence.

column to generate a columnar profile of the region. For example, let $I(x, y, k)$ be the k^{th} frame of size $N \times M$, and the intersection of the coronal plane with this frame be approximated at column $x = P$. A columnar region $W(x, y)$ corresponding to $x \in (P - w/2 : P + w/2)$ and $y \in (1 : M)$ is chosen such that the columnar intensity (CI) for the k^{th} sagittal frame can be computed as:

$$CI(y, k) = \sum_{x} I(x, y, k) \qquad \forall x, y \in W. \tag{5.1}$$

The average columnar intensity (avCI) for the first few frames (n) in each sequence (assuming the swallow process has not begun yet) is used as a benchmark intensity profile against which subsequent profiles are compared. The avCI can be computed as follows:

$$avCI(y) = \frac{1}{n} \sum_{k} CI(y, k) \quad for \quad k = 1..n. \tag{5.2}$$

The difference in intensity caused by motion of the bolus in the columnar region is computed as follows:

$$CIdiff(y, k) = |CI(y, k) - AvCI|. \tag{5.3}$$

We observe the difference in intensity peaks at the y-location where the path of the bolus intersects the columnar region. This can be seen by comparing Fig. 5.4 which corresponds to Frame 26 to Frame 1 of the sequence shown in Fig. 5.5. We search for the maximum value in CIdiff and also the y-location where this value occurred by using MATLAB's in-built function *max*. The function *max* returns the maximum value in the array provided to it as input and also the location where the value occurred:

$$[mVal(k), mInd(k)] = \mathbf{max}(CIdiff(y, k)). \tag{5.4}$$

We compute $mVal$ from Eq. 5.4 over all the frames of the video sequence, and search for frame at which the maximum value of $mVal$ occurs, as shown in Eq.

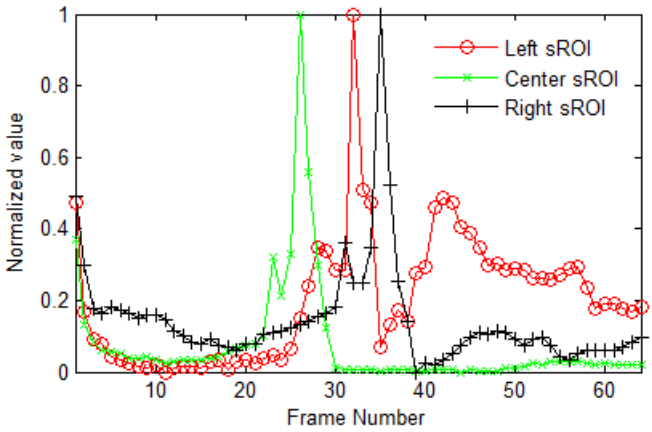


Figure 5.6: Dynamic Intensity Profiles over sagittal regions of interest (sROI). The peak in each profile indicates the frame at which the maximum bolus passes through that region.

5.5. The frame at which the maximum value is achieved is termed as an 'Sframe'.

$$[val, Sframe] = \mathbf{max}(mVal(k)) \;\; for \;\; k = 1..K \tag{5.5}$$

The region of interest in the sagittal sequence is determined from the Sframe as a $w \times w$ pixel region at intersection of the bolus path and the columnar region as – $sROI(x, y, k) = I(x, y, k)$ for $x \in (P-w/2 : P+w/2)$ and $y \in [mInd(Sframe)-w/2 : mInd(Sframe) + w/2]$. The pseudocode for computing the sROI is shown in Table 5.1. The dynamic intensity profile over the sagittal region of interest is the sum of pixel intensities in sROI represented over time, as shown in Fig. 5.6. Thus, for the three dynamic sagittal MRI sequences we compute three dynamic intensity profiles over the ROIs as shown in Fig. 5.6.

5.2.4 Matching Maxima in Intensity Profiles

In the previous section we described how regions of interest are identified in the sagittal plane. As illustrated in Fig. 5.7, these ROIs have to be projected to the fiducial volume and onto the coronal imaging plane using the affine transforms computed in Section 5.2.2. In the fiducial volume, the height of the sagittal ROI is computed as a w pixel wide region around the pixel with the highest intensity

Table 5.1: Pseudocode for computing sROI and Sframe

$avCI(y) = \frac{1}{n}\sum_k CI(y,k)$ for $k = 1..n$
For k=1:K *%K is the length of the video sequence*
$CI(y,k) = \sum_x I(x,y,k) \forall x, y \in W$
$CIdiff(y,k) = |CI(y,k) - AvCI|$
$[mVal(k), mInd(k)] =$ **max**$(CIdiff(y,k))$
*% **max** returns the maximum value in the array and also the*
% location where the value occurred
End
$[val, Sframe] =$ **max**$(mVal)$
$sROI(x,y,k) = I(x,y,k) \forall x \in (P - w/2 : P + w/2)$
and $y \in [mInd(Sframe) + w/2 : mInd(Sframe) - w/2]$

in the sROI profile $y \in [mInd(Sframe) + w/2 : mInd(Sframe) - w/2]$. This height is transferred directly to the coronal projection of the fiducial volume. The width in the coronal projection is computed as a w pixel wide region around the intersection of the dynamic sagittal MRI and the dynamic coronal MRI. Thus a $w \times w$ coronal ROI (cROI) is identified in the fiducial volume. By using the image transform computed during initial registration operation, we project the cROI in the fiducial volume to a cROI in the coronal image plane. Note that for three planar intersections between the left-right-center and coronal image place, there are three cROIs computed.

Once the cROIs have been identified, we compute the dynamic intensity profiles over these ROIs in the same manner as the sROIs. We compute average intensity in the cROI when no bolus motion is present in the image as follows:

$$avCI = \frac{1}{n}\sum_k \sum_{x,y} I(x,y,k) \cap cROI(x,y) \;\; for \;\; k = 1..n, \tag{5.6}$$

where n are the first few frames of the video sequence. The intersection of I and $cROI$ only returns that region of I where the mask $cROI$ is 1. We compute the difference in pixel intensities in the coronal ROI over time as follows:

$$\begin{array}{c} CI(k) = \sum_{x,y} I(x,y,k) \cap cROI \\ CIdiff(k) = |CI(k) - AvCI| \end{array} \tag{5.7}$$

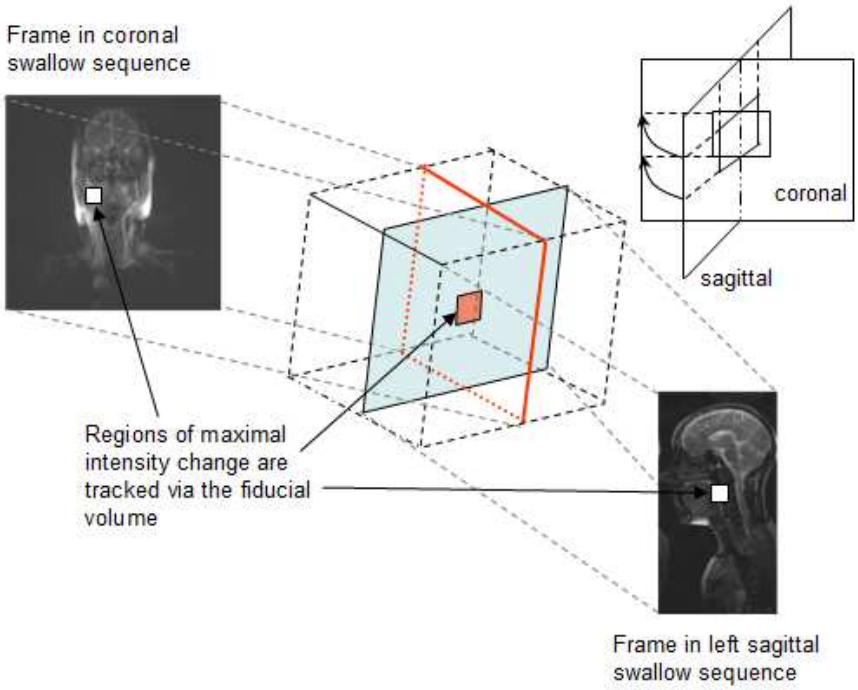


Figure 5.7: sROI is projected onto the fiducial volume and then further projected onto the coronal imaging plane.

The frame (Cframe) in the coronal sequence that corresponds to the bolus path intersecting this place is determined by finding the frame for which $CIdiff$ is maximum, *i.e.*:

$$[val, Cframe] = \mathbf{max}(CIdiff). \tag{5.8}$$

The pseudocode showing these operations in determining the Cframe is presented in Table 5.2. Figure 5.8 shows the normalized dynamic intensity profile for the three coronal regions of interest. Once the Sframes and corresponding Cframes have been computed, the sequences are simply lined up as per this correspondence, as shown in Fig. 5.9.

5.3 Results

All test files used in this chapter as well as corresponding results are available at [48]. A video of the 4D alignment is also available online at [48] and representative

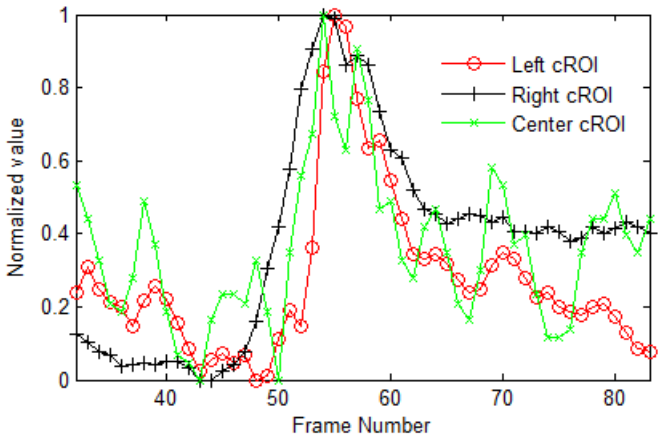


Figure 5.8: Dynamic Intensity Profiles over coronal regions of interest (cROI). The peak in each profile indicates the frame at which the maximum bolus passes through that region.

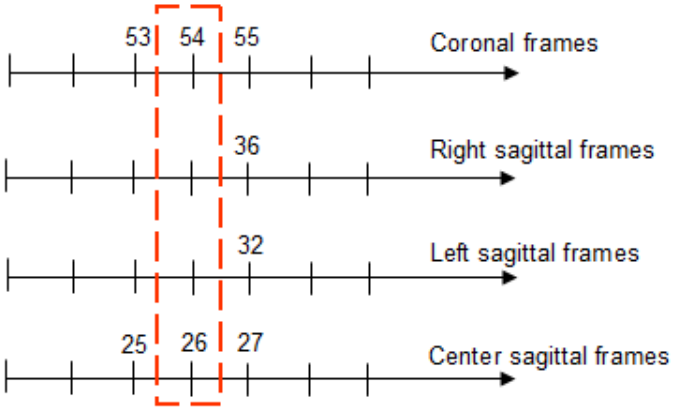
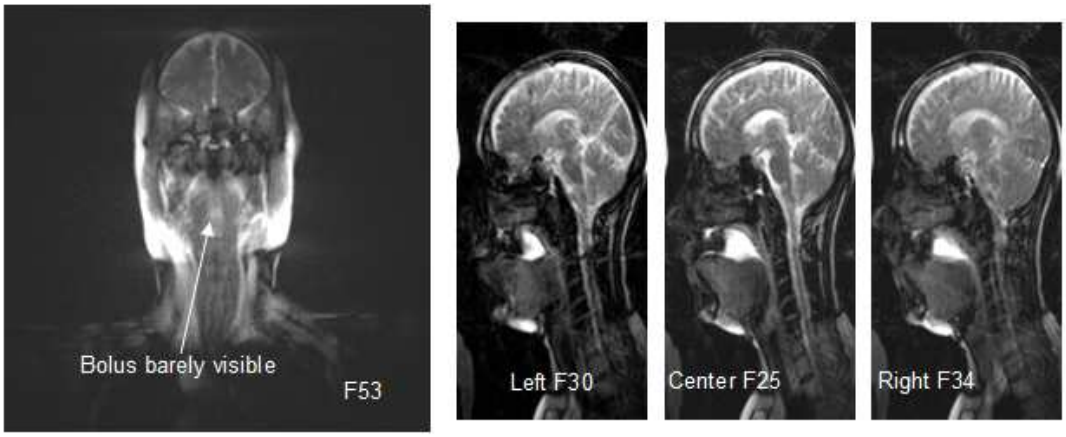


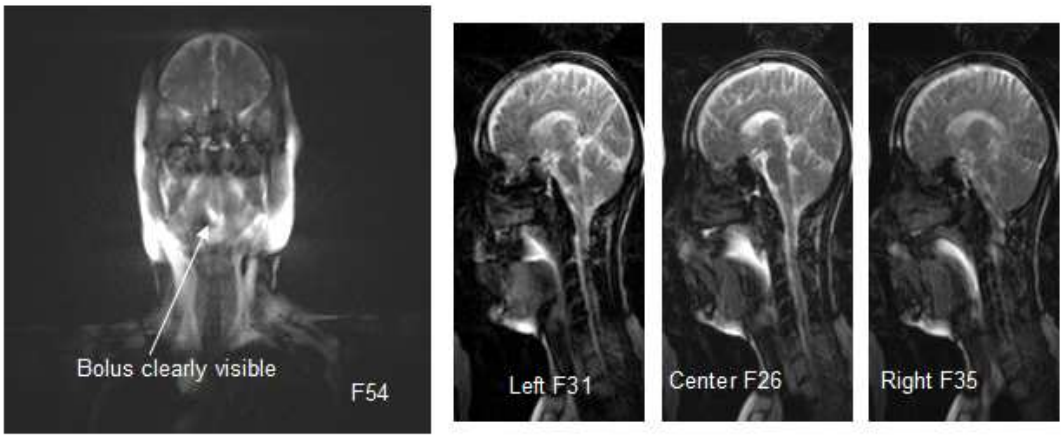
Figure 5.9: Frame number correspondence computed between sagittal and coronal sequences.

Table 5.2: Pseudocode for computing Cframe

$avCI = \frac{1}{n} \sum_k \sum_{x,y} I(x,y,k) \cap cROI$ for $k = 1..n$
% where n is the first few frames of the sequence
*% **and** returns only that region of I where the mask cROI is 1*
For k=1:K
  $CI(k) = \sum_{x,y} I(x,y,k) \cap cROI$
  $CIdiff(k) = |CI(k) - AvCI|$
End
$[val, Cframe] =$ **max**$(CIdiff)$
*% **max** returns the maximum value in the array and also the*
% location where the value occurred

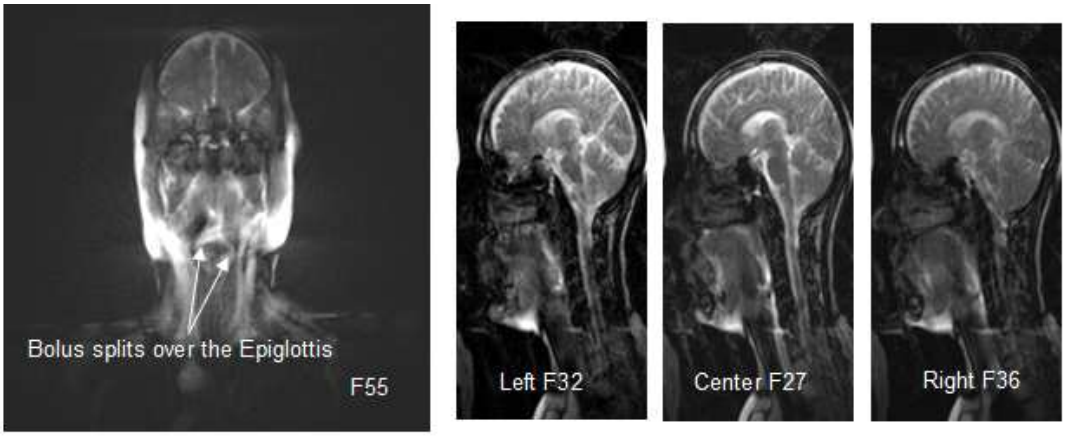


(a)

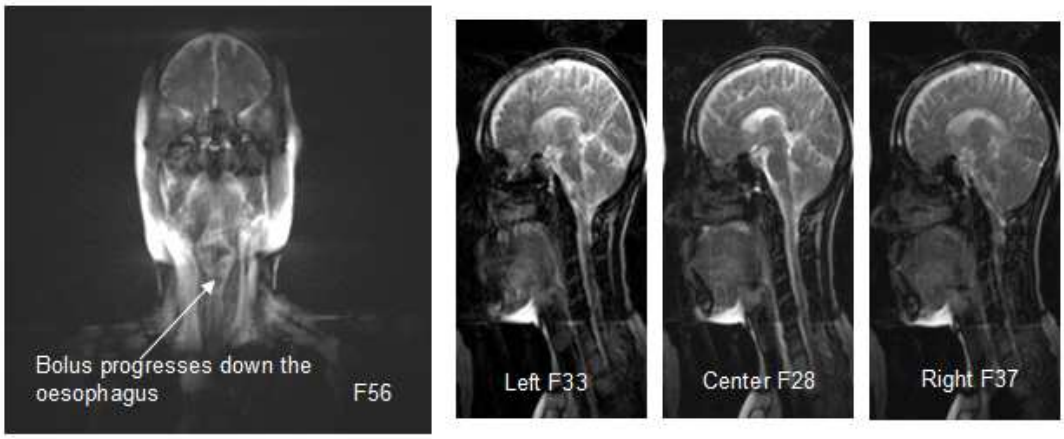


(b)

Figure 5.10: A section of the MRI sequences indicating the alignment result. Corresponding frames in the coronal and sagittal planes when (a) the soft palate has been pushed up and the bolus is ready to descend into the pharynx, (b) the epiglottis has descended and the leading edge of the bolus has reached the epiglottis, (c)-(d) continued in Fig. 5.11.



(c)



(d)

Figure 5.11: A section of the MRI sequences indicating the alignment result. Corresponding frames in the coronal and sagittal planes when (c) the bolus splits over the epiglottis and (d) the bolus begins its descend into the oesophagus.

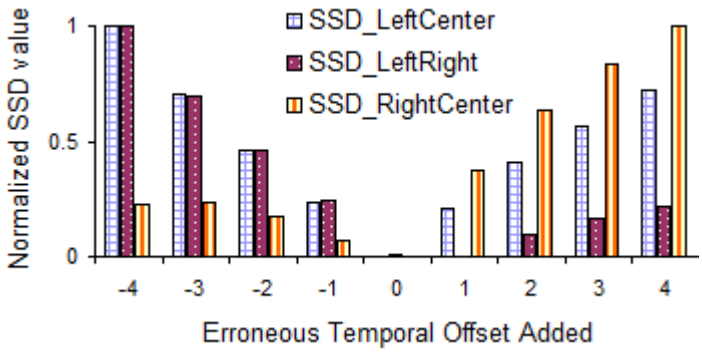


Figure 5.12: Verification of results using SSD metric. The minimum error at offset=0 indicates the accuracy of our method.

frames are shown in Appendix B, Section B.3. A section of the computed sequence alignment is presented in Fig. 5.10, where four corresponding images from each of the dynamic sequences are shown.

5.3.1 Verification

In addition to a visual assessment of the alignment results (shown in Fig. 5.10), we also verify the results using sum of squared differences (SSD) verification as proposed by Werner *et al.* [15]. Our method computes the alignment between the orthogonal planes, *i.e.* each sagittal sequence is aligned to the coronal sequence; the sagittal sequences are not aligned directly with respect to each other, but rather as a consequence of their alignment to the coronal sequence. Thus, the efficacy of our method can be assessed by evaluating the sagittal alignment. If the alignment between the sagittal sequences is accurate then the SSD metric between the corresponding frames should be smaller than that for misaligned sequences. Note that since the sagittal sequences are from different slices the SSD values will never be zero, but they will be minimal for the correct alignment. We add an erroneous time offset (in the range of ± 4 frames) to our computed alignment, and measure the SSD over all the frames thus aligned. The result of the SSD verification are shown in Fig. 5.12, where the SSD values have been normalize to $[0, 1]$. The horizontal

axis represents the erroneous temporal offset added to the sequence alignment, and the vertical axis represents the normalized SSD measure. It can be seen from Fig. 5.12 that the minimum SSD value indeed occurs when the error added to the computed temporal offset is 0, which indicates the alignment computed by our method.

5.4 Summary

In this chapter, we presented a method to align bidirectional dynamic MRI sequences for 4D visualization. We identified regions of motion common to the bidirectional imaging planes and computed an intensity profile of these regions of interest. The frames that correspond to the maxima in the intensity profiles are used to align the video sequences for further visualization. Interestingly, aligning the sagittal sequences to a single coronal sequence resulted in an accurate alignment between the sagittal sequences themselves.

Chapter 6

A Confidence Measure to Choose Low Resolution Sequences in SR Imaging

In the previous chapters, Chapter 3–Chapter 5, we developed various algorithms for video super-resolution (SR) imaging. We used all available low resolution sequences to assess registration and reconstruction quality. However, we did not consider the scenario that not all sequences will contribute positively towards SR reconstruction. In the absence of time-stamp information registration between multiple video sequences is estimated using optimization techniques. Therefore, not all computed registration is reliable and registration accuracy declines with decrease in acquisition rates. In addition, not all sequences contribute useful information towards reconstruction from multiple non-uniformly distributed sample sets. The objectives of the work in this chapter are: (i) to choose the low resolution sample sets that should be combined in order to maximize reconstruction accuracy and (ii) to minimize the number of sample sets needed to achieve that level of accuracy.

This chapter is organized as follows. We present a general overview of the problem in Section 6.1. In Section 6.2, we present the pre-processing proposed required in order to compute the confidence measure and iterative ranking. In Section 6.3 we present our confidence measure (along with a detailed discussion of various influencing factors) and an iterative greedy rank based reconstruction method.

Evaluation of the confidence measure and ranking algorithm with 1D (synthetic and audio) and 2D (real video and MRI) data is presented in Section 6.4. Lastly, conclusions of this work are presented in Section 6.5.

6.1 Introduction to the Problem

Most temporal registration techniques are based on optimizations, such as linear least squares [57] and variational analysis [55]. In the absence of time-stamp information, temporal registration computed using the aforementioned methods is only a best estimate of the actual registration. Errors in feature extraction and tracking, which are often the preliminary steps in registration, can also increase the inaccuracy of the computed registration. It is intuitive that erroneously registered video sequences lead to poor reconstruction. While various methods have been developed in the past to compute temporal registration and reconstruct SR video sequences, none of these methods address the issue that not all computed registration will be accurate and also not all video sequences will contribute useful information for reconstruction purposes.

In order to address this issue we use concepts developed in the field of *recurrent non-uniform sample* (RNUS) reconstruction, which has been reviewed in Section 2.12 (see Chapter 2). Although RNUS was developed for applications where accurate time stamp information is available and it is assumed that the sample sets are from the same continuous time signal (which does not always hold true for SR reconstruction), it still provides useful insight into some of the factors that must be considered for SR reconstruction.

The work proposed in this chapter is unique as it introduces the concept of a confidence measure in temporal registration and reconstruction from recurrent non-uniform samples. The formulation criterion for the confidence measure is two fold – (1) it provides an estimate of how much confidence we have in the registration

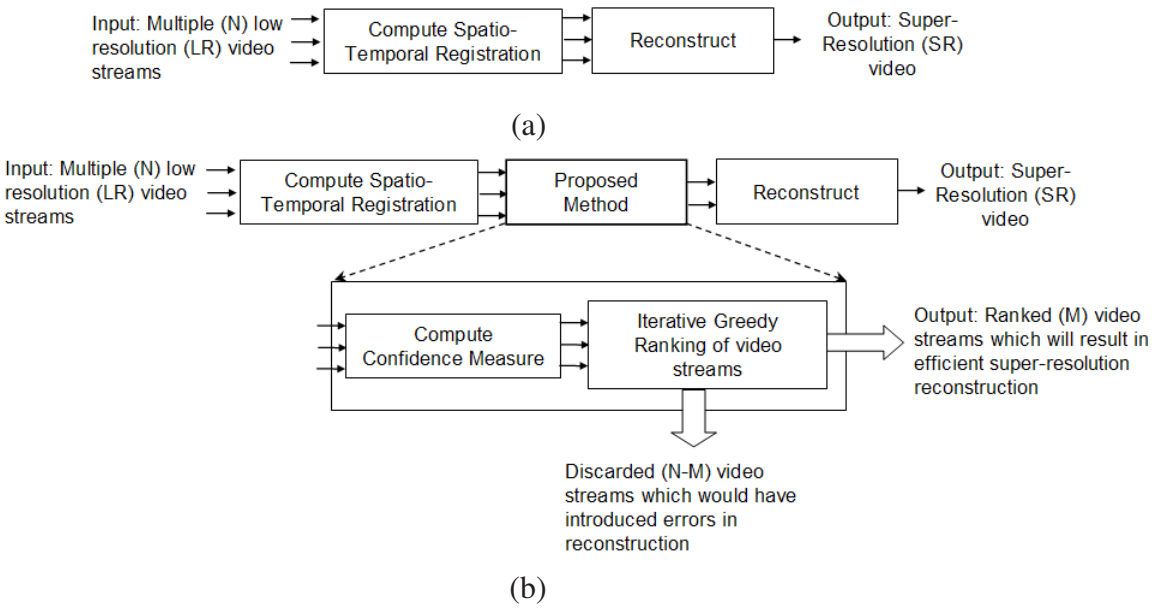


Figure 6.1: Simplified flowcharts of (a) standard SR reconstruction process, (b) enhanced SR reconstruction process based on computed confidence measure and iterative greedy ranking algorithm.

and (2) it also provides an estimate of how much new information is added to the reconstruction process by the inclusion of a particular sample set. We also present an iterative ranking method that not only prioritizes the sample sets, but given that some registration may be inaccurate, it also introduces a threshold limit beyond which adding more sample sets becomes redundant. These concepts are presented in detail in the following sections.

In standard SR reconstruction, as shown in Fig. 6.1(a), all available input sequences are registered and a HR sequence is reconstructed. One of the motivations behind this work is to develop an enhanced SR system, as shown in Fig. 6.1(b), which receives multiple low resolution videos of the same (or similar) scene as input and delivers as output a ranking of sequences which should be used for reconstruction. The system discards those sequences which either do not provide any new information for reconstruction or whose registration is unreliable. 'Spatio-Temporal Registration', in Fig. 6.1(b), can be considered to be the pre-processing

step performed before the proposed method, and 'Reconstruction' can be considered to be the post-processing step. Within the 'Proposed Method', there are two main modules – computation of a confidence measure and an iterative greedy ranking algorithm. We discuss these system modules next.

6.2 Pre-processing

Prior to SR reconstruction, registration is done either by using all the pixels in the video frames (which can be computationally expensive) or by extracting feature and using feature trajectories for alignment. Let $\{S_i,\ 1 \leq i \leq N\}$, denote N video sequences that are acquired at a constant frame rate and are offset from each other by a random time interval τ_n. Each sequence S_i has M frames (I) such that $I_{i,k}$ denotes the k^{th} frame of the i^{th} video sequence. Features are extracted in all sequences to generate discrete trajectories $\Omega_{i,k,p}$, $1 \leq p \leq P$, where P is the number of features extracted.

Features can be extracted based on point characteristics such as corners [24] [47] or based on region characteristics such as shape and color [39] or a combination of both. The video sequences we have used to test our method have a single predominant region that exhibits motion. Thus, in this work we implement a region based feature extraction method where we extract a single blob region based on motion and color information and use the centroid of the blob as a feature. The centroid of the blob represents the average motion for all the pixels in the blob. If multiple features are extracted, they can be tracked using algorithms such as the KLT tracker [43], Kalman Filter [81] or the RANSAC algorithm [18] to generate feature trajectories. For the sake of brevity in the following discussion, we will ignore the subscript p and assume that $\Omega_{i,k}$ refers to all the extracted features.

On their own the feature trajectories are discrete representations of an event or activity in the scene, and we need to interpolate between the discrete representations

for sub-frame registration. An efficient approach to generate continuous representations of the discrete trajectories is to generate event models. We apply the concept of Event Models, presented in Chapter 3, to build a continuous time event model $\Omega_{i,t}$ of the discrete feature space $\Omega_{i,k}$ as follows:

$$\Omega_{i,t} = \Omega_{i,k}\beta_i + \epsilon_i, \tag{6.1}$$

where β_i is the regression parameter and ϵ_i is the model error term. An approximate regression parameter $\hat{\beta}_i$ is iteratively computed such that the following weighted residual error is minimized:

$$\epsilon_i = \sum_{k=1}^{M} w_k ||\Omega_{i,t} - \hat{\Omega}_{i,t}||^2 : t = k, \hat{\Omega}_{i,t} = \Omega_{i,k}\hat{\beta}_i, \tag{6.2}$$

where $||.||$ represents the norm. The method of computing weights w_k is described in Appendix A. Using event models [57] results in a more accurate estimate of the subframe temporal offset compared to the commonly used linear interpolation approach. Once the event models $\Omega_{i,t}$ are available, the temporal offset (τ_n) between the i^{th} and j^{th} sequences is computed by minimizing the following function:

$$\tau_n = [\underset{\tau_n}{\operatorname{argmin}} \sum_{t} ||\Omega_{i,t} - \Omega_{j,t+\tau_n}||^2.] : i \neq j \tag{6.3}$$

The above minimization formulation deals with event models derived from the entire sequence length and therefore results in more accurate estimation of the offset τ_n.

6.3 Proposed Method

In the following sections we first discuss the various factors that influence reconstruction from multiple sample sets. We then present an algorithm to compute a confidence measure which is representative of these factors, followed by an algorithm to iteratively rank the sequences.

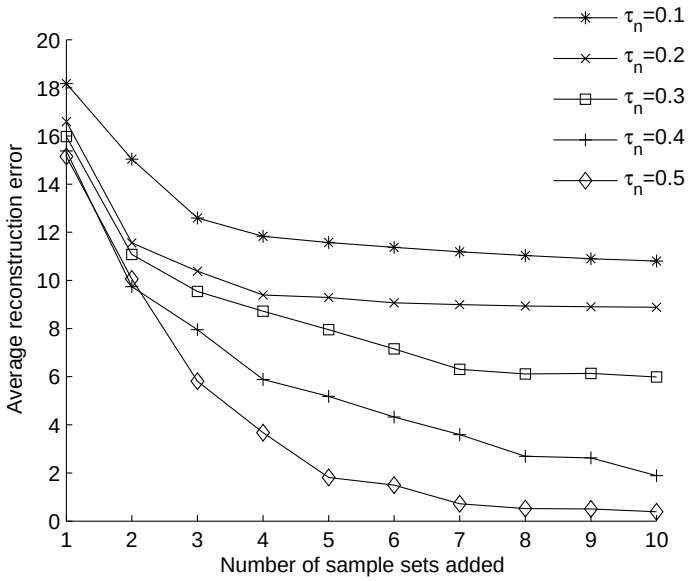


Figure 6.2: Illustration of decrease in reconstruction error with increase in τ_n (reported as a normalized number $[0, 1]$, where 1 corresponds to the sampling rate T).

6.3.1 Factors Affecting Sample Confidence

When reconstructing signals from multiple sample sets, two factors need to be kept in mind: (i) the uniformity (or lack thereof) of sample data and (ii) the accuracy with which the datasets have been registered. In the following sections we look at the influence of both these factors on the development of an efficient approach to super-resolution registration.

Non-uniformity of Sample Sets

Consider the reconstruction formulation discussed in Section 2.12, and the system of linear equations Eq. 2.31 which can be used to derive approximations of samples at uniform instances from the known non-uniform samples. Due to a finite window of reconstruction, the system of equations represents only an approximate linear relation between uniform and non-uniform samples, and this approximation can result in an ill-conditioned linear system. Also, if the non-uniform sampling instances are close to each other, then due to finite precision, round off error or er-

roneous computation of the offset τ_n, the system of equations can become singular. By maximizing the row-wise difference between the coefficients of matrix A in Eq. 2.31 we can reduce the chances of A becoming ill-conditioned or singular. Maximizing the difference between the coefficients of A translates into maximizing the distance between the closest sampling time instances of the i^{th} and j^{th} recurrent non-uniform sample sets as follows:

$$maximize[(t'_i - t_0) - (t'_j - t_0)] : i \neq j,$$
$$maximize[t'_i - t'_j] \Rightarrow maximize[\tau_{ij}]. \tag{6.4}$$

One interpretation of Eq. 6.4 is that for optimal reconstruction, the sampling instances of the recurrent sample sets should be as far away from each other as possible. Intuitively, without any *a priori* information about the signal, this will allow for sampling of the major trends in a signal. Proponents of non-uniform sampling will argue that sampling such that higher number of samples are taken in high frequency regions would be an optimal sampling approach. However, note that most acquisition methods have fixed sampling rates with little user control over the sampling process. The validity of criterion in (Eq. 6.4) can be demonstrated experimentally as follows. We generate random HR signals bandlimited to a user-controlled frequency (by applying a low-pass filter), and sample them to create multiple LR sample sets. We assume that there is no temporal registration error and the location of the sample sets with respect to each other is known accurately. We add up to 10 sample sets iteratively during reconstruction, repeating the experiment with 100 signals with different values of temporal offset τ_n. The decrease in the reconstruction error as more and more sample sets are combined (for different values of τ_n) is shown in Fig. 6.2.

In Fig. 6.2 the horizontal axis represents the number of samples sets that were used for reconstruction and the vertical axis represents the average reconstruction error computed between the reconstructed signal and the actual signal. Consider the

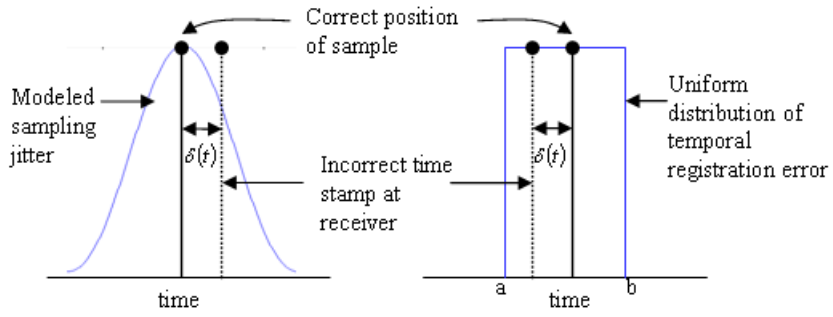


Figure 6.3: Modeling the error in temporal registration as a (a) Gaussian distribution, (b) Uniform distribution.

reconstruction error for a temporal offset of $\tau_n = 0.5$. This temporal offset implies that the sample locations were approximately midway from each other. It can be seen that the reconstruction error for this offset decreases at the largest rate as more and more sample sets are used in the reconstruction, as compared to other offset values. Thus sample sets that have large offsets between each other result in lower reconstruction errors at lower number of sample sets added, when compared to sample sets that have smaller offset from each other. This implies that being able to measure the non-uniformity between sample sets is important when reconstructing from RNUS.

Error in temporal registration

Computing the temporal offset between sample sets is a non-trivial task and there is possibility of error in the computation. A similar problem of estimating error in computation of sampling instances is encountered in digital communication. The error in actual sampling time instances and computed sampling time instances is termed as sampling jitter [83], and is often modeled as Gaussian distribution, as shown in Fig. 6.3(a). If the sampling instance in a communication system is denoted by τ, then assuming a Gaussian distribution, the probability density function of the

jitter can be expressed as:

$$\delta(\tau) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(\tau-\mu)^2}{2\sigma^2}), \tag{6.5}$$

where mean μ is the expected arrival time instance and σ^2 is the variance expected at the receiver. However, in our case we model the error in temporal registration as a uniform distribution, as shown in Fig. 6.3(b), with a range $[a, b]$ as follows:

$$\delta(\tau) = \begin{cases} 0 & \text{for} \quad \tau < a \\ \frac{1}{b-a} & \text{for} \quad a \leq \tau \leq b \\ 0 & \text{for} \quad \tau > b \end{cases} \tag{6.6}$$

Thus, if actual temporal offset between two sample sets is τ, due to error $\delta(\tau)$ in computing the offset, the offset used for reconstruction is $\tau + \delta(\tau)$. Experimentally we found that there are two effects of error in temporal registration, as illustrated in Fig. 6.4.

- As the error increases, more and more sample sets are needed to achieve the same reconstruction efficiency as with fewer more accurately registered sample sets.

- For a given distribution of error, there exists a threshold number of sample sets beyond which adding more sample sets does not improve the reconstruction error, and adding more sample sets is redundant. In some cases, adding more sample sets can deteriorate the reconstruction accuracy.

Figure 6.4 plots the average reconstruction error (Squared Error SE) vs. the number of sample sets used for reconstruction for temporal registration errors ranging from 5% to 55%. It can be seen that for the experimental sample sets, when the error in temporal registration is above 25%, adding more sample sets does not improve the reconstruction results, and infact the reconstruction results deteriorate. Since error in temporal registration determines a threshold limit to the number of sample sets needed to achieve a certain reconstruction efficiency, we need to include a suitable representation of this error in our confidence measure.

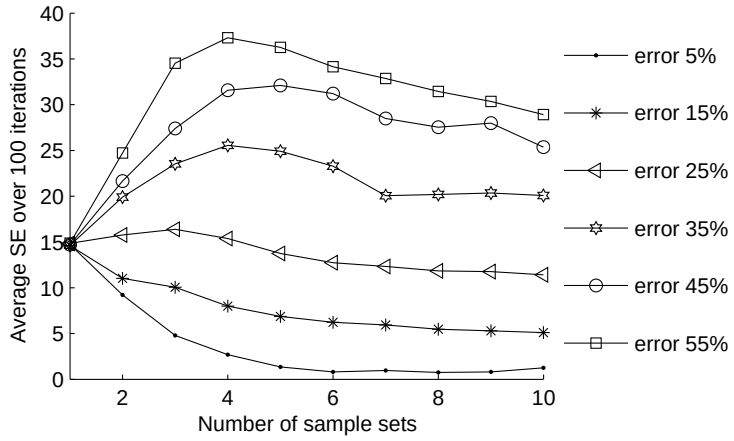


Figure 6.4: Effects of error in temporal registration on reconstruction.

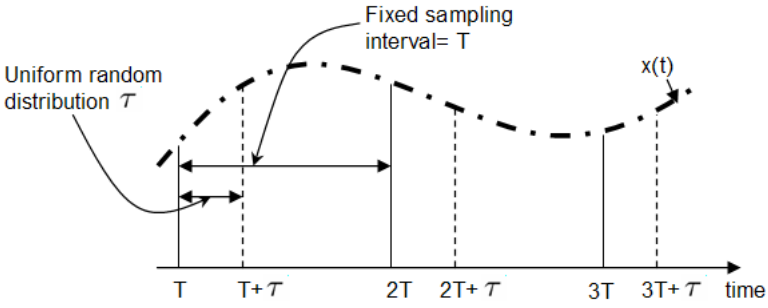


Figure 6.5: Illustration of recurrent non-uniform sampling with two sample sets.

6.3.2 Computing the Confidence Measure

We have discussed two factors related to recurrent non-uniform samples that affect the reconstruction process. Due to lack of correct time-stamp information we can neither accurately determine the non-uniformity of sample sets, nor the error in temporal registration. We can, however, determine other parameters which are indicative of non-uniformity and temporal registration error. We define two such parameters in the form of objective functions Φ_g and Φ_l, which are presented next. Given two sample sets $x(kT)$ and $x(kT + \tau_n)$ (as shown in Fig. 6.5) and their respective feature space $\Omega_{i,k,p}$ (defined in Section 6.2), we define an objective function

that estimates the non-uniformity of the sample using the following equation:

$$\Phi_g = \sum_{p}^{P} \sum_{k=1}^{M} (\|\Omega_{i,kT,p} - \Omega_{j,kT+\tau_n,p}\|^2) : i, j \in (1..N). \tag{6.7}$$

Intuitively, Φ_g represents the global registration error of the discrete trajectories. Discrete samples that are closer to each other have relatively smaller differences in sample values as compared to samples that are farther apart in time from each other. Thus the formulation of Φ_g (in Eq. 6.7), as a sum of difference between the discrete feature trajectories (after they have been approximately registered), reflects how far the sample set are in time.

We also define the following objective function that estimates the error in temporal registration (subsequent to computing the continuous event models $\hat{\Omega}_{i,t}$):

$$\Phi_l = \sum_{p}^{P} \sum_{t} (\|\hat{\Omega}_{i,t,p} - \hat{\Omega}_{j,t+\tau_n,p}\|^2) : i, j \in (1..N). \tag{6.8}$$

Intuitively, Φ_l represents the offset-compensated registration error of the event models, and event models that have been incorrectly registered result in larger values of Φ_l. The objective functions, defined in Eqs. 6.7 and 6.8, give a general idea of the confidence in the temporal registration. However, they do not relate to the confidence by a simple proportionality, *i.e.* a large Φ_g does not imply a poor confidence in the registration. It has been experimentally observed that a large Φ_g indicates more uniform distribution of the sample sets and a small Φ_l indicates better registration of offset compensated signals, hence better signal reconstruction or higher confidence in the choice of sample sets. Therefore, we define the confidence measure as a linear weighted sum of Φ_g and Φ_l as follows:

$$\chi = w_g(\Phi_g)^q + w_l(\Phi_l)^r, \tag{6.9}$$

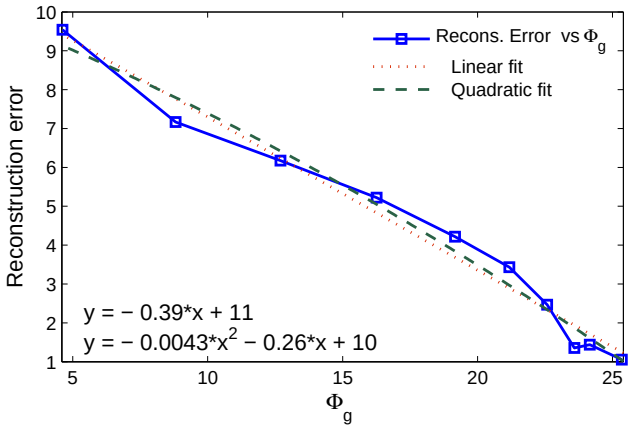
where w_g and w_l are weights assigned to the contribution of both the objective functions to the overall confidence measure. Relational parameters q and r, that define whether the objective functions and the confidence measure are directly or

inversely related, are examined shortly. A method to compute the weights is discussed later in this section. We now present two hypotheses with respect to Φ_g and Φ_l which are intuitively supported by our discussion in Section 6.3.1 and supported by experimental validation that follows.

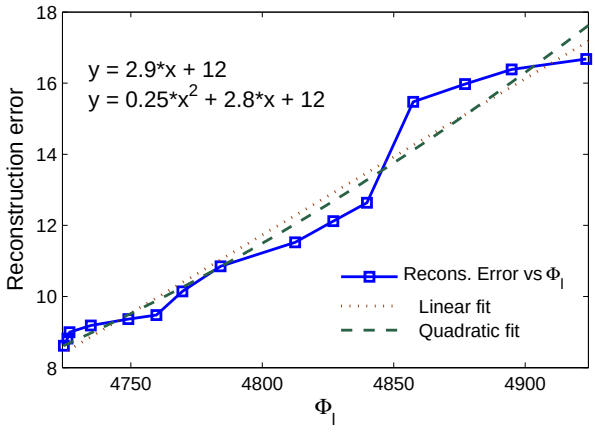
Hypothesis 1: Φ_g is an indicator of τ_n , and a value of τ_n which places the sample sets as far apart from each other as possible, results in a better reconstruction. Hence an increase in Φ_g should increase the confidence measure.

Hypothesis 2: Φ_l is an indicator of the overall error in registration, and a large Φ_l results in poorer reconstruction. Hence an increase in Φ_l should decrease the confidence measure.

We validate the above hypotheses on 1D synthetic data. For each experiment a pseudo-random high resolution (HR) 1D signal is generated at a user specified bandwidth using the 'rand' function in MATLAB. LR sample sets are generated by sampling the HR signal with a fixed sampling rate and a uniformly distributed temporal offset τ_n. Φ_g and Φ_l are computed for various combinations of the LR sample sets . An approximate HR signal is also reconstructed from the LR combinations using code provided in [45] for Feichtinger's algorithm [17]. Reconstruction error is computed as the sum of squared error (SSE) between the reconstructed signal and the original signal. Figure 6.6(a) plots the reconstruction error versus Φ_g computed for the synthetic test signals. It can be seen that Φ_g demonstrates a linear relationship with the reconstruction error: as Φ_g increases the reconstruction error decreases, *i.e.* the confidence measure which should be associated with Φ_g should be in direct increasing proportion. Therefore 'q' in (Eq. 6.9) can be approximated to '1'. We also fitted the reconstruction error versus Φ_g curve with a quadratic function and it can be seen from Fig. 6.6(a) that a linear fit is a sufficiently good approximation of the curve. The same analysis is applied to the relationship between the reconstruction error and Φ_l, which is shown in Fig. 6.6(b). It is observed that



(a)



(b)

Figure 6.6: (a) Relationship between reconstruction error and objective function Φ_g. (b) Relationship between reconstruction error and objective function Φ_l.

as Φ_l increases, the reconstruction error increases. Hence, 'r' in (Eq. 6.9) can be approximated to '-1'. The scale of the values of Φ_g and Φ_l^{-1} are different, hence they are normalized to lie between [0,1]. The normalization of Φ_g (and similarly Φ_l^{-1}) is computed as follows:

$$\widetilde{\Phi_g} = \frac{\Phi_g - min(\Phi_g)}{[max(\Phi_g) - min(\Phi_g)]} \tag{6.10}$$

The proposed confidence measure χ can therefore be expressed as follows:

$$\chi = w_g\widetilde{\Phi_g} + w_l\widetilde{\Phi_l}^{-1} \tag{6.11}$$

Computing Weights w_g and w_l

Ideally, we want the confidence measure to *linearly* increase with a decrease in the associated reconstruction error. This requirement is a key factor in determining the weights (w_g, w_l) that control the effect of sampling non-uniformity and offset estimation. Suppose we could hypothetically arrange the sample sets in increasing order of reconstruction error as shown in Fig. 6.7, where the horizontal axis represents combinations of sample sets and the vertical axis corresponds to normalized values of confidence measure. The confidence measure (χ) computed using arbitrary values of weights may not be a linear curve. For example, Fig. 6.7 illustrates confidence measure values with two sets of weights, $\chi(w_g1, w_l1)$ and $\chi(w_g2, w_l2)$. Note that ideally we want the confidence measure to relate linearly to the reconstruction error, therefore the goal of tuning the weights is to reduce residual of a linear fit of the confidence measure. In Fig. 6.7 for example, weights w_g2, w_l2 result in smaller residual error in the linear estimation of χ as compared to w_g1, w_l1, and are hence more suited for computing the confidence measure.

In reality, we cannot estimate the reconstruction error of a set of samples, since the original signal is not available to us. However, the objective functions Φ_g and Φ_l can be computed and as validated previously, these functions are linearly related

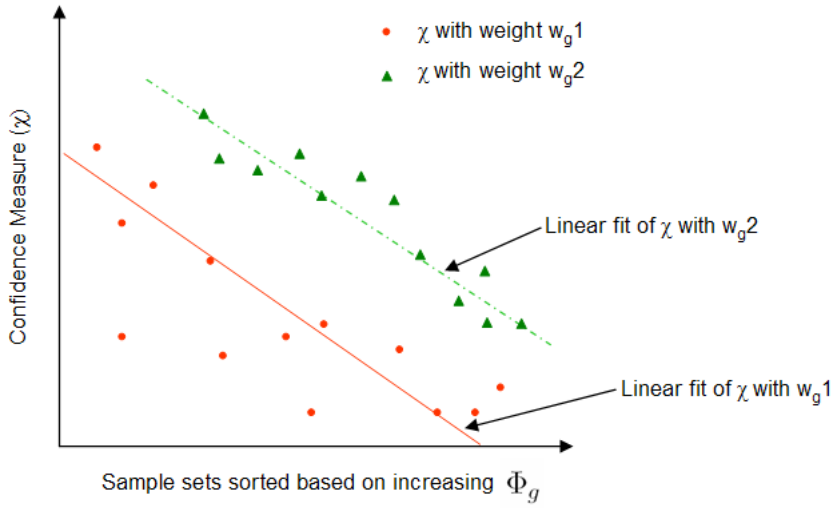


Figure 6.7: Confidence measure values computed for two different set of weights (w_g1 and w_g2). Linear fit for w_g2 results in a smaller residual.

to the reconstruction error. An additional assumption that is made with respect to the weights is that they sum to unity, *i.e.* $w_g + w_l = 1$. A pseudocode for computing optimal weights is presented in Table 6.1. Let there be N sample sets of an event from which we compute Φ_g and Φ_l for all ${}^{\mathrm{N}}\mathrm{C}_2$ pairs of sample sets. We sort the sample set pairs based on either Φ_g or Φ_l (we found experimentally that the choice of the objective function does not affect the computation of optimal weights). For incremental increases Δw_g in the value of w_g ($w_l = 1 - w_g$), we iterate over steps (i)-(iv) described in Table 6.1. The weight corresponding to the minimal residual value is chosen as the optimal weight.

In order to test the proposed weight optimization strategy we generated five sample sets from a high resolution signal. We took ten combinations (${}^5\mathrm{C}_2$) of these samplesets and reconstructed an estimate of the original signal. The aim of the experiment was to optimize weights (based on the linearization strategy discussed) such that the *highest* confidence measure for the combinations corresponds to the lowest reconstruction error. The results for four such test cases, with optimized

Table 6.1: Pseudocode for computing optimal weights.

For $w_g = 0 : \Delta w_g : 1$

(i) Compute $\chi(w_g) = w_g\widetilde{\Phi_g} + w_l\widetilde{\Phi_l}^{-1} \Rightarrow w_g\widetilde{\Phi_g} + (1 - w_g)\widetilde{\Phi_l}^{-1}$.

(ii) Perform a quick sort based on either Φ_g or Φ_l.

(iii) Construct a linear fit to estimate $\hat{\chi}(w_g)$.

(iv) Compute the residual of the linear fit as: $\mathcal{R}(w_g) = ||\hat{\chi}(w_g) - \chi(w_g)||^2$.

End

(v) Optimize weights as: $w_g^{opt} = \underset{w_g}{\operatorname{argmin}}\, \mathcal{R}(w_g) = \underset{w_g}{\operatorname{argmin}}\, ||\hat{\chi}(w_g) - \chi(w_g)||^2$

(vi) Compute $w_l^{opt} = 1 - w_g^{opt}$.

Table 6.2: Reconstruction error values for optimized and sub-optimal weights w_g and $w_l = 1 - w_g$.

	Weight	Category	χ	SSE
Case1	$w_g = 0.8$	optimal	0.711	24.08
	$w_g = 0.5$	suboptimal	0.462	620.1
Case2	$w_g = 0.7$	optimal	0.625	186.3
	$w_g = 0.3$	suboptimal	0.649	712.2
Case3	$w_g = 0.8$	optimal	0.731	112.4
	$w_g = 0.5$	suboptimal	0.497	986.0
Case4	$w_g = 0.9$	optimal	0.852	29.17
	$w_g = 0.5$	suboptimal	0.571	717.6

and randomly chosen sub-optimal weights, are shown in Table 6.2. The highest confidence measure and the corresponding reconstruction error are also presented. It can be seen that optimizing the weights results in the desired correspondence between a high confidence measure and low reconstruction error, whereas, the same relationship does not hold true for sub-optimal weights in the confidence measure.

6.3.3 Iterative Rank-based Method

In reconstructing from multiple sample sets we need to order the sample sets such that the information added for reconstruction is maximized and the error in the reconstruction is minimized. This can be accomplished by ranking the multiple sample sets based on the proposed confidence measure. We use ranking instead of

directly using the numerical confidence measure scores as the scales of the confidence may change over each iteration, while ranking is a more consistent relative measure of the confidence measure. We assume that in each iteration the number of distinct ranks decreases by 1. In practice, however, confidence measure scores may result in ties. In such cases a weighted measure of the previous rank score can be added to the current rank to break the tie. This weighted addition of the previous rank incorporates prior rank information rather than arbitrarily choosing one sample set over another.

A flowchart of the iterative rank based reconstruction (IRBR) algorithm is shown in Fig. 6.8. The IRBR method is implemented as a greedy algorithm. In the first iteration the algorithm computes confidence measures between all possible combinations of two sample sets. Sample set combinations are then ranked based on the confidence measure. The sample set combination which has the highest confidence measure is combined to reconstruct the first sample set of a new sample set array. The remaining sample sets are then added to this new sample set array in no particular order. Next, confidence measures are computed between the reconstructed sample set from the previous iteration and all other remaining sample sets in the current iteration. This step is what defines IRBR as a greedy algorithm, since the reconstruction error minima which was computed in the very first iteration determines the path that the following iterations take. If absolute difference between the signal reconstructed at the current iteration and previous iteration is less than a threshold (empirically determined), or if all the sample sets have been combined, the iterations are stopped.

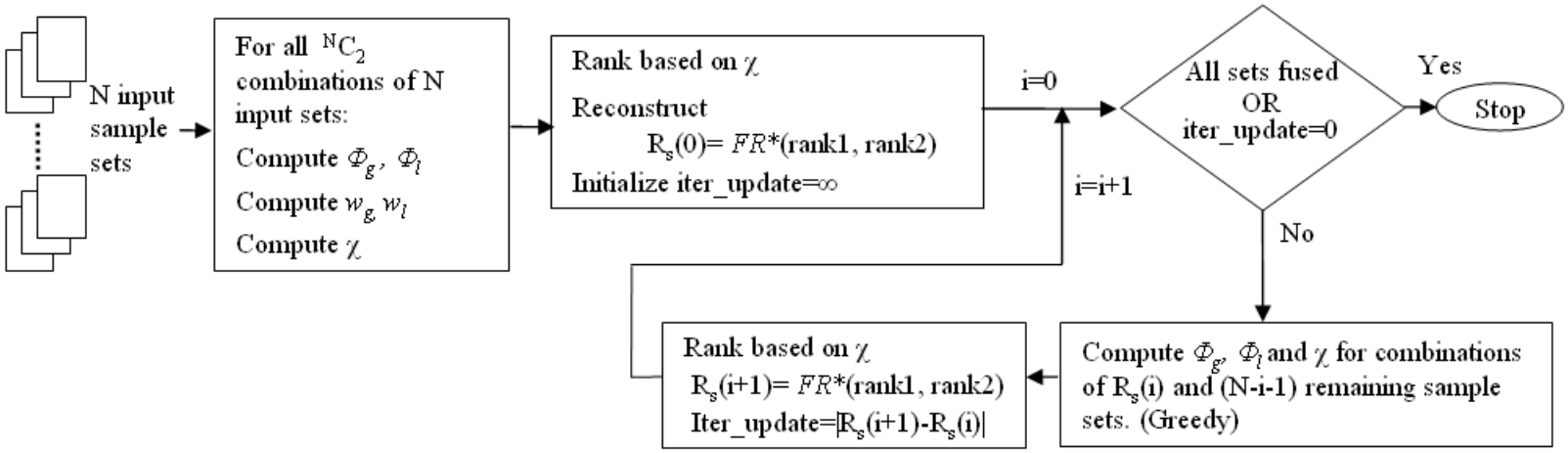


Figure 6.8: Flowchart of IRBR method based on the proposed confidence measure. FR* indicates a RNUS reconstruction algorithm from [45] which was reviewed in Chapter 2.

6.4 Performance Evaluation of the Proposed Method

In this section we present the experimental setup and validation of each module of the proposed system. We first evaluate each objective function independently and present a representative result that illustrates why a weighted measure of both the objective functions (Φ_g and Φ_l) is more suitable than using either objective function independently. We then evaluate the confidence measure (χ) with synthetic and real (video) data sets. An evaluation of the iterative ranking algorithm is also presented with synthetic and real (audio) data sets. We also evaluate the proposed approach on low resolution MRI data and demonstrate the improvement in reconstruction. Lastly, we discuss the computational complexity of the system.

6.4.1 Independent Evaluation of Φ_g and Φ_l

In order to understand the complementary nature of Φ_g and Φ_l, we evaluate each objective function independently. We set up our experiments such that six synthetic sample sets are divided into two experimental cases of three sample sets each. The objective of the experiment is to determine which pair of three sample sets (*i.e.* sample1-sample2, sample2-sample3 or sample1-sample3) will result in the minimum reconstruction error when combined. Since the sample sets are generated synthetically, the actual signal is known and the reconstruction error (SSE) can be computed. The SSE error is only used to validate the decisions that we take with respect to the sample set combinations.

In the experiments, we compute Φ_g, Φ_l^{-1} and χ for all possible combinations of sample sets. These values along with the SSE error for sample set combinations in Case 1 and Case 2 are shown in Table 6.3, where values of Φ_g and Φ_l^{-1} are normalized within each test case to lie between $[0, 1]$. If we observe the values of Φ_g for the three combinations in Case 1, and choose the combination corresponding to the highest Φ_g as the best combination (sample1-sample2), that decision will be

Table 6.3: Experimental results for independent evaluation of objective functions Φ_g and Φ_l.

	Combinations	Φ_g	Φ_l^{-1}	χ	SSE
Case1	sample1-sample2	1.0	0.0	0.2	47.70
	sample2-sample3	0.161	0.0165	0.0454	318.85
	sample1-sample3	0	1.0	0.8	31.86
Case2	sample4-sample5	0	1.0	0.2	1469.31
	sample5-sample6	0.8983	0.0066	0.7199	162.33
	sample4-sample6	1.0	0	0.8	148.41

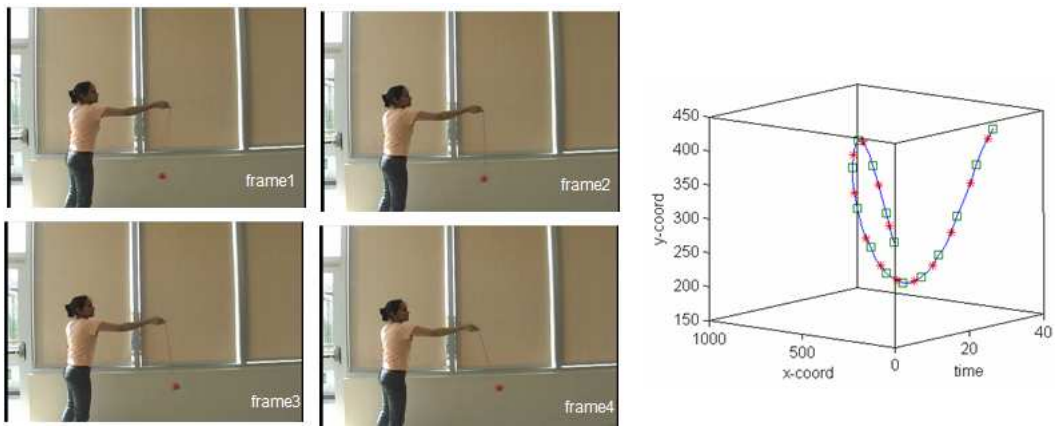
incorrect as sample1-sample2 combination does not correspond to the lowest SSE. However, if we were to choose based on the highest value of Φ_l^{-1}, the decision would be correct. Now, consider the results for Case 2. For Case 2, choosing based on Φ_g will result in the correct choice, while Φ_l^{-1} will result in an incorrect answer. Thus, using only Φ_g or Φ_l^{-1} as a metric to choose sample set combinations results in unreliable decisions. However, it can be seen for both cases that the confidence measure χ accurately determines the best combination in both these cases.

6.4.2 Evaluation of Confidence Measure

We evaluated the proposed confidence measure on both synthetic and real data. Samples of synthetic data and real test videos are presented in Appendix B, Section B.1 and can also be viewed at:

`www.ece.ualberta.ca/~meghna/J2008.html`.

Synthetic data was generated as a high resolution random signal which was band-limited to a user controlled frequency. This high resolution data was then sampled at a low sampling rate. For example, a 25 Hz band-limited signal was sampled at 2 Hz. Multiple sample sets at a fixed low sampling rate were also generated by initializing the starting point of the sample sets randomly with a uniform distribution. Temporal registration was then computed using methods described in Section 6.2. These multiple sample sets were then iteratively fused together, one at



(a) (b)

Figure 6.9: (a) Sample frames from real data sequence, (b) Sample trajectory from real data sequence.

a time, based on the computed time stamp information. Reconstruction algorithm from [17] was used to reconstruct a signal from the fused samples.

For our *real test cases*, we used video sequences of an individual swinging a ball tied to the end of a string. The video sequences were captured at 30 frames a second and the trajectory of the ball was extracted via background subtraction techniques and motion tracking. This trajectory was then used as a high resolution signal which was further down-sampled at low sampling rates, as shown in Fig. 6.9(a-b). An event model was used to compute the temporal registration between the undersampled signals. In each experiment, we arbitrarily chose one sample set as the parent against which other recurrent sample sets were registered. The two objective functions defined in (Eq. 6.7) and (Eq. 6.8), and the confidence measure (Eq. 6.11) for these sample sets were computed. These values along with the reconstruction error (SSE) for both synthetic and real data are presented in Table 6.4. A higher confidence measure indicates that the corresponding recurrent set is a better candidate for reconstruction, as corroborated by the corresponding reconstruction error. It can be seen that the proposed confidence metric is a suitable indicator of the reconstruction error. Further results with MRI data are presented in Section 6.4.4.

Table 6.4: Confidence measure χ and corresponding reconstruction error for synthetic sample sets.

	$\Phi_g 1$	$\Phi_l 1$	$\widetilde{\Phi}_g 1$	$\widetilde{\Phi}_l^{-1} 1$	$\chi 1$	SSE1	$\Phi_g 2$	$\Phi_l 2$	$\widetilde{\Phi}_g 2$	$\widetilde{\Phi}_l^{-1} 2$	$\chi 2$	SSE2
S1	44.0	230.3	1.0	0.0241	0.8048	1.3	35.6	596.7	0.79	0.0	0.6320	2.6
S2	15.7	49.8	0.2959	0.1663	0.2673	4.0	4.0	8.9	0.0	1.0	0.2	13
S3	22.5	374.0	1.0	0.0	0.7	205.4	16.8	212.4	0.0	1.0	0.3	355.3
S4	97.5	2959.0	1.0	0.0	0.7	2.4	55.1	776.0	0.4796	1.0	0.6357	4.3
S5	16.3	2387.7	0.0	0.0851	0.0255	9.7	17.4	2590.6	0.0136	0.0506	0.0247	9.3
S6	26.5	22.8	1	0	0.7	47.3	8.7	20.58	0	1	0.3	68.6
S7	43.7	7081.5	1.0	0.0	0.7	8.9	19.7	2228.7	0.3315	0.1454	0.2757	16.1
S8	7.8	443.4	0.0	1.0	0.3	10.9	28.9	3687.7	0.5877	0.0615	0.4299	10.3
S8	21.6	7409.3	0.3967	0.0	0.2777	8.7	16.8	5970.7	0.0	0.1850	0.0555	9.3
S9	28.9	5823.3	1.0	0.1983	0.7595	7.9	22.3	3121.5	0.4545	1.0	0.6182	10.7
S11	4.6	169.5	0.0	1.0	0.3	15.6	30.3	6190.9	1.0	0.0	0.7	9.9

Table 6.5: Confidence measure χ and corresponding reconstruction error for real video sequences.

Scene	Sequence	Φ_g	Φ_l	$\widetilde{\Phi}_g$	$\widetilde{\Phi}_l^{-1}$	χ	SSE
Scene 1	seq1-seq2	446	41.1	0.5072	0.3943	0.4733	1700
	seq2-seq3	652	22.3	1.0	1	1	420
	seq1-seq3	234	90.3	0	0	0	8120
Scene 2	seq1-seq2	129	4.6	0	1	0.3	20.9
	seq2-seq3	364	321	1	0	0.7	4.0

6.4.3 Evaluation of IRBR Method

Synthetic Test Sequence

We evaluated the rank-based reconstruction system on 100 synthetic signals and one audio signal. The synthetic signals were generated using MATLAB's random number generator function 'rand'. For each synthetic signal, 21 sample sets were created by sampling the original high resolution signal with random initial points. Thus, each signal had 21 recurrent non-uniform sample sets. The IRBR algorithm was used to reconstruct high resolution signal from 21 low resolution samples sets and the averaged results are shown in Fig. 6.10. It can be seen that our ranking algorithm, which utilizes the proposed confidence measure, performs much better than a random ordering of the sample sets during reconstruction. In some cases, the proposed system resulted in lower reconstruction error than all sample sets combined, as illustrated by point-A and point-B in Fig. 6.10.

Audio Test Signal

We also used a section of an audio signal available in the MATLAB demo data as 'toilet.wav' as a 1D test signal. The original audio signal is shown in Fig. 6.11(a). Multiple LR audio files are generated by subsampling the original HR wavefile. Two such LR audio signals are shown in Fig. 6.11(b) and (c). The sampling rate after sub-sampling is much below the Nyquist rate. Sample sets are temporally aligned and reconstructed based on the proposed confidence measure and ranking

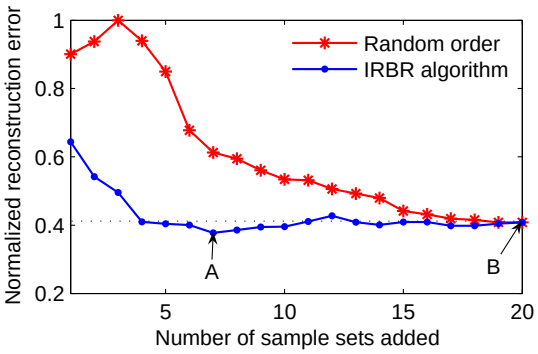


Figure 6.10: Performance of iterative rank based reconstruction (IRBR) algorithm with synthetic data.

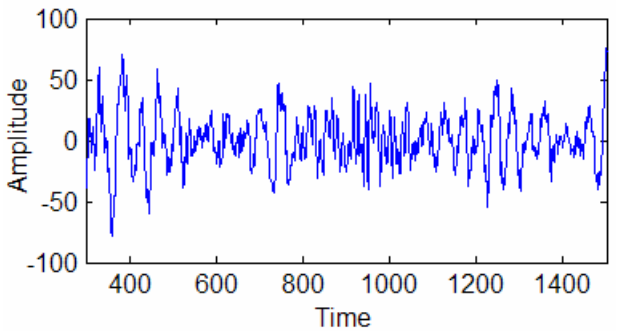
algorithm. Results are shown in Fig. 6.12. It can be seen that the proposed confidence measure and the IRBR method successfully order the audio sample sets such that lesser number of sample sets are needed to reconstruct the same signal, as compared to a random ordering of the audio sample sets.

6.4.4 Evaluation with MR Imaging

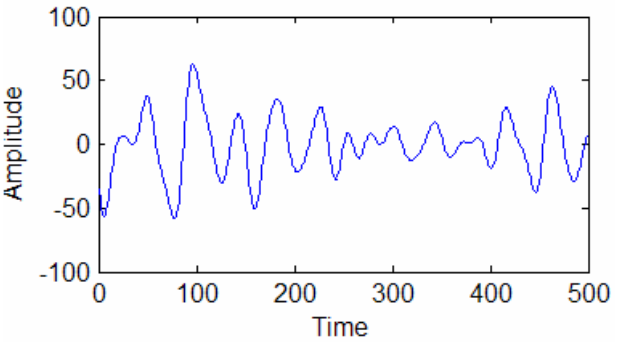
We validate the proposed algorithms by showing that LR MR video combinations with high confidence measure have better SR reconstruction in terms of improvement in both spatial and temporal resolution. A few representative frames of the LR MRI sequences are shown in Fig. 6.13(a)-(b). The following sections detail the processing and reconstruction algorithms used for the experiments.

Feature Extraction in MRI

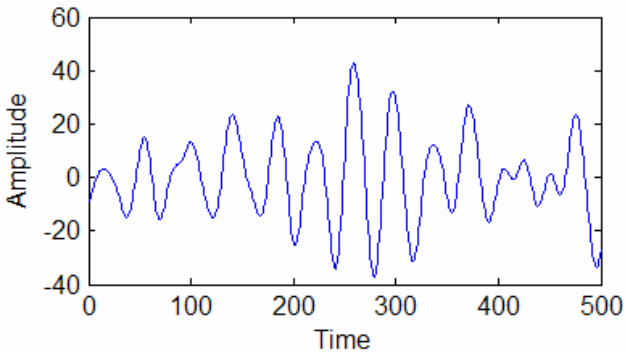
The first few frames of the LR MRI videos are used to spatially register the video frames to each other. This step ensures that any slight movement of the subject in the MRI scanner is compensated for. Next, the progression of the bolus (water) down the oral-pharyngeal tract is segmented using standard background subtraction techniques [3]. Centroid coordinates are computed from this moving blob region to generate feature trajectories for all three LR MRI videos. The multiple centroid



(a)



(b)



(c)

Figure 6.11: (a) Original 'toilet.wav' audio signal, (b)-(c) Two representative sections of the original audio signal that were used in the experiments.

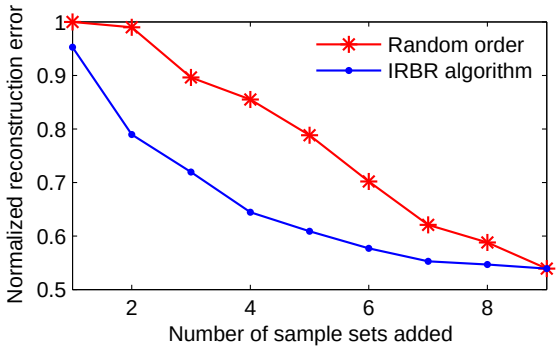


Figure 6.12: Performance of iterative rank based reconstruction (IRBR) algorithm with audio data.

(a) (b) (c)

Figure 6.13: (a) Illustrative frame from LR video2; (b) Closest corresponding frame in LR video3; (c) Intermediate frame reconstructed using (a) and (b).

Table 6.6: SNR values for 4 ROIs in LR and SR video sequences and Confidence measures.

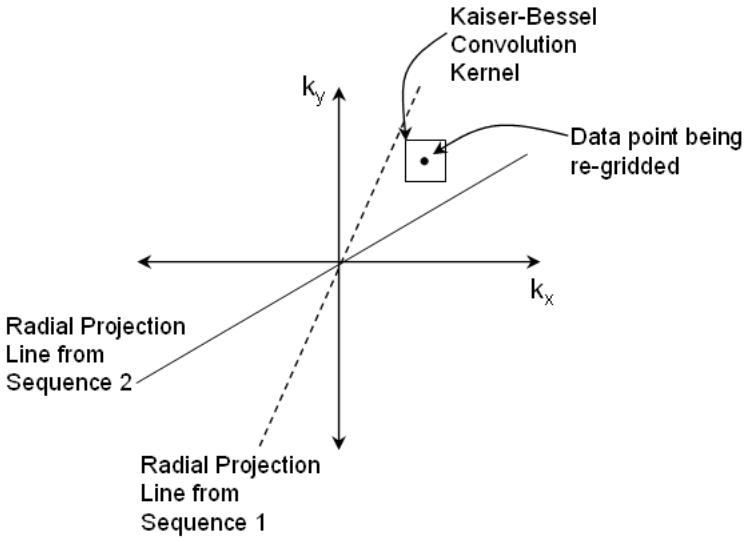
Sequence	SNR				Confidence Measure
	ROI 1	ROI 2	ROI 3	ROI 4	
vid1	7.4419	17.445	19.576	27.329	
vid2	6.6593	14.794	15.642	36.559	
vid3	6.1011	15.472	16.683	29.99	
vid1-vid2	10.672	24.478	39.766	58.35	27.64
vid2-vid3	20.257	95.31	97.632	154.44	38.70
vid1-vid3	13.645	27.96	39.785	69.543	28.15

trajectories can be considered to be LR sample sets acquired from the same continuous event – swallow. These centroid trajectories are used to compute the temporal registration and also the confidence measure between the LR videos. The confidence measures computed between the three LR sequences are presented in Table 6.6.

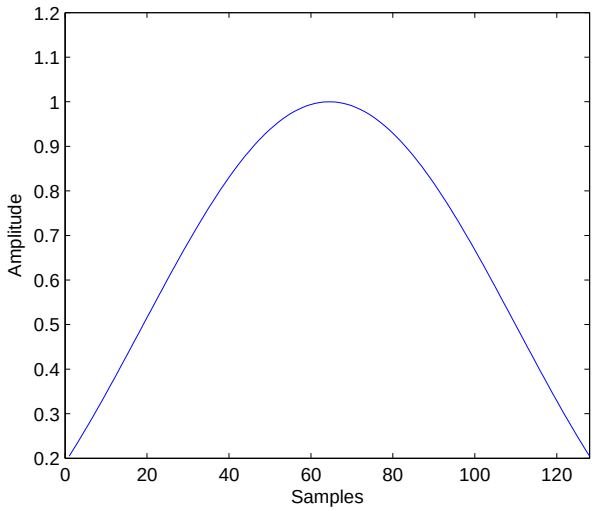
Reconstruction

Subsequent to computing the confidence measure, reconstruction of a higher resolution MRI video is done in the frequency domain. Even and odd undersampled projection lines from corresponding frames of the registered LR videos are combined to form a higher resolution radially sampled dataset in frequency space. The Event Models based registration algorithm described in Section 6.2 is used to determine which radial projections from multiple LR MR images correspond to the same instance of the event. An illustration of the radial projection alignment is shown in Fig. 6.14.

These projections can be combined to increase the sampling resolution in k-space. The inverse Fourier transform of these radial projection lines cannot be directly computed using standard inverse Fourier transform implementations. Therefore, these LR radial samples are regridded to a cartesian representation by weighting the data samples (based on distances from cartesian coordinates) and convolving



(a)



(b)

Figure 6.14: (a) Illustration of two radial projection lines. A data point in between the projection lines is re-gridded by convolving with a symmetric Kaiser kernel. (b) A 1D Kaiser window, $\beta = 3$.

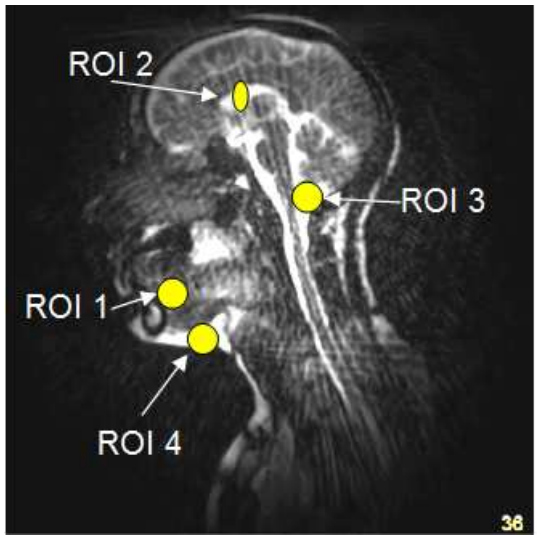


Figure 6.15: ROIs used to compute SNR values in Table 6.6.

with a finite kernel. A symmetric Kaiser window ($\beta = 3$) is usually used to interpolate frequency information in between the radial projections [29]. A 2D Kaiser window of size $M \times M$ can be computed as follows:

$$w(i,j) = \begin{bmatrix} \left[\frac{I_0(\beta\sqrt{1-(\frac{2i}{M}-1)^2})}{I_0(\beta)}, \frac{I_0(\beta\sqrt{1-(\frac{2j}{M}-1)^2})}{I_0(\beta)} \right], & 0 \leq i,j \leq M \\ 0, & otherwise \end{bmatrix} \tag{6.12}$$

A 1D Kaiser window is shown in Fig. 6.14. Computing the inverse Fourier transform of the regridded and interpolated data results in the super-resolved MR images.

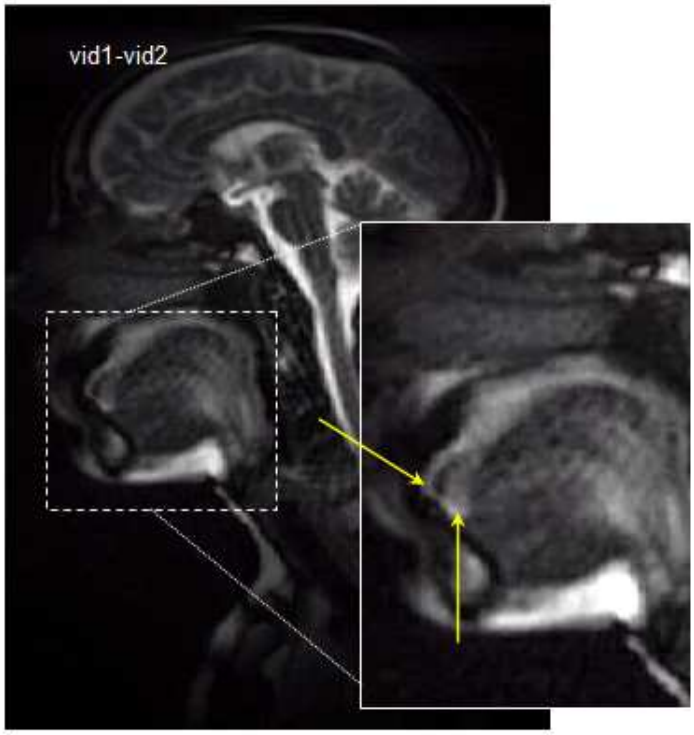
Performance Results

Super-resolution reconstruction of LR MR video sequences results in improvement in both the spatial and temporal resolution of data. However, some combinations of LR input videos result in better reconstruction than others. This can be validated using the confidence measure as well quantitatively measured by computing signal to noise ratios (SNR). The SNR is computed as follows. For each video sequence, two consecutive frames (with no bolus motion) are used to compute a difference image. A region of interest (ROI), corresponding to homogeneous tissues, is manually chosen in one of the frames and the mean pixel intensity μ is computed. For the

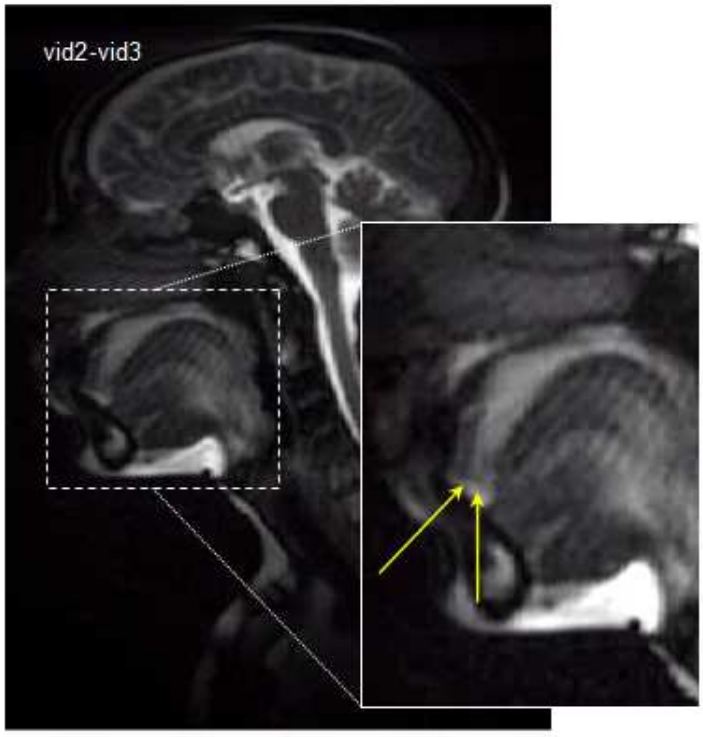
same ROI in the difference image, the standard deviation σ of the pixel intensities is computed. The SNR of that video sequence is then measured as:

$$SNR = \frac{\sqrt{2} \cdot \mu}{\sigma} \tag{6.13}$$

This method of computing SNR in MR images is commonly used when image homogeneity is poor [11]. The improvement in the spatial resolution after SR reconstruction can be seen in Table 6.6, where the SNR values computed for four different ROIs in the LR and SR sequences are presented. These ROIs are highlighted in Fig. 6.15. From these SNR values it can be seen that while SR reconstruction improves the signal to noise ratio of all the video combinations, vid2-vid3 combination has the highest SNR for all four ROIs, which agrees with the computed confidence measure. Visually the improvement in the spatial resolution of the data can be seen in Fig. 6.13(c), where a reconstructed SR frame is presented. The LR video frames that contributed to the SR frame are shown as Fig. 6.13(a) and (b). It can be seen that the SR frame has much less noise compared to either of the LR frames. The improvement in the temporal resolution of the data is demonstrated by the reduction in the motion blurring in the SR video sequence which can be viewed online at [56]. Figure 6.16 shows two SR frames from sequence combinations: vid1-vid2 and vid2-vid3. A zoomed section of the tongue is also shown in order to highlight the two visibly distinct positions of the tip of the tongue in vid1-vid2, which is caused by poor temporal registration. The zoomed section in Fig. 6.16(b) shows that this spatial distinction is less visible for the sequence combination vid2-vid3, which also has a higher confidence measure. Another illustrative result is shown in Fig. 6.17 (zoomed section shown in Fig. 6.18), where after the oesophageal stage, the first frame in which the epiglottis becomes visible are shown. The position of the epiglottis has been highlighted in each frame with an arrow. It can be seen that for vid2-vid3 combination the spatial detail of the epiglottis is the clearest, while for other video combinations two distinct positions of the epiglottis are visible. Thus,



(a)



(b)

Figure 6.16: Representative frames of SR MRI videos. (a) vid1-vid2, $\chi = 27.64$, zoomed position of the tongue shows incorrect registration, (b) vid2-vid3, $\chi = 38.7$, zoomed position of tongue shows correct registration.

fusing vid2-vid3 results in the better registration and reconstruction as compared to vid1-vid2 or vid1-vid3. From the confidence measures listed in Table 6.6 it can be seen that the confidence measure for vid2-vid3 combination is indeed the highest, thus corroborating the subjective evaluation of the reconstruction.

6.5 Summary

In this chapter we presented a confidence measure based strategy that allows us to choose recurrent non-uniform samples sets such that the overall signal reconstruction error is minimized. The confidence measure was developed a linear weighted sum of two objective functions that are based on two precepts: (i) sample sets that are placed farther apart from each other will result in better reconstruction and proposed objective function Φ_g is a suitable estimate of this relation, and, (ii) proposed objective function Φ_l can be used to determine the reliability of the computed temporal registration. We independently evaluated the objective functions to highlight their complementary nature. We also presented a method to determine the optimal weights for the linear weighted sum of the objective functions. An iterative ranking system was also proposed, that updates the rank assigned to sample sets and fuses two sample sets to optimize reconstruction. Such a ranking system based on the confidence measure is shown to out-perform a random ordering of the sample set, which would otherwise have been used when no prior information about the sample-set order is known. We demonstrated the applications of this work in three areas, namely, super-resolution MR imaging, enhanced video reconstruction and enhanced audio generation.

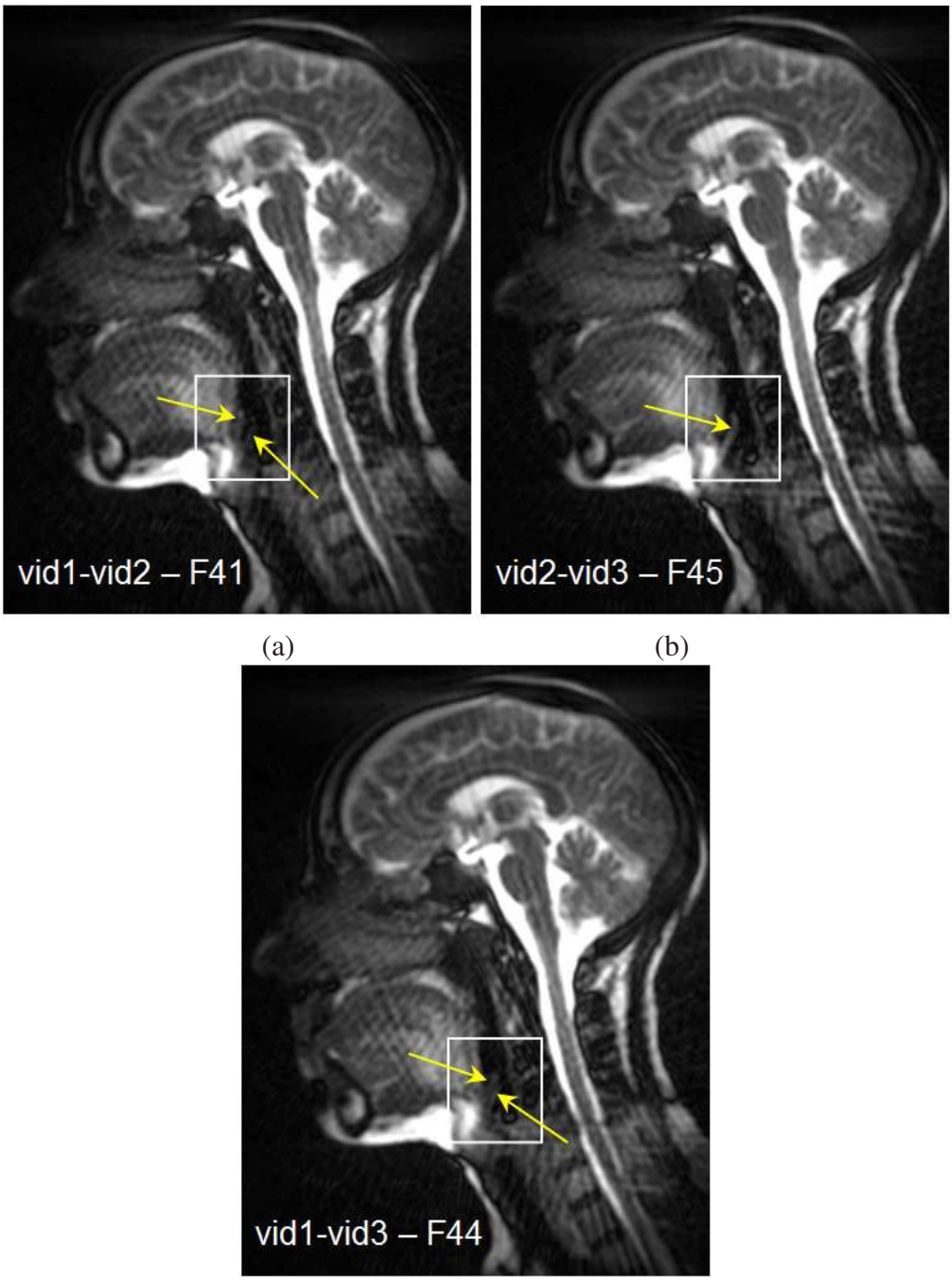


Figure 6.17: Representative frames of SR MRI videos. (a) vid1-vid2, (b) vid2-vid3 and (c) vid1-vid3. The position of the epiglottis has been highlighted with arrows.

(a)

(b)

(c)

Figure 6.18: Zoomed in sections of SR MRI frames shown in Fig. 6.17. (a) vid1-vid2, (b) vid2-vid3 and (c) vid1-vid3. The position of the epiglottis has been highlighted with arrows.

Chapter 7

Summary and Conclusions

The major goal of this thesis work was to improve the efficiency of spatio-temporal super-resolution imaging. We proposed a novel strategy of using multiple related events for video super-resolution. This approach can be used to generate dynamic Super-resolution MRI sequences which can improve diagnosis of functional disorders. We also developed a unique metric to choose between low resolution sequences inorder to maximize super-resolution reconstruction accuracy. The major contributions of this research work are presented in Section 7.1. The publications resulting from this work are listed in Section 7.2. Future research directions are identified in Section 7.3.

7.1 Contributions

Super-resolution imaging is fast becoming a key component in many applications such as forensic imaging, high definition video processing and 4D medical imaging. Although spatial super-resolution techniques have been available since mid-eighties, research in spatio-temporal super-resolution is still in its infancy. Due to low frame rate limitations (specifically in MRI), varied view points and variability in temporal scale of activities and events captured on video sequences, spatio-temporal SR is a challenging domain. Our research work has focused on improving the efficiency and accuracy of spatio-temporal SR imaging. We have contributed

to improving the state-of-the-art, proposed novel methods to incorporate information from multiple related video sequences and proposed a novel metric to measure the contribution of a sequence towards optimal SR reconstruction. The following paragraphs summarize the major contributions of this work.

1. In this thesis we first examined the simple case that the video sequences do not have a dynamic temporal scale and have been acquired within a single occurrence. Even in such simple cases, if the acquisition rate of the cameras is lower than the Nyquist rate (*i.e.*, the speed of the event is high), then the current state-of-the-art methods used to solve for the alignment between the sequences do not perform well. Our first contribution lies in proposing an elegant paradigm shift in the alignment strategy so that instead of using a local matching criterion, a more global approach to alignment is used. We built a parametric model of the event that incorporates the motion trend in the video sequence and computes a continuous motion trajectory; as compared to previous approaches of linear interpolation between discrete samples. Alignment of the video sequences was computed as the minimization of the Euclidean distance between the event models. Based on recently published works [21][20][22], we showed that for low temporal resolution videos our parametric **event-model** based alignment algorithm performs a more accurate temporal alignment than contemporary approaches.

2. In the second contribution of this work we considered the scenario where multiple LR videos of the same scene are not available; instead multiple LR videos of an ensemble of the same event are available. Within an ensemble of sequences the temporal duration of the event can vary dynamically and non-linearly. Computing a sub-frame temporal transformation between the LR videos of the ensemble will reduce the effect of temporal aliasing and provide a framework for increasing the temporal resolution of video sequences.

Since we proposed to use an ensemble of the dynamic scene to generate a high resolution video, the temporal misalignment within the ensemble could no longer be modeled as a 1D affine transform. We developed a novel **symmetric transfer error** (**STE**) that was iteratively minimized to align video sequences. In this work we addressed a multitude of variations in the input sequences at the same time – view invariance, sub-frame accuracy and dynamic temporal alignment. Contemporary methods only address a small subset of these variations. This piece of work can be used to compare, index and retrieve video sequences of activities that vary on temporal scale but are essentially the same event. For example, two different individuals (say a teacher and a learner) performing the same movements of Tai-Chi or Ballet, where the learner can examine the alignment computed between his/her movements and the teachers movements. An application of the STE to MRI SR reconstruction, where the multiple LR sequences vary in temporal scale, was presented.

3. Contributions 1–2 dealt with aligning and reconstructing SR data from LR sequences that are acquired in the same imaging plane or have sufficient overlap in the viewpoint for feature matching to be computed. The next contribution of this thesis dealt with **alignment of LR MR sequences that are acquired in bi-directional orthogonal planes**. In orthogonal acquisitions, the cross section of the anatomy being viewed is widely different, hence feature matching cannot be used. In our third contribution, we developed a method that used the dynamics of events in the individual MR sequences to register the orthogonal video sequences. This resulted in the generation of 4D MR data for enhanced medical visualization.

4. Lastly, our investigation of the factors that affect SR reconstruction resulted in

the determination that SR accuracy is based on two criteria– non-uniformity of the sample set (or video sequence frames) and the accuracy of the registration or alignment algorithm. The last contribution of this thesis is a **confidence measure** that takes into account the two criteria mentioned above and provides a quantitative number based on which we can choose LR sequences that will improve the reconstruction performance and discard those sequences that will deteriorate the reconstruction. It is important to note that the confidence measure can be added to any system irrespective of the method of alignment and the method of reconstruction. This screening of LR sequences is a unique concept in SR imaging, as before this thesis work, the determination of which LR sequence to use would be made post-reconstruction by visual comparison between all combinations or all LR sequences would be used. We also developed an **iterative greedy algorithm** that uses the confidence measure to rank multiple LR sequences and efficiently reconstruct SR sequences.

5. Through the development of this thesis, a database of video sequences used for synchronization was also compiled and has been made publicly available to the research community.

7.2 Publications

Parts of this work have been published in (or submitted) to the following journals and conferences:

Published

1. Meghna Singh, Anup Basu and Mrinal Mandal, "Event Dynamics based Temporal Registration," *IEEE Transactions on Multimedia*, Vol.9, No.5, pp. 1004-1015, Aug. 2007.

2. Meghna Singh, Richard Thompson, Anup Basu, Jana Rieger and Mrinal Mandal, "Image Based Temporal Registration of MRI data for Medical Visualization," *Proc. of IEEE International Conference on Image Processing ICIP*, pp. 1169-1172, Atlanta, Georgia, Oct 2006.

3. Meghna Singh, Anup Basu, Mrinal Mandal, "Temporal Alignment of Time Varying MRI Datasets for High Resolution Medical Visualization," *Proc. of* 2^{nd} *International Symposium on Visual Computing*, Lake Tahoe, Nevada, pp. 222-231, Nov 6-8, 2006.

4. Meghna Singh, Mrinal Mandal and Anup Basu, "Confidence Measure for Temporal Registration of Recurrent Non-uniform Samples," *Proc. of Intl. Conf. on Pattern Recognition and Machine Intelligence*, pp. 608-615, Kolkata, India, Dec 2007.

5. Meghna Singh, Mrinal Mandal and Anup Basu, "A Confidence Measure and Iterative Rank based method for Temporal Registration," *Proc. of the* 33^{rd} *International Conference on Acoustics, Speech and Signal Processing ICASSP*, pp. 1289-1292, Las Vegas, March 30-April 4, 2008.

6. Meghna Singh, Lin Irene Cheng, Mrinal Mandal and Anup Basu, "Optimization of Symmetric Transfer Error for Sub-frame Video Synchronization," *Proc. of European Conference on Computer Vision* ECCV 2008, pp. 554-567, Marseille, France, Oct 12-18, 2008.

7. Meghna Singh, Lin Irene Cheng, Mrinal Mandal, "4D Alignment of Bidirectional Dynamic MRI sequences," *IEEE Engineering in Medicine and Biology Conference 2008*, pp. 5893 - 5896 , Vancouver, Canada, 20-25 Aug, 2008.

In Preparation

1. Meghna Singh, Mrinal Mandal and Anup Basu, "Choice of Low Resolution Sample Sets for Super-Resolution Signal Reconstruction," in preparation for submission to *IEEE Trans. on Multimedia*, 30 pages.

7.3 Future Research Direction

Research work conducted so far has pushed the boundaries of spatio-temporal SR imaging, video synchronization and 4D SR medical imaging. However, more research work should be done to generalize the techniques proposed in this work, which will facilitate the universal adoption of these techniques. We identify some future research areas as follows:

- Scale-Space Representation of Event Models.

  In this thesis, we have implemented a single event model for each sequence. However, in a more generalized scenario a single sequence can have multiple motion trajectories that need to be modeled. The concept of event models can be extended to a piece-wise model for multiple complex motion. For long sequences a scale-space approach can be used to develop different scale-levels of the event models.

- Generalized Symmetric Transfer Error

  The symmetric transfer error developed as part of this thesis is based on the assumption that the scene geometry is related by a homography. This assumption can be removed by generalizing the spatial relationship to the fundamental matrix (which includes homography as a special case). The algorithm can also be extended to incorporate information from multiple trajectory alignments. A consensus seeking algorithm (based on the same principles as the random sample and consensus RANSAC algorithm) can be used to incorporate this information.

- Enhancement of the Confidence Measure

  The confidence measure developed as part of this work is formulated as a weighted linear combination of the local and global registration errors between single-trajectory event models in two (or more) sequences. It would be interesting to investigate the effects of multiple motion trajectories (per sequence) on the confidence measure and suitable modifications will be needed to the algorithm in order to compute a confidence measure in this scenario.

Bibliography

[1] A. Anagnostara, S. Stoeckli, O.M. Weber, and S.S. Kollias. Evaluation of the anatomical and functional properties of deglutition with various kinetic high-speed MRI sequences. *Journal of Magnetic Resonance Imaging*, 14:194–199, 2001.

[2] E. Bruno and D. Pellerin. Video structuring, indexing and retrieval based on global motion wavelet coefficients. *In Proceedings of International Conference of Pattern Recognition (ICPR)*, August 2002.

[3] D.E. Butler, V.M. Bove, Jr., and S. Sridharan. Real-time adaptive foreground/background segmentation. *EURASIP Journal of Applied Signal Processing*, (14):2292–2304, 2005.

[4] R. L. Carceroni, F. L. C. Padua, G. A. M. R. Santos, and K. N. Kutulakos. Linear sequence-to-sequence alignment. *In Proc. Computer Vision and Pattern Recognition CVPR*, pages I: 746–753, 2004.

[5] Y. Caspi and M. Irani. Spatio-temporal alignment of sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1409–1424, Nov 2002.

[6] Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. *Intl. Journal of Computer Vision*, 68(1):53–64, 2006.

[7] V. Cheung, B.J. Frey, and N. Jojic. Video epitomes. *In Proc. of Conference on Computer Vision and Pattern Recognition CVPR*, 2005.

[8] C. Dai, Y. Zheng, and X. Li. Subframe video synchronization via 3d phase correlation. In *Proc. Intl. Conference on Image Processing ICIP*, pages 501–504, 2006.

[9] J. Davis, A. Bobick, and W. Richards. Categorical representation and recognition of oscillatory motion patterns. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 628–635, June 2000.

[10] F. Dekeyser, P. Bouthemy, P. Perez, and E. Payot. Super-resolution from noisy image sequences exploiting a 2d parametric motion model. *In Proc. 15th International Conference on Pattern Recognition*, 3:350–353 vol.3, 2000.

[11] O. Dietrich, J.G. Raya, S.B. Reeder, M.F. Reiser, and S.O. Schoenberg. Measurement of signal-to-noise ratios in mr images: Influence of multi-channel coils, parallel imaging and reconstruction filters. *Magnetic Resonance Imaging*, 26(2):375–385, 2007.

[12] A. Divakaran, A. Vetro, K. Asai, and H. Nishikawa. Video browsing system based on compressed domain feature extraction. *IEEE Trans. on Consumer Electronics*, 46(3):637–644, Aug 2000.

[13] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. *In Proc. IEEE International Conference on Computer Vision ICCV*, 2003.

[14] O. Engwall. A 3d tongue model based on mri data. *In Proc. of 6th Intl. Conf. on Spoken Language Processing (ICSLP)*, pages 901–904, 2000.

[15] R. Werner *et al.* Motion artifact reducing reconstruction of 4d ct image data for the analysis of respiratory dynamics. *Methods Inf. Med.*, 46, 2007.

[16] S. Achenbach *et al.* Noninvasive coronary angiography by retrospectively ecg-gated multislice spiral CT. *Journal of Circulation*, pages 2823–2828, Dec 2000.

[17] H. G. Feichtinger and T. Werther. Improved locality for irregular sampling algorithms. In *Proc. of the International Conference Acoustics, Speech, and*

Signal Processing ICASSP, pages 3834–3837, Washington, DC, USA, 2000. IEEE Computer Society.

[18] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[19] H. Freeman. In *Discrete-Time Systems*. John Wiley and Sons Inc., 1965.

[20] S. Genc and F.T. Yarman-Vural. Morphing as a tool for motion modeling. *In Proc. of International Conference on Image Analysis and Processing*, pages 538–543, 1999.

[21] M. A. Giese and T. Poggio. Morphable models for the analysis and synthesis of complex motion patterns. *Int. J. of Computer Vision*, 38(1):59–73, 2000.

[22] R. Gonzalez, R. Woods, and S. Eddins. *Digital Image Processing using Matlab*.

[23] H. Greenspan, S. Peled, G. Oz, and N. Kiryati. Mri inter-slice reconstruction using super-resolution. In *Proc. of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention MICCAI*, pages 1204–1206, London, UK, 2001. Springer-Verlag.

[24] C. Harris and M.J. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, pages 147–152, 1988.

[25] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[26] D. Hazen, R. Puri, and K. Ramachandran. Multicamera video resolution enhancement by fusion of spatial disparity and temporal motion fields. *Proc of IEEE International Conference on Computing Vision Systems ICVS*, 2006.

[27] R. Hess and A. Fern. Improved video registration using non-distinctive local image features. *In Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR*, pages 1–8, June 2007.

[28] M. Irani and S. Peleg. Image sequence enhancement using multiple motions analysis. *In Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR*, pages 216–221, Jun 1992.

[29] J.I. Jackson, C.H. Meyer, D.G. Nishimura, and A. Macovski. Selection of a convolution function for fourier inversion using gridding [computerised tomography application]. *IEEE Transactions on Medical Imaging*, 10(3):473–478, Sep 1991.

[30] S. Jeannin and A. Divakaran. Mpeg-7 visual motion descriptors. *IEEE Trans. on Circuits and Systems for Video Technology*, 11(6):720–724, June 2001.

[31] N. Jojic, B.J. Frey, and A. Kannan. Epitomic analysis of appearance and shape. *In Proc. Intl. Conference on Computer Vision*, 2003.

[32] H.A. Karim, M. Bister, and M.U. Siddiqi. Low rate video frame interpolation-challenges and solution. *In Proc. of Intl. Conference on Acoustics, Speech and Signal Processing ICASSP*, 3:III–117–20, April 2003.

[33] J.A. Kennedy, O. Israel, A. Frenkel, R. Bar-Shalom, and Haim Azhari. Super-resolution in pet imaging. *IEEE Transactions on Medical Imaging*, 25(2):137–147, Feb. 2006.

[34] E. Keogh and M. Pazzani. Derivative dynamic time warping. In *SIAM International Conference on Data Mining*, 2001.

[35] D. Lauzon and E. Dubois. Representation and estimation of motion using a dictionary of models. *In Proc. of Intl. Conference on Acoustics, Speech and Signal Processing ICASSP*, 5:2585–2588, 1998.

[36] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: establishing a common coordinate frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):758–767, Aug 2000.

[37] Cheng Lei and Yee-Hong Yang. Tri-focal tensor-based multiple video synchronization with subframe optimization. *IEEE Transactions on Image Processing*, 15(9):2473–2480, Sept. 2006.

[38] J.L. Len. On nonuniform sampling of bandwidth-limited signals. *IRE Trans. on Circuit Theory*, CT-3:251–257, Dec 1956.

[39] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318, December 1993.

[40] J. Listgarten, R.M. Neal, S.T. Roweis, and A. Emili. Multiple alignment of continuous time series. *Advances in Neural Information Processing Systems*, 2005.

[41] J.J. Little and J.E. Boyd. Describing motion for recognition. *In Intl. Symposium on Computer Vision*, pages 235–240, 1995.

[42] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 60(2):91–110, 2004.

[43] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Intl. Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[44] Y.F. Ma and H. Zhang. Motion texture: A new motion based video representation. *In Proc. Intl. Conference on Pattern Recognition*, 2:548–551, 2002.

[45] F. Marvasti. *Nonuniform Sampling Theory and Practice*. Kluwer Academic, 2001.

[46] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *Proceedings of British Machine Vision Conference*, I:384–393, 2002.

[47] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Intl. Journal Computer Vision*, 65(1-2):43–72, 2005.

[48] M.Singh. http://www.ece.ualberta.cam̃eghna/embc08.html.

[49] A.J. Patti, M.I. Sezan, and A. Murat Tekalp. Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *IEEE Transactions on Image Processing*, 6(8):1064–1076, Aug 1997.

[50] D. Perperidis, R.H. Mohiaddin, and D. Rueckert. Spatio-temporal free-form registration of cardiac mr image sequences. *Medical Image Analysis*, 9(5):441–456, October 2005.

[51] R. Piroddi and T. Vlachos. A simple framework for spatio-temporal video segmentation and delayering using dense motion fields. *IEEE Signal Processing Letters*, 13(7):421–424, July 2006.

[52] D. W. Pooley, M. J. Brooks, A. J. van den Hengel, and W. Chojnacki. A voting scheme for estimating the synchrony of moving-camera videos. *In Proc. of Intl. Conference on Image Processing*, 1:I–413–16 vol.1, 14-17 Sept. 2003.

[53] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 2nd edition, 1992.

[54] C. Rao, A. Gritai, M. Shah, and T. F. Syeda Mahmood. View-invariant alignment and matching of video sequences. In *Proc. of the Intl. Conference on Computer Vision*, pages 939–945, 2003.

[55] E. Shechtman, Y. Caspi, and M. Irani. Space-time super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):531–545, April 2005.

[56] M. Singh. www.ece.ualberta.ca/ meghna/j2008.html.

[57] M. Singh, A. Basu, and M. Mandal. Event dynamics based temporal registration. *IEEE Transactions on Multimedia*, 9(5):1004–1015, Aug. 2007.

[58] M. Singh, A. Basu, and M. K. Mandal. Temporal alignment of time varying mri datasets for high resolution medical visualization. In *In Proc. of Intl. Symposium on Visual Computing (1)*, pages 222–231, 2006.

[59] M. Singh, I. Cheng, and M. Mandal. 4d alignment of bidirectional dynamic mri sequences. *30th Intl. Conference of IEEE Engineering in Medicine and Biology Society EMBC 2008*, pages 5893 – 5896, 2008.

[60] M. Singh, I. Cheng, M. Mandal, and A. Basu. Optimization of symmetric transfer error for sub-frame video synchronization. *In Proc. of European Conf. on Computer Vision*, pages 554–567, 2008.

[61] M. Singh, M. Mandal, and A. Basu. A confidence measure and iterative rank-based method for temporal registration. *Proc. of Intl. Conference on Acoustics, Speech and Signal Processing ICASSP*, pages 1289–1292, April 2008.

[62] M. Singh, M. K. Mandal, and A. Basu. Confidence measure for temporal registration of recurrent non-uniform samples. In *Proc. of Intl. Conference on Pattern Recognition and Machine Intelligence*, pages 608–615, 2007.

[63] M. Singh, W. Sungkarat, J. Jeong, and Y. Zhou. Extraction of temporal information in functional mri. *IEEE Trans. on Nuclear Science*, 49(2), Oct 2002.

[64] M. Singh, R. Thompson, A. Basu, J. Rieger, and M. Mandal. Image based temporal registration of mri data for medical visualization. *In Proc. of IEEE International Conference on Image Processing*, pages 1169–1172, Oct. 2006.

[65] H. Stark and P. Oskoui. High-resolution image recovery from image-plane arrays using convex projections. *Journal of Optical Society*, pages 1715–1726, 1989.

[66] T. Strohmer and J. Tanner. Fast reconstruction algorithms for periodic nonuniform sampling with applications to time-interleaved adcs. *In Proc. of Intl. Conference on Acoustics, Speech and Signal Processing ICASSP*, 3:III–881–III–884, April 2007.

[67] T. Strohmer and Jiadong Xu. Fast algorithms for blind calibration in time-interleaved analog-to-digital converters. *In Proc. of Intl. Conference on Acoustics, Speech and Signal Processing ICASSP*, 3:III–1225–III–1228, April 2007.

[68] C. Su, H.M. Liao, and K. Fan. A motion-flow-based fast video retrieval system. *In Proceedings of the 7th ACM SIGMM international Workshop on Multimedia information Retrieval*, November 2005.

[69] A.M. Tekalp, M.K. Ozkan, and M.I. Sezan. High resolution image reconstruction from lower resolution image sequences and space-varying image reconstruction. *In Proc. of Intl. Conference on Acoustics, Speech and Signal Processing ICASSP*, pages 169–172, 1992.

[70] R. B. Thompson and E. R. McVeigh. High temporal resolution phase contrast MRI with multiple echo acquisitions. *Magnetic Resonance in Medicine*, 47:499–512, 2002.

[71] B.C. Tom and A.K. Katsaggelos. Resolution enhancement of video sequences using motion compensation. *In Proc. International Conference on Image Processing*, 1:713–716, Sep 1996.

[72] A.M. Tourapis, C. Hey-Yeon, M.L. Liou, and O.C. Au. Temporal interpolation of video sequences using zonal based algorithms. *Proc. of Intl. Conference on Image Processing*, 3, Oct 2001.

[73] P. Tresadern and I. Reid. Synchronizing image sequences of non-rigid objects. *In Proc. of the British Machine Vision Conference*, 2:629–638.

[74] R.Y. Tsai and T.S. Huang. Multiframe image restoration and registration. *Advances in Computer Vision and Image Processing*, pages 317–339, 1984.

[75] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *Int. Journal of Computer Vision*, 59(1):61–85, 2004.

[76] T. Tuytelaars and L. J. VanGool. Synchronizing video sequences. In *Proc. of Intl. Conference on Computer Vision and Pattern Recognition*, pages I: 762–768, 2004.

[77] Patrick Vandewalle, Sabine Sasstrunk, and Martin Vetterli. A Frequency Domain Approach to Registration of Aliased Images with Application to Super-Resolution. *EURASIP Journal on Applied Signal Processing (special issue on Super-resolution)*, 2006.

[78] N. Vasconcelos and A. Lippman. A spatiotemporal motion model for video summarization. *In Proc. of Intl. Conference on Computer Vision and Pattern Recognition*, 1998.

[79] M. vonSiebenthal *et al.* 4d mr imaging of repiratory organ motion and its variability. *Phys. Med. Biol.*, 52:1547–1564, 2007.

[80] H. Wechsler, Z. Duric, F. Li, and V. Cherkassky. Motion estimation using statistical learning theory. *IEEE Trans. PAMI*, 26(4):466–478, April 2004.

[81] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[82] Wikipedia. History of the camera — wikipedia, the free encyclopedia, 2008. [Online; accessed 21-July-2008].

[83] B. Wittenmark and N. Trngren. Timing problems in real-time control systems. *Proc. American Control Conference*, 1995.

[84] L. Wolf and A. Zomet. Wide baseline matching between unsynchronized video sequences. *Intl. Journal of Computer Vision*, 68(1):43–52, June 2006.

[85] C.R. Wren, A. Azarbayejani, T. Darrel, and A. Pentland. Pfinder: Real time tracking of the human body. *IEEE Trans. on PAMI*, pages 780–785, 1997.

[86] C. Yang and M. Stone. Dynamic programming method for temporal registration of three-dimensional tongue surface motion from multiple utterances. *Speech Communications*, 38(1):201–209, 2002.

[87] X. Yuan and G. Chi-Fishman. Volumetric tongue reconstruction by fusing bidirectional mr images. *Proc. of 3rd IEEE Intl Symp. on Biomedical Imaging*, 2006.

[88] L. Zelnik-Manor and M. Irani. Event-based analysis of video. *In Proc. of Intl. Conference on Computer Vision and Pattern Recognition*, 2:123–130, 2001.

Appendix A

Weighted Least Square Regression

In least square(LSQ) regression the unknown parameters β of a linear model such as A.1 are approximated by estimating β by A.2, such that $\hat{Y} = X\hat{\beta}$ and the sum of squared errors $SSE = \sum_{i=1}^{n} ||y_i - \hat{y}||^2$ is minimized. In the case of probabilistic models the parameters can be estimated by maximizing the the expectation of the likelihood of the parameters, which is the well known Expectation Maximization algorithm. However, in linear regression we do not deal with probabilistic models.

$$Y = X\hat{\beta} + \epsilon \tag{A.1}$$

$$\hat{\beta} = (X^T X)^{-1} X^T Y \tag{A.2}$$

In weighted LSQ, the following weighted SSE is minimized:

$$SSE = \sum_{i=1}^{n} w_i ||y_i - \hat{y}||^2 \tag{A.3}$$

Let Q be the square root of the weight matrix W (w_i are diagonal elements of W), then $W = Q^T Q$. If we multiply both sides of A.1 by Q we get:

$$\begin{aligned} (Y = X\beta + \epsilon) \times Q \\ QY = QX\beta + Q\epsilon \end{aligned} \tag{A.4}$$

The best linear estimate of β in terms of QX, in the least square sense is:

$$\begin{aligned} \hat{\beta} &= ((QX)^T (QX))^{-1} (QX)^T QY \\ \hat{\beta} &= (X^T Q^T QX)^{-1} X^T Q^T QY \\ \hat{\beta} &= (X^T (W) X)^{-1} X^T (W) Y \end{aligned} \tag{A.5}$$

Since the weights of the LSQ method are unknown, they are computed iteratively in the following manner:

1. Initialize weight matrix to I (identity matrix).

2. Perform least squares regression (A.1) to get an estimate of β.

3. Compute the residual $\hat{R} = Y - \hat{R}$ where $\hat{Y} = X\hat{\beta}$.

4. Compute the adjustment factor $r_{adj} = \frac{1}{\sqrt{1-H_i}}$ where $H = X(X^TX)^{-1}X^T$. The matrix H down-weighs high leverage data points that have a large effect on the least squares.

5. Standardize the residual $r_{std} = (\hat{r})/(MAD) \times r_{adj}$, where MAD is the mean absolute deviation of the residuals.

6. Compute the Bi-square weights as:

$$w_i = \begin{cases} (1 - (r_{std})^2)^2, & if\ |r_{std}| < 1 \\ 0, & if\ |r_{std}| \geq 1 \end{cases}$$

7. Compute estimate of β again and iterate step 2-6 with new weights until convergence.

Appendix B

Video DataBases

We have developed and used a wide range of synthetic 1D, audio and video test sequences in this PhD thesis. In this Appendix we describe the generation of the test sequences and present some representative frames (or sample sets) in order to give the reader an understanding of the experimental data.

The databases have been broadly categorized into three groups – Database 1, Database 2 and Database 3. Database 1 comprises of both synthetic and real test sequences and has been used primarily in Chapter 3 and Chapter 6 of this thesis. Activities like swinging and throwing a ball were captured. Database 2 comprises of synthetic and real test sequences as well. However, the method of generation of synthetic sequences for Database 2 is different. The real sequences in Database 2 are from two sources – (i) MCCL Lab at the Univ of Alberta (our sequences) and (ii) Vision Lab at the University of Florida (UCF videos). Database 3 comprises of MRI sequences in three sagittal planes and one coronal plane. These databases are described in detail in the following sections.

B.1 Database 1

Database 1 (DB-1) consists of six real video sequences and eight synthetic test sequences which were used in Chapter 3 and Chapter 6 of this thesis. These test sequences are described in the following sections.

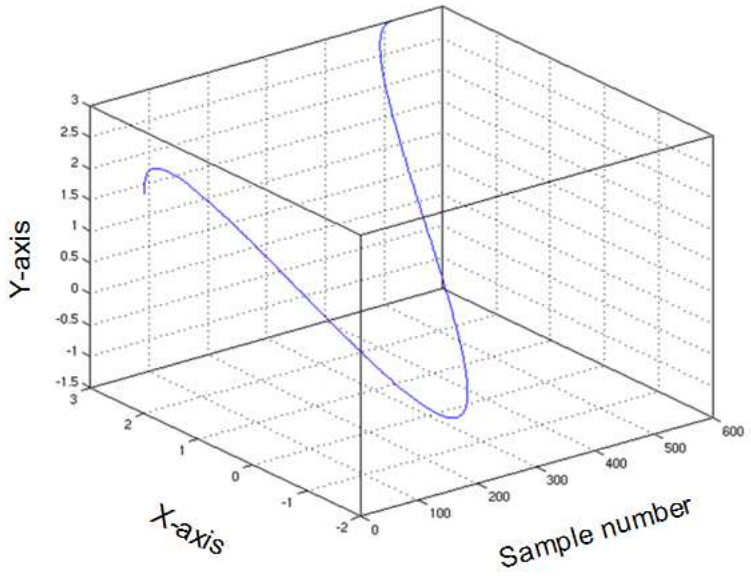


Figure B.1: DB-1 Synthetic 3D trajectory generated by using the MATLAB commands displayed in Section B.1.1

B.1.1 DB-1 Synthetic Sequences

Synthetic data was created by downsampling eight high temporal resolution trajectories of hyperbolic functions, such as $tanh$, $cosh$, $sinh$ and combinations of these functions. An excerpt from the MATLAB code used to generate the trajectories is as follows:

$\gg$ i=1:512 % length of the synthetic trajectory.

$\gg$ sig=tanh(2*pi*i./512)+ 2*cosh(2*pi*i./512);

$\gg$ plot3(i, sig, sig);

The MATLAB code generates a 3D trajectory as shown in Fig.B.1. Similarly, other combinations of signals have been used to generate the synthetic trajectories shown in Fig. B.2. The synthetic trajectories vary in length from 126 to 512 samples. Each high resolution trajectory was downsampled into two trajectories each. One of these two downsampled trajectories was also offset by a known temporal difference in order to create temporal misalignments.

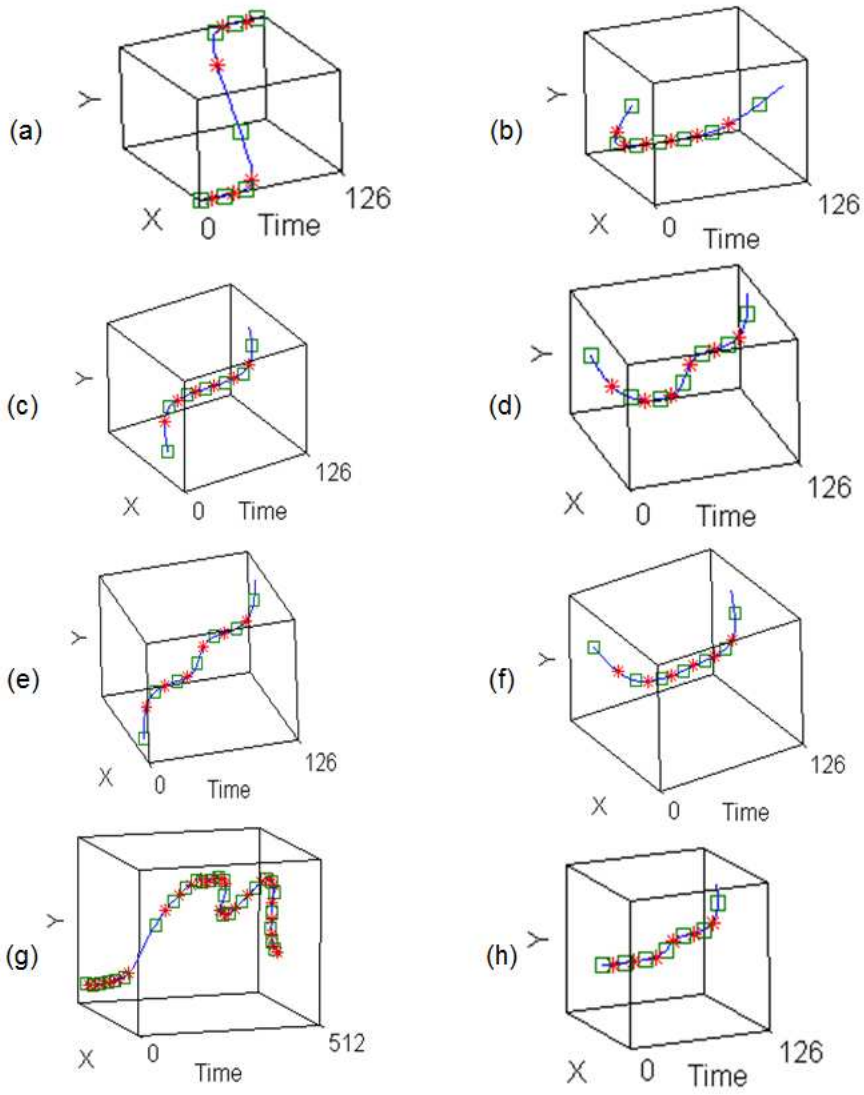


Figure B.2: DB-1 (a)-(h) Sample trajectories from synthetic data. (a) Traj 1, (b) Traj 2, (c) Traj 3, (d) Traj 4, (e) Traj 5, (f) Traj 6, (g) Traj 7, and (h) Traj 8.

Video Name	Number of Frames	Frame Rate
Throw 1	60	30fps
Throw 2	40	30fps
Throw 3	20	30fps
Swing 1	40	30fps
Swing 2	30	30fps
Swing 3	60	30fps

Table B.1: DB-1 List of Real Sequences.

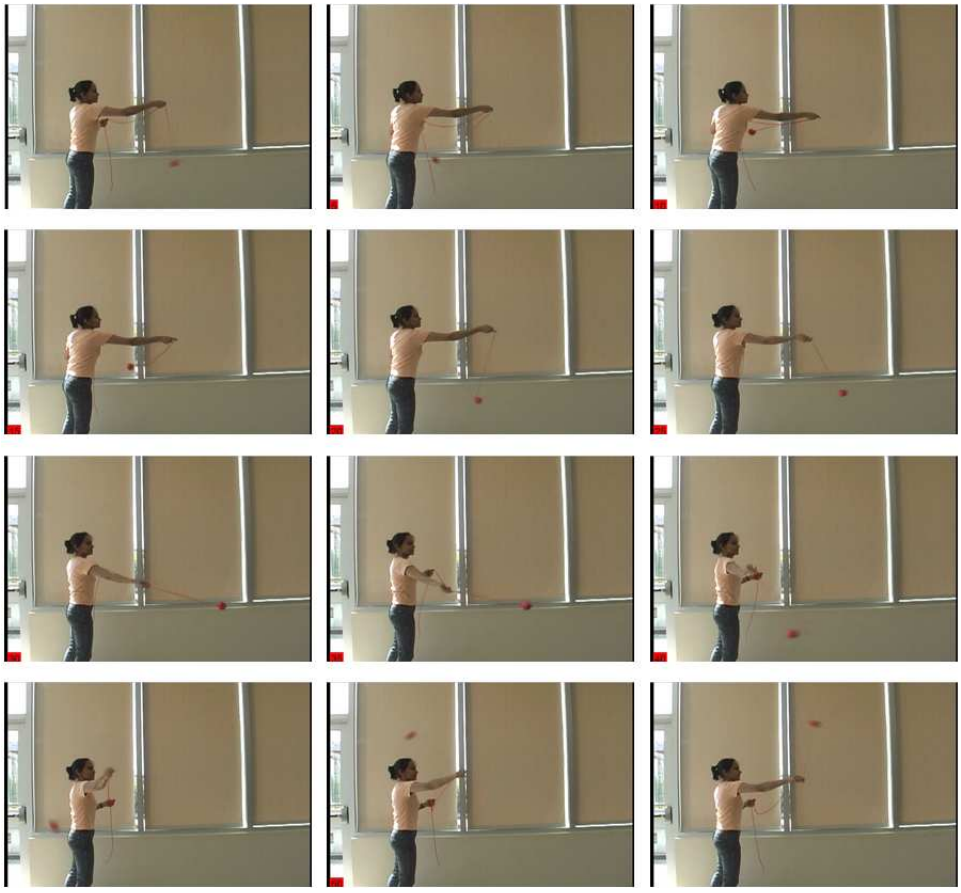


Figure B.3: DB-1 Illustrative Video Frames from Swing 3 video. Sequences should be read row-wise from top-left corner, every fifth video frame is shown.

B.1.2 DB-1 Real Sequences

We captured multiple video sequences of throwing a ball and swinging a ball tied to the end of a string. The frame size and frame rate of the sequences is 640×480 and 30 frames per second respectively. The sequences vary in length from 20 frames to 60 frames. Table B.1 lists the six real video sequences.

Sample frames from the Swing 3 video sequence are shown in Fig. B.3. Sample frames from the Throw 1 video sequence are shown in Fig. B.4, a white box has been used to highlight the position of the ball in the frames.

These sequences were captured at 30fps and were downsampled into two sub-sequences each, with varying frame rates ranging from 15fps to 3.3fps. One of these two downsampled trajectories was also offset by a known temporal difference in order to create temporal misalignments. The moving ball was segmented using

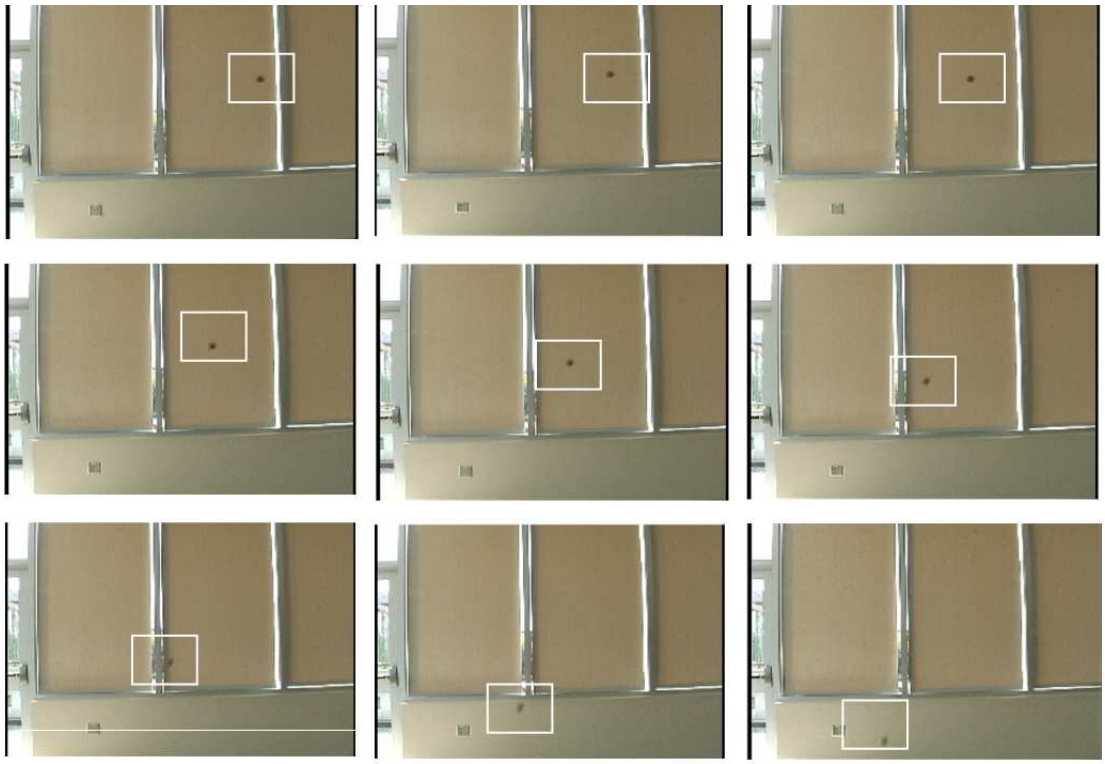


Figure B.4: DB-1 Illustrative Video Frames from Throw 1 video. Sequences should be read row-wise from top-left corner, every alternate video frame is shown.

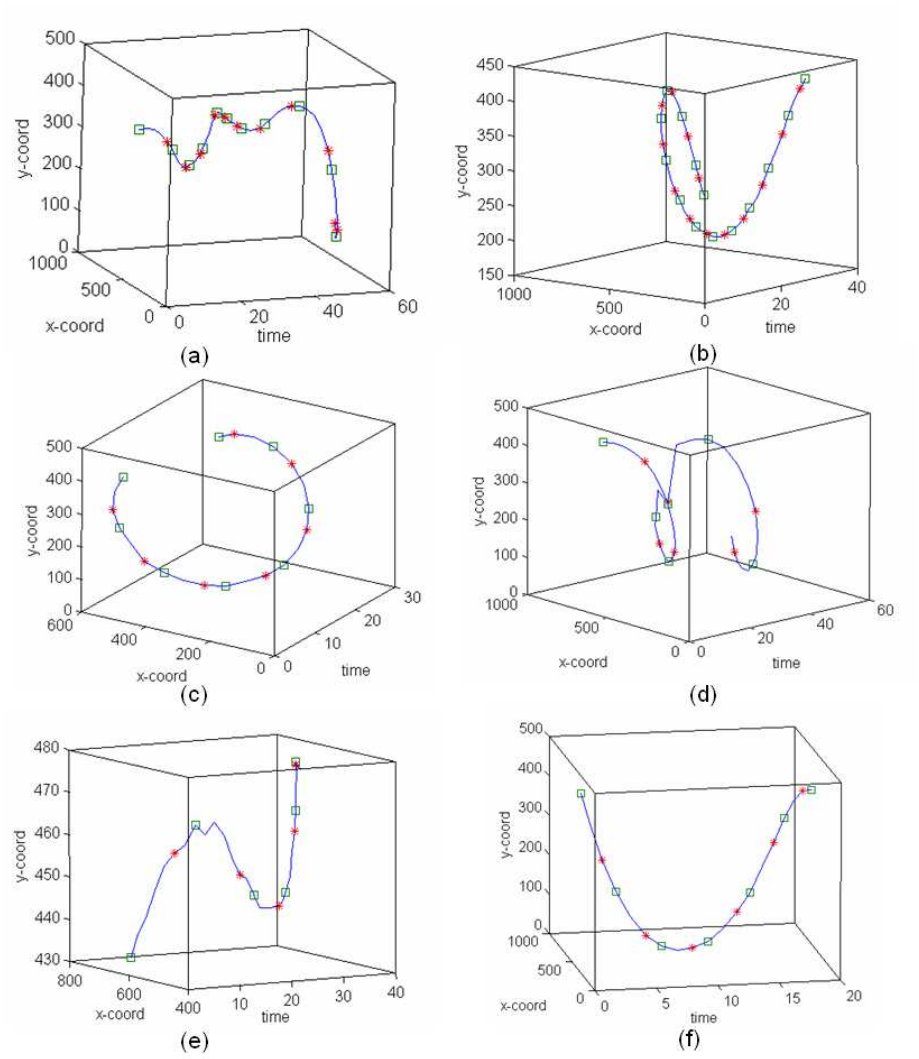


Figure B.5: DB-1 (a)-(f) Sample trajectories from the real data. (a) Throw 1, (b) Swing 1, (c) Swing 2, (d) Swing 3, (e) Throw 2 and (f) Throw 3.

a color based scheme and the centroid of the segmented region was computed. The centroid trajectories for the real sequences are shown in Fig. B.5.

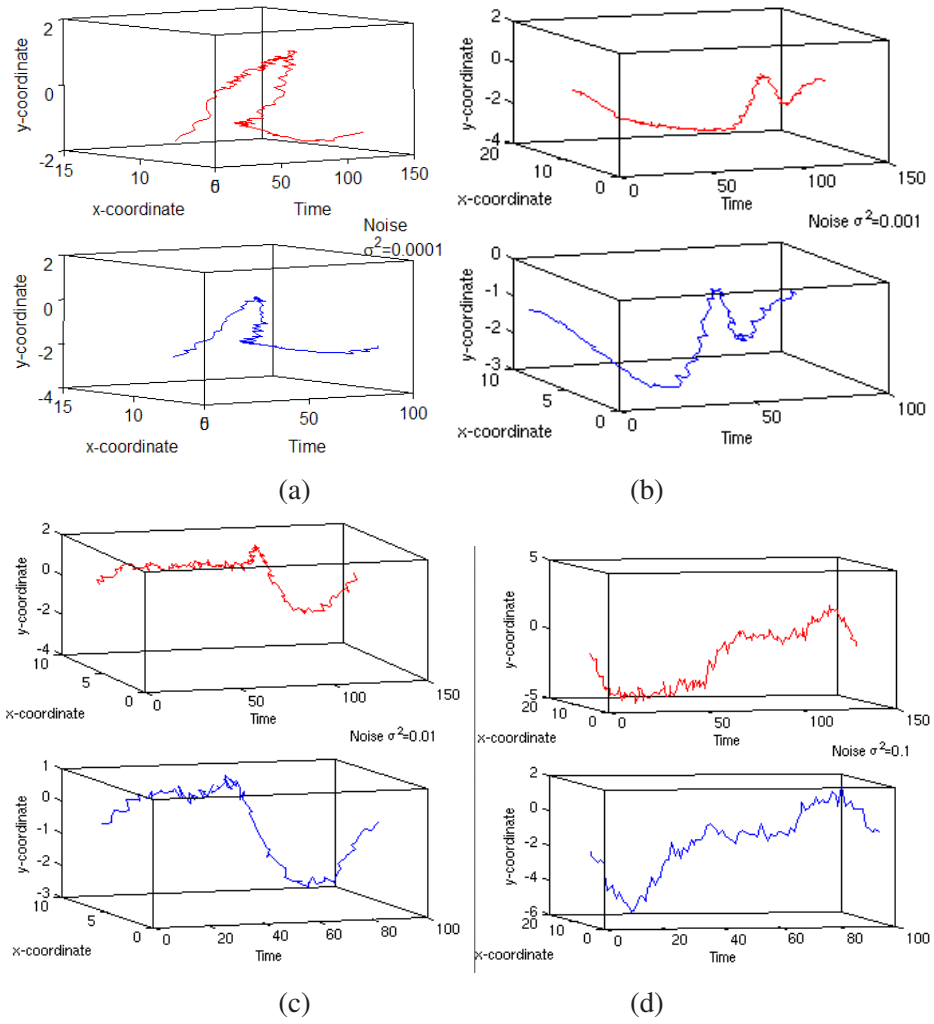


Figure B.6: DB-2 Representative noisy synthetic trajectories with noise variance $\sigma^2 =$ (a) 0.0001, (b) 0.001, (c) 0.01, (d) 0.1.

B.2 Database 2

Database 2 (DB-2) consists of six real video sequences and eight synthetic test sequences which were used in Chapter 4 of this thesis.

B.2.1 DB-2 Synthetic Sequences

In synthetic tests, we generate planar trajectories, 100 frames long, using a pseudo-random number generator ('rand' function in MATLAB) and band-limiting the signal to be under a specified frequency (function 'blim' in the MATLAB code shown in Table B.2). These trajectories are then projected onto two image planes using

Table B.2: DB-2 MATLAB code to generate synthetic trajectories.

```
Xstatic = [12 34 54 23];
Ystatic = [19 87 34 67];
Zstatic = 500 * ones(1, length(Xstatic));

% generate 3D dynamic moving points
frames=[1 : 100];
freqx=4;
freqy=2;
freqz=0.2; % k keep depth constant for planar motion

rand('state',sum(100*clock));
disp('*Generating Synthetic Trajectories');
for i = 1 : 8
    X(:, i) = 100 * blim(length(frames), freqx);
    Y(:, i) = 150 * blim(length(frames), freqy);
    Z(:, i) = 1 * blim(length(frames), freqz);
end
```

```
function xx=blim(n,m);
yy=rand(n,1)-0.5;
k1 = round(m/2);
% Define a symetric frequency-filter
filt = [ones(1,k1+1) zeros(1,n-2*k1-1) ones(1,k1)];
yy=real(ifft(fft(yy).*filt'));
xx=yy/norm(yy);
```

user defined camera projection matrices. One such representative projection matrix is:

$$P = \begin{bmatrix} -0.9037 & -0.4282 & 0 & 13.3187 \\ -0.3869 & 0.8167 & -0.4282 & -0.0153 \\ 0.1833 & -0.3869 & -0.9037 & 11.0730 \\ 0.0183 & -0.0387 & -0.0904 & 2.1073 \end{bmatrix}.$$

The camera matrices are designed so that the acquisition emulates a homography, and are used only for generation purpose and not thereafter. A time warp is then applied to a section of one of the trajectory projections, such that its length now becomes 140 frames. Both the RCB and the STE methods are then applied to the synthetic trajectories to compute the alignment between them. This process is repeated on 100 synthetic trajectories. In order to test the effect of noisy trajectories on the STE method, we add normally distributed zero mean noise to the trajecto-

ries from two synthetic sequences. A representative set of the noisy trajectories is shown in Fig.B.6.

B.2.2 DB-2 Real Sequences

We used video sequences provided by Rao *et al.* at http://server.cs.ucf. edu/∼vision/ and also acquired our own video sequences of activities similar to their data. The activities in the sequences were – (i) picking up a cup from a desk, placing the cup on a shelf and then bringing the cup back to the desk and (ii) opening a cabinet door. Feature trajectories were available for the UCF video files. For our test video sequences, we provided an input template image of a coffee cup that was tracked in the video sequences to generate feature trajectories.

UCF video sequence

Sequences captured from the University of Florida database varied in length from 84 frames to 174 frames. Frame size and frame rate of acquisition were 720×480 and 30 frames per second respectively. Some representative frames from UCF Sequence 1 are shown in Fig. B.7.

UofA Test Sequence

Sequences captured in the University of Alberta database varied in length from 40 frames to 270 frames. Frame size and frame rate of acquisition were 320×240 and 30 frames per second respectively. Some representative frames from the UofA Test Sequence 1 are shown in Fig. B.8.

Figure B.7: DB-2 Illustrative Frames from UCF test video. Sequence should be read row-wise from Top-left corner.

Figure B.8: DB-2 Illustrative Frames from UofA test video. Sequence should be read row-wise from Top-left corner.

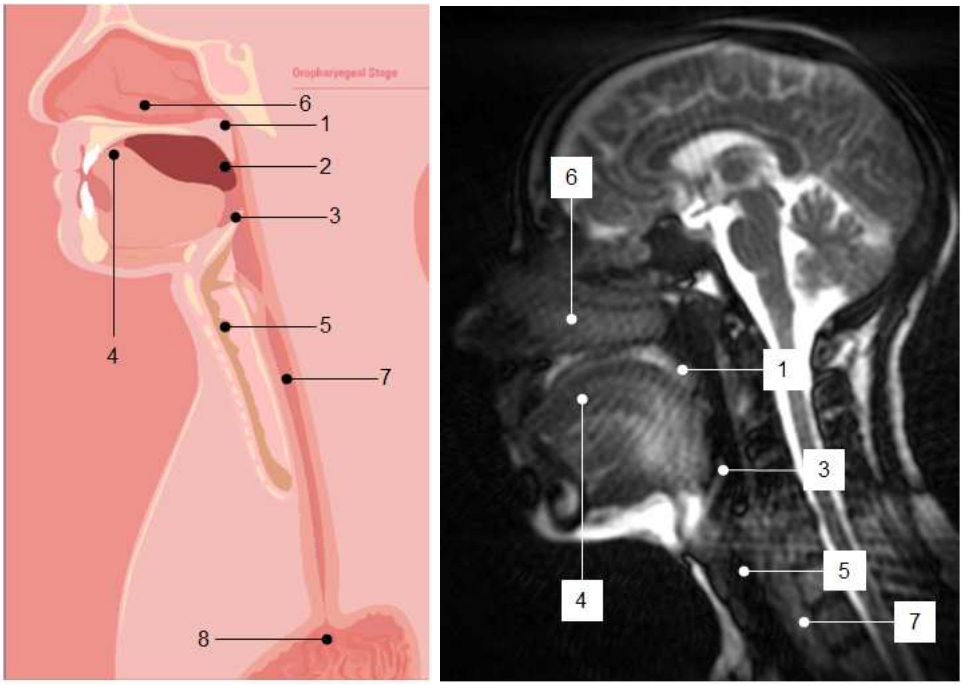
B.3 Database-3 MRI video sequences

MRI video sequences have been used in all the contributory chapters of this thesis (Chapter 3-6). In the following sections we describe the method of acquisition of the MRI sequences, map anatomical structures in representative images and show some frames from these sequences.

B.3.1 Acquisition

MRI data is acquired via radial acquisition as a series of radial projections though the center of k-space (animated video of radial acquisition is available online at [56]). For short scan times, radial acquisition results in higher spatial resolution of undersampled data as compared to conventional cartesian acquisition which suffers from aliasing and streaking artifacts. The MRI scan was conducted at the University of Alberta at the Centre for the NMR Evaluation of Human Function and Disease. All image data was acquired with subjects lying supine in a Siemens SonataTM 1.5T MRI scanner. Measured amounts of water (bolus) were delivered to the subject via a system of tubing and the swallow event was captured in the mid-sagittal plane. As current work deals with a prototype system, we captured only three repetitions of the swallow (available as video1, video2 and video3 at [56]). The data was acquired as 96 radial projections of 192 points and reconstructed to an image size of 384x384. Acquisition time for each image with the above configuration is 0.138 seconds, which computes to a frame rate of 7.2fps. Various anatomical structures related to the process of swallowing in the sagittal and coronal plane have been annotated in Fig. B.9 and Fig. B.16, respectively.

We acquire four video sequences corresponding to center (Fig. B.11),right (Fig. B.13) and left (Fig. B.15) MRI slice planes, as illustrated in Fig. B.10. The MRI video sequences are subjected to dynamic temporal offsets in the motion of the bolus. The trailing and leading edges of the bolus are extracted from the MRI sequences using standard background separation techniques [3]. The center of the trailing bolus is extracted using horizontal and vertical profiles, and is used to generate feature trajectories in the three sequences. In the following sections we illus-



(a) (b)

Figure B.9: DB-3 (a) Illustrative image showing major anatomical parts associated with swallowing. (b) MRI image depicting major anatomical parts. Legend: (1)– Soft Palate, (2)–Bolus, (3)–Epiglottis, (4)– Tongue, (5)– Trachea (or wind pipe), (6)–Naso-Pharynx (or nasal passage), (7)–Oesophagus, (8)– Stomach.

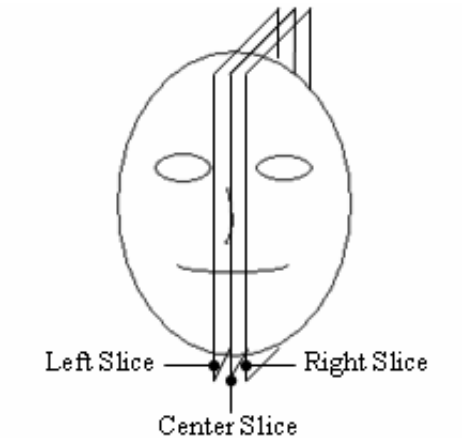


Figure B.10: DB-3 Illustration of spatial alignment of MRI slices.

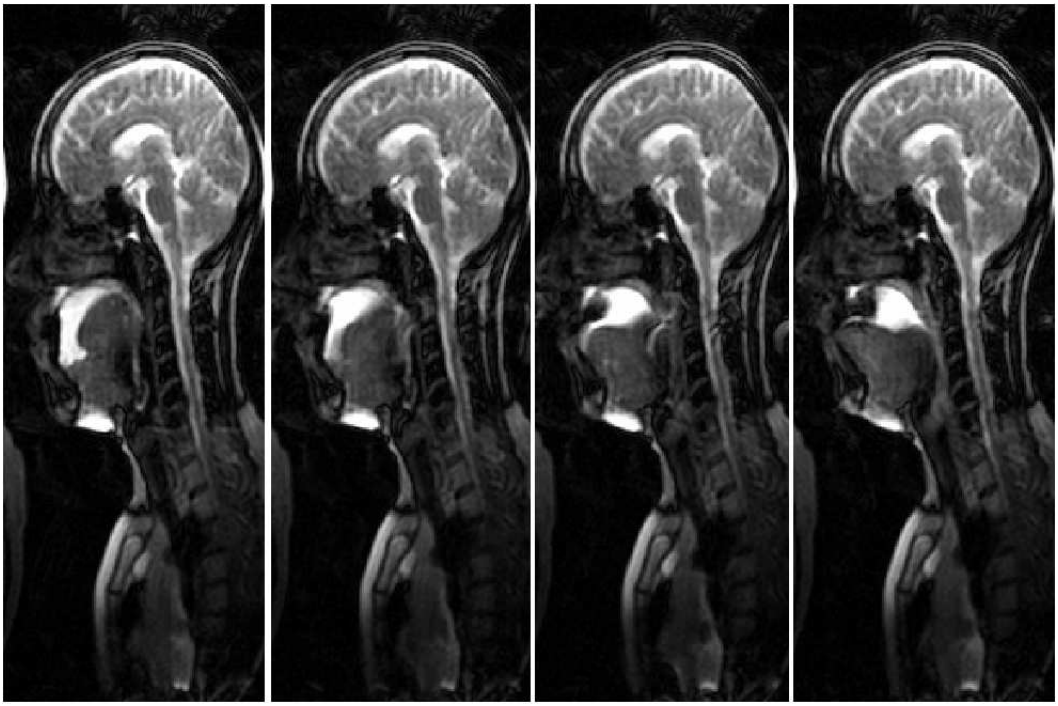


Figure B.11: DB-3 MRI frames for Center Sequence. Sequence should be read from Left to Right corner.

trate representative frames from the left, right, center and coronal (Fig. B.17) MRI sequences.

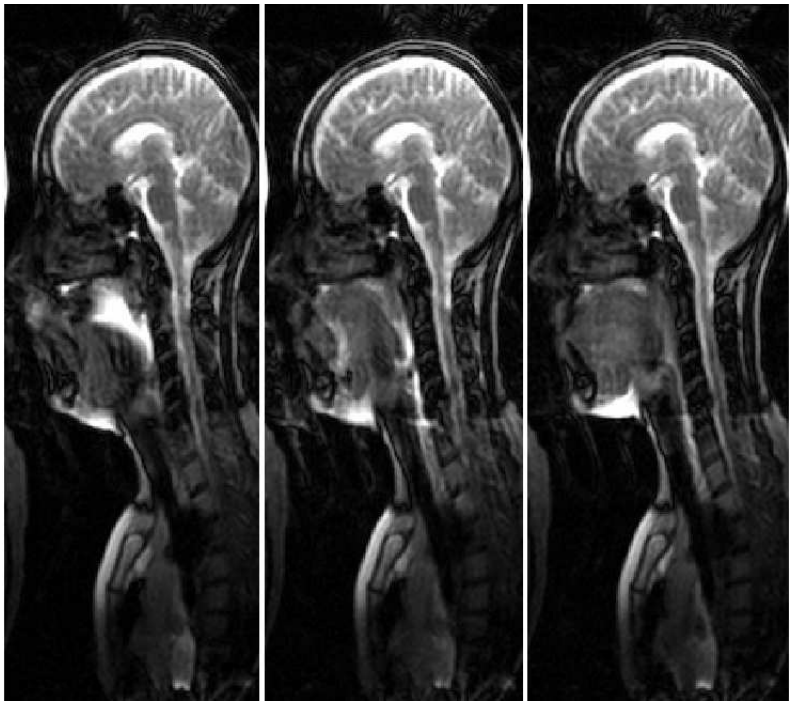


Figure B.12: DB-3 Fig. B.11 continued.

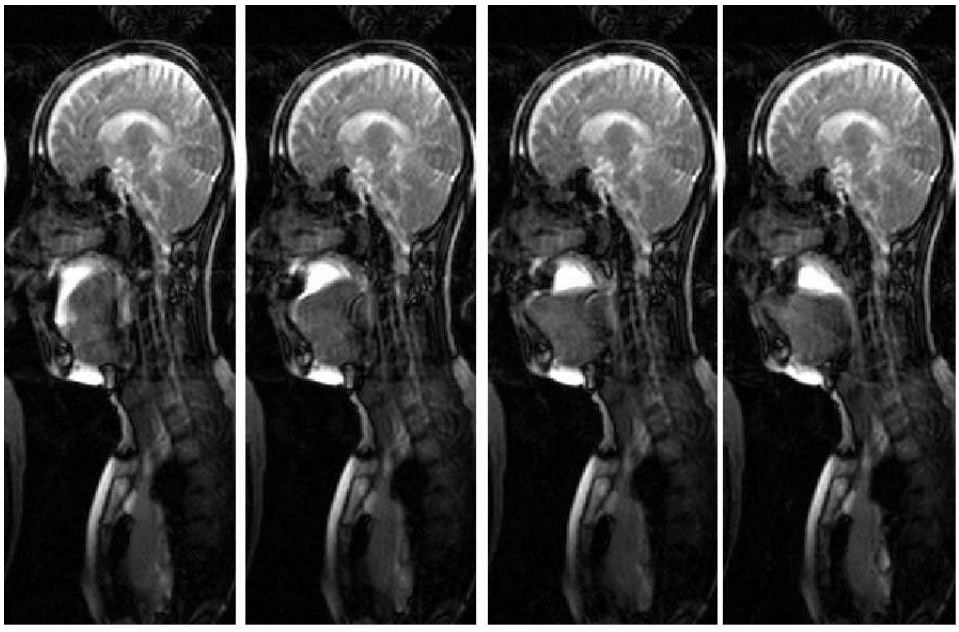


Figure B.13: DB-3 MRI frames for Right Sequence. Sequence should be read from Left to Right corner.

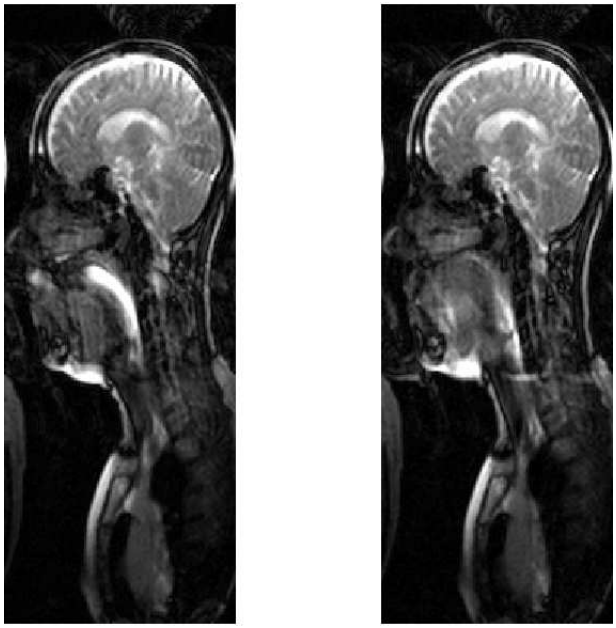


Figure B.14: DB-3 Fig. B.13 continued.

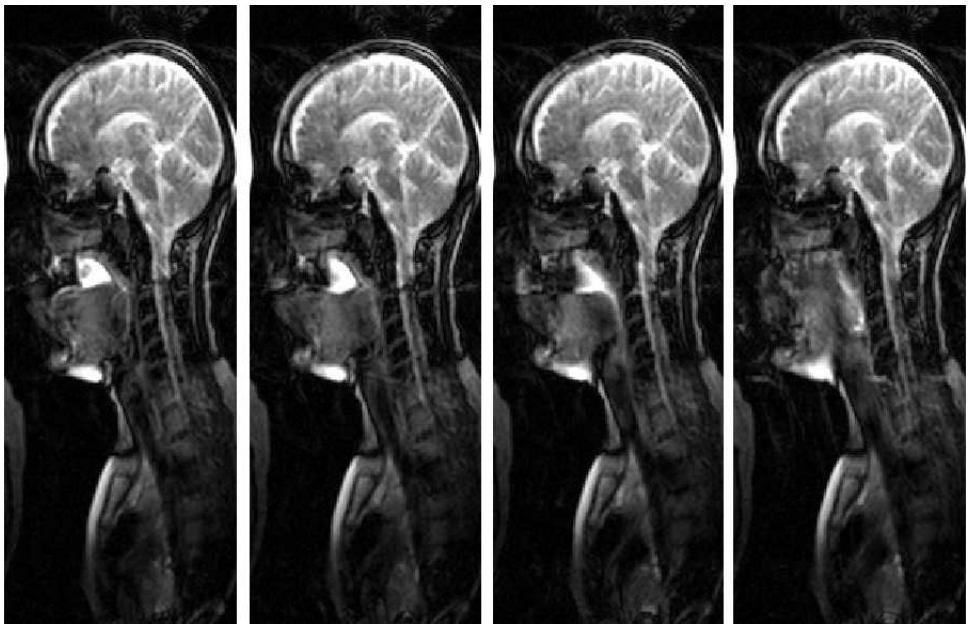


Figure B.15: DB-3 MRI frames for Left Sequence. Sequence should be read from Left to Right corner.

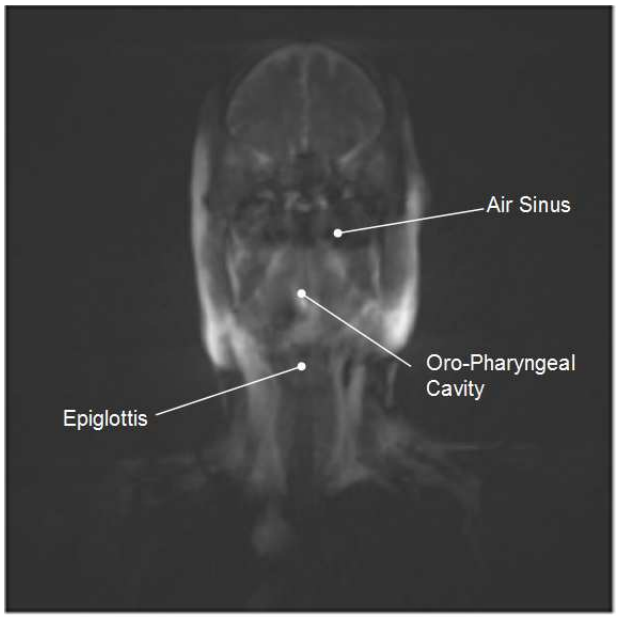


Figure B.16: DB-3 An illustrative Coronal MRI image with some anatomical regions marked.

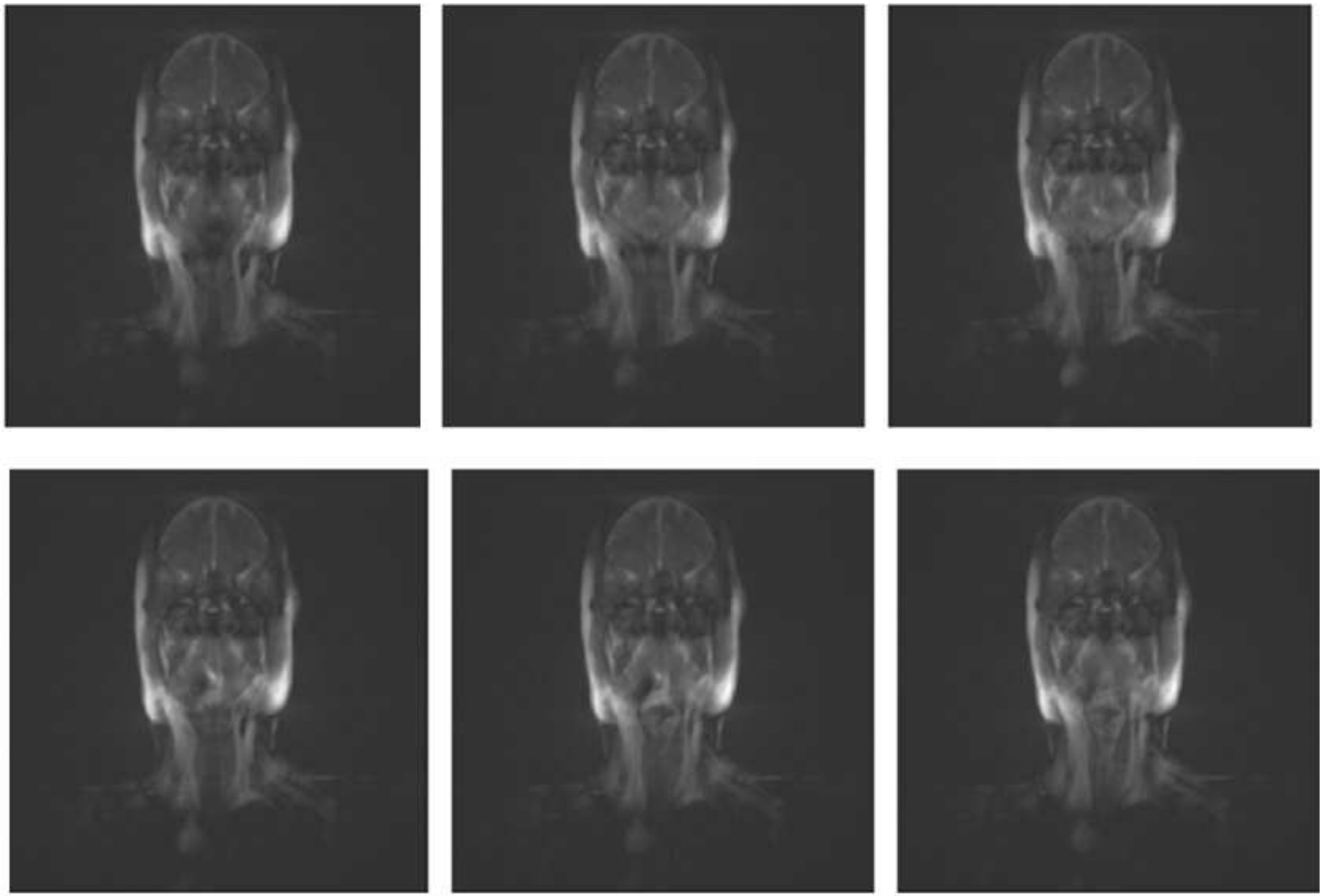


Figure B.17: DB-3 Coronal MRI frames showing the bolus approach the oro-pharyngeal cavity and splitting over the epiglottis.