

**FEATURE-BASED CORRESPONDENCES TO
INFER THE LOCATION OF ANATOMICAL
LANDMARKS**

by

Robert J. Tamburo

B.S., Delaware State University, 1999

M.S., University of Pittsburgh, 2002

Submitted to the Graduate Faculty of
the School of Engineering in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2006

UNIVERSITY OF PITTSBURGH
SCHOOL OF ENGINEERING

This dissertation was presented

by

Robert J. Tamburo

It was defended on

April 24, 2006

and approved by

G. Stetten, M.D., Ph.D., Associate Professor, Departments of Bioengineering and

Radiology

F. Boada, Ph.D., Associate Professor, Departments of Radiology and Bioengineering

J. R. Boston, Ph.D., Professor, Departments of Electrical and Computer Engineering,

Bioengineering, and Communications Science and Disorders

C.C. Li, Ph.D., Professor, Departments of Electrical and Computer Engineering, and

Computer Science

Dissertation Director: G. Stetten, M.D., Ph.D., Associate Professor, Departments of

Bioengineering and Radiology

Copyright © by Robert J. Tamburo

2006

ABSTRACT

FEATURE-BASED CORRESPONDENCES TO INFER THE LOCATION OF ANATOMICAL LANDMARKS

Robert J. Tamburo, Ph.D.

University of Pittsburgh, 2006

A methodology has been developed for automatically determining inter-image correspondences between cliques of features extracted from a reference and a query image. Cliques consist of up to three features and correspondences between them are determined via a hierarchy of similarity metrics based on the inherent properties of the features and geometric relationships between those features. As opposed to approaches that determine correspondences solely by voxel intensity, features that also include shape description are used. Specifically, medial-based features are employed because they are sparse compared to the number of image voxels and can be automatically extracted from the image.

The correspondence framework has been extended to automatically estimate the location of anatomical landmarks in the query image by adding landmarks to the cliques. Anatomical landmark locations are then inferred from the reference image by maximizing landmark correspondences. The ability to infer landmark locations has provided a means to validate the correspondence framework in the presence of structural variation between images. Moreover, automated landmark estimation imparts the user with anatomical information and can hypothetically be used to initialize and constrain the search space of segmentation and registration methods.

Methods developed in this dissertation were applied to simulated MRI brain images, synthetic images, and images constructed from several variations of a parametric model. Results indicate that the methods are invariant to global translation and rotation and can operate

in the presence of structure variation between images. The automated landmark placement method was shown to be accurate as compared to ground-truth that was established both parametrically and manually. It is envisioned that these automated methods could prove useful for alleviating time-consuming and tedious tasks in applications that currently require manual input, and eliminate intra-user subjectivity.

Keywords: feature-based correspondences, anatomical landmark placement, medialness, the correspondence problem, similarity metrics.

TABLE OF CONTENTS

| | | |
|------------|--|----|
| 1.0 | INTRODUCTION | 1 |
| 1.1 | Goals and Motivation | 1 |
| 1.2 | Summary of Goals And Claims | 4 |
| 1.3 | Description of Chapters | 7 |
| 2.0 | BACKGROUND | 8 |
| 2.1 | Overview of Medical Imaging | 8 |
| 2.2 | Medical Image Analysis | 11 |
| 2.2.1 | Medical Image Segmentation | 12 |
| 2.2.2 | Medical Image Registration | 16 |
| 2.3 | Medial-Based Image Analysis | 19 |
| 2.3.1 | Core Atoms | 20 |
| 2.3.2 | Medial Nodes | 21 |
| 3.0 | AUTOMATED FEATURE-BASED CORRESPONDENCES | 24 |
| 3.1 | Preliminaries | 26 |
| 3.2 | Unary Correspondences | 26 |
| 3.3 | Binary Correspondences | 29 |
| 3.3.1 | The Doublet Signature | 30 |
| 3.3.2 | The Binary Metric | 35 |
| 3.4 | Trinary Correspondences | 39 |
| 3.5 | Measuring Correspondence Sensitivity and Specificity | 43 |
| 3.6 | Empirical Results | 50 |
| 3.6.1 | Automated Feature-Based Correspondences | 53 |

| | | |
|------------|---|------------|
| 3.6.1.1 | Identical Synthetic Images | 53 |
| 3.6.1.2 | Rotational and Translational Invariance | 63 |
| 3.6.2 | Measuring Correspondence Sensitivity and Specificity | 87 |
| 4.0 | IMPLIED TRANSFORMS TO IMPROVE CORRESPONDENCES . | 90 |
| 4.1 | Computing Implied 3D Affine Transforms | 92 |
| 4.2 | Applying the Affine Transform and Computing Correspondence Likelihood | 96 |
| 4.3 | Empirical Results | 98 |
| 5.0 | GENERATING VALIDATION DATA FROM A PARAMETERIZED MODEL | 105 |
| 5.1 | Determining Intensity Range of Computer Generated Data | 106 |
| 5.2 | Creating Brain Structures From Shape Primitives | 108 |
| 5.3 | Empirical Results | 110 |
| 6.0 | AUTOMATICALLY ESTIMATING ANATOMICAL LANDMARK LOCATIONS | 113 |
| 6.1 | Defining Landmarks | 114 |
| 6.2 | Calculating Correspondences Between Landmarks | 116 |
| 6.3 | Correspondence Optimization to Infer Landmark Location | 118 |
| 6.4 | Initializing Query Landmarks | 126 |
| 6.5 | Landmark Correspondence Optimization Techniques | 126 |
| 6.5.1 | Brute Force Optimization | 129 |
| 6.5.2 | Gradient Ascent-Like Optimization | 139 |
| 7.0 | ASSESSING METRIC PERFORMANCE IN THE PRESENCE OF GEOMETRIC VARIATION: | 141 |
| 7.1 | Displacement Along the Cardinal Axes | 142 |
| 7.2 | Medial and Lateral Displacement | 146 |
| 7.3 | Scale Variation | 148 |
| 7.4 | Scale variation and the Unary Metric | 149 |
| 7.5 | Location and Scale Variation | 152 |
| 7.6 | Comparison to Manual Landmark Identification | 152 |
| 7.6.1 | Software Interface and Landmark Placement Protocol | 153 |

| | |
|--|------------|
| 7.6.2 Empirical Results | 153 |
| 8.0 PROGRAMMING TECHNIQUES | 159 |
| 9.0 CONCLUSIONS | 160 |
| 9.1 Claims Revisited | 161 |
| 9.2 Future Research Directions | 164 |
| APPENDIX A. SIGNATURE INVARIANCE TO ROTATION IN THE DEGENERATE SUBSPACE | 169 |
| APPENDIX B. COMPUTING CORRESPONDENCES EFFICIENTLY | 171 |
| BIBLIOGRAPHY | 175 |

LIST OF TABLES

| | | |
|-----|--|-----|
| 3.1 | Possible classifications for ROC curve analysis | 44 |
| 3.2 | Parameter settings used to extract medial nodes from images containing con- centric ellipsoids | 53 |
| 3.3 | Average and standard deviation calculations of difference measures between corresponding nodes for validation in the presence of rotation and translation | 86 |
| 3.4 | Calculated A_{ROC} for each correspondence metric | 89 |
| 4.1 | Core atom parameter settings for extracting medial nodes to demonstrate the triplet transform metric | 101 |
| 4.2 | Calculated A_{ROC} for each correspondence metric, including the triplet trans- form metric | 104 |
| 5.1 | Basic operations of constructive solid geometry | 108 |
| 5.2 | Geometric representations and assigned Hounsfield Units of modeled brain structures | 109 |
| 5.3 | Parameters used to model brain structures | 110 |
| 6.1 | Examples of potential landmarks for select brain structures | 115 |
| 6.2 | Outline for utilizing the correspondence framework to infer landmark locations in a query image | 121 |
| 6.3 | The set of landmarks used for demonstrating the brute force optimization technique | 131 |
| 6.4 | Figure of merit for each metric when the unfixed landmark is at its ground-truth | 133 |
| 6.5 | Figure of merit for each metric when the unfixed landmark is at the maximum | 133 |

| | | |
|-----|--|-----|
| 7.1 | Average landmark error and standard error for each displacement of the right eyeball along the cardinal axes | 145 |
| 7.2 | The average of each landmark's location error as a function of medial or lateral displacement | 148 |
| 7.3 | Each landmark location error for multiple values of b_μ | 150 |

LIST OF FIGURES

| | | |
|------|---|----|
| 2.1 | A homogeneous rectangle on a uniform background. Circles are bi-tangent and the dashed line indicates the medial axis of a rectangle as defined by Blum | 20 |
| 2.2 | Basic medial shapes with corresponding core atom configurations | 21 |
| 2.3 | The lambda triangle | 22 |
| 3.1 | Framework for identifying correspondences between cliques of medial nodes. . | 25 |
| 3.2 | Framework for identifying singlet correspondences with the unary metric . . . | 27 |
| 3.3 | A unary correspondence matrix for organizing unary correspondences | 28 |
| 3.4 | Identifying corresponding doublets with the binary metric | 30 |
| 3.5 | Signatures of two cylindrical nodes in different orientations | 32 |
| 3.6 | Cylindrical node 1 paired with cylindrical node 2 at the “equator” of node 1’s coordinate system | 33 |
| 3.7 | Signature of two cylindrical nodes at 45° “latitude” | 33 |
| 3.8 | Cylindrical node 1 paired with cylindrical node 2 at 45° “latitude” of node 1’s coordinate system | 34 |
| 3.9 | Two superimposed signatures illustrating the signature metric | 36 |
| 3.10 | A binary correspondence matrix for organizing binary correspondences | 37 |
| 3.11 | Identifying corresponding triplets with the trinary metric | 39 |
| 3.12 | Triplets from the reference and query images are shown | 40 |
| 3.13 | A trinary correspondence matrix for organizing binary correspondences . . . | 41 |
| 3.14 | Scatter plot of correspondence values versus ground-truth | 45 |
| 3.15 | Theoretical ROC curves for the medial correspondence metrics | 47 |
| 3.16 | Theoretical histograms for unary and binary correspondence metrics | 48 |

| | |
|---|----|
| 3.17 Computing the area under the ROC curve with trapezoidal rule | 49 |
| 3.18 Cross-sections of various images containing two concentric ellipsoids | 51 |
| 3.19 Cross-sections of an image obtained from the BrainWeb database | 51 |
| 3.20 The lambda triangle with eigenvalue thresholds defined | 52 |
| 3.21 An image consisting of two concentric ellipsoid and the set of medial nodes extracted from it | 54 |
| 3.22 Example of unary correspondences between nodes extracted from identical images containing two concentric ellipsoids | 56 |
| 3.23 Unary correspondence matrix calculated from identical images containing two concentric ellipsoids | 57 |
| 3.24 Three examples of binary correspondences between doublets extracted from identical images containing two concentric ellipsoids | 58 |
| 3.25 Binary correspondence matrix calculated from identical images containing two concentric ellipsoids | 59 |
| 3.26 Two examples of trinary correspondences between triplets extracted from iden- tical images containing concentric ellipsoids | 60 |
| 3.27 Trinary correspondence matrix calculated from identical images containing two concentric ellipsoids | 61 |
| 3.28 Correspondence matrices calculated from the concentric ellipsoids data set . . | 62 |
| 3.29 Images containing rotated and translated concentric ellipsoids | 64 |
| 3.30 Correspondences determined for the reference and query images shown in Fig. 3.29 | 65 |
| 3.31 Collection of nodes extracted from the rotated concentric ellipsoids data set . | 66 |
| 3.32 Two examples of unary correspondences between nodes extracted from images containing two concentric ellipsoids, one of which is rotated | 67 |
| 3.33 Unary correspondence matrix calculated from images containing two concen- tric ellipsoids, one of which is rotated | 68 |
| 3.34 Three examples of binary correspondences between doublets extracted from identical images containing concentric ellipsoids. | 70 |

| | |
|--|----|
| 3.35 Binary correspondence matrix calculated from identical images containing two concentric ellipsoids | 71 |
| 3.36 Two examples of trinary correspondences between triplets extracted from identical images containing concentric ellipsoids | 72 |
| 3.37 Trinary correspondence matrix calculated from identical images containing two concentric ellipsoids | 73 |
| 3.38 Correspondence matrices calculated to demonstrate the effects of global image rotation on the correspondence metrics | 75 |
| 3.39 Data set containing two concentric ellipsoids for demonstrating translational invariance | 76 |
| 3.40 Two examples of unary correspondences between nodes extracted from images that contain concentric ellipsoids, which differ by translation | 77 |
| 3.41 Unary correspondence matrix calculated for nodes extracted from images that contain concentric ellipsoids, which differ by translation | 78 |
| 3.42 Three examples of binary correspondences between doublets extracted from images containing two concentric ellipsoids for demonstrating translational invariance | 79 |
| 3.43 Binary correspondence matrix calculated from images containing two concentric ellipsoids for demonstrating translational invariance | 80 |
| 3.44 Two examples of trinary correspondences between triplets extracted from two images containing concentric ellipsoids, one of which is translated | 81 |
| 3.45 Trinary correspondence matrix calculated from two images containing concentric ellipsoids, one of which is translated | 82 |
| 3.46 Correspondence matrices calculated from the translated concentric ellipsoids data set | 84 |
| 3.47 Image obtained from the BrainWeb simulated brain database with medial nodes overlayed | 87 |
| 3.48 Calculated ROC curves for each correspondence metric | 89 |
| 4.1 Implied transform between corresponding triplets | 91 |
| 4.2 Integrating implied affine transforms into the correspondence framework | 91 |

| | | |
|-----|--|-----|
| 4.3 | A triplet tetrahedron formed from an equilateral triplet triangle | 93 |
| 4.4 | A triplet tetrahedron to illustrate the derivation of h | 94 |
| 4.5 | Computing the triplet transform metric | 98 |
| 4.6 | Improvement of correspondences with triplet transforms demonstrated on cor- respondence matrices | 99 |
| 4.7 | BrainWeb image used to demonstrate the triplet transform metric shown with extracted medial nodes | 102 |
| 4.8 | Correspondence matrices calculated by evaluating medial nodes extracted from identical BrainWeb images | 103 |
| 4.9 | Calculated ROC curves for each correspondence metric, including the triplet transform metric | 104 |
| 5.1 | Assigning gray values to image pixels by windowing the range of Hounsfield Units | 106 |
| 5.2 | Cross-sections and 3D rendering of a computer generated CT brain with struc- tures labeled | 111 |
| 5.3 | Cross-sections and renderings of a synthetic brain image | 112 |
| 6.1 | Reference image depicting an eyeball and query image depicting an eyeball with geometric variation. | 114 |
| 6.2 | Matrix consisting of correspondences and cross-correspondences between nodes and landmarks | 117 |
| 6.3 | System for inferring landmark locations from clique correspondences | 118 |
| 6.4 | Inferring landmark locations in a query image from correspondences | 120 |
| 6.5 | Ideal landmark correspondence matrix | 123 |
| 6.6 | Cross-sections of a synthetic brain image with the sets of medial nodes and landmarks used to demonstrate the brute force optimization technique. | 130 |
| 6.7 | Correspondence matrices shown with extracted landmark correspondence ma- trices when the unfixed landmark is at ground-truth | 132 |
| 6.8 | Figure of merit calculated with cross-correspondence penalty as a function of landmark error | 134 |

| | | |
|------|--|-----|
| 6.9 | Figure of merit calculated without cross-correspondence penalty as a function of landmark error | 135 |
| 6.10 | The unfixed landmark's correspondence metric value as a function of landmark error | 136 |
| 6.11 | The figure of merit is displayed for the landmark in the x-y plane for various z-coordinates | 137 |
| 6.12 | The figure of merit is displayed for the landmark in the x-y plane for various z-coordinates | 138 |
| 7.1 | Landmark error vs. displacement of the right eyeball along the x-axis | 142 |
| 7.2 | Landmark error vs. displacement of the right eyeball along the y-axis | 143 |
| 7.3 | Landmark error vs. displacement of the right eyeball along the z-axis | 144 |
| 7.4 | Landmark error vs. medial displacement of the eyeballs | 146 |
| 7.5 | Landmark error vs. lateral displacement of the eyeballs | 147 |
| 7.6 | Landmark error as a function of change in node scale for the nodes extracted from the eyeball | 149 |
| 7.7 | Trinary correspondence matrices calculated with the query landmarks placed at their estimated locations while varying the scale term of the unary metric b_μ | 151 |
| 7.8 | Average landmark error for each trial as the right eyeball was displaced along the x-axis | 154 |
| 7.9 | Average landmark error for each trial as the right eyeball was displaced along the y-axis | 155 |
| 7.10 | Average landmark error for each trial as the right eyeball was displaced along the z-axis | 155 |
| 7.11 | Comparison of manual versus automated method of estimating landmark location in the presence of medial displacement of the eyeballs | 156 |
| 7.12 | Comparison of manual versus automated method of estimating landmark location in the presence of lateral displacement of the eyeballs | 157 |
| 7.13 | Comparison of methods for estimating landmark location in the presence of lateral displacement and scale variation of the eyeballs | 158 |

| | | |
|-----|---|-----|
| 9.1 | Image of the contrast-enhanced CT chest scan with a surface model of the segmentation, the set of extracted medial nodes, and aortic arch landmarks . | 165 |
| 9.2 | Correspondence matrices computed for nodes extracted from the aorta segmentation and landmarks placed on the aorta aortic arch | 166 |
| B1 | Data structure for efficiently storing doublet correspondences | 173 |

LIST OF NOTATION (ACRONYMS)

| <u>ACRONYM</u> | <u>DESCRIPTION</u> | <u>PAGE</u> |
|----------------|--|-------------|
| 3D | Three-Dimensional | 1 |
| 2D | Two-Dimensional | 8 |
| NMR | Nuclear Magnetic Resonance | 8 |
| RF | Radio Frequency | 8 |
| MRI | Magnetic Resonance Imaging | 8 |
| CT | Computed Tomography | 9 |
| ECG | electrocardiogram | 9 |
| RT3D | Real-Time 3-Dimensional | 9 |
| cine-CT | Cine Computerized Tomography | 9 |
| FLAIR | FLuid Attenuation Inversion Recovery | 10 |
| SPGR | SPoiled Gradient Recall | 10 |
| fMRI | Functional MRI | 10 |
| SPECT | Single Photon Emission Computed Tomography | 10 |
| PET | Positron Emission Tomography | 10 |
| PCA | Principal Component Analysis | 15 |
| ICP | Iterative Closest Point | 17 |
| DoOG | Difference of Offset Gaussians | 20 |
| ROC | Receiver Operator Characteristic | 25 |
| NCS | Normalized Component Squared | 31 |
| ROC | Receiver Operating Characteristic | 43 |
| TP | True Positive | 44 |
| FP | False Positive | 44 |

| | | |
|--------|--|-----|
| FN | False Negative | 44 |
| FP | False Positive | 44 |
| TPF | True Positive Fraction | 45 |
| FPF | False Positive Fraction | 45 |
| SBD | Simulated Brain Database | 50 |
| RMS | Root Mean Squared | 97 |
| HU | Hounsfield Unit | 106 |
| CSG | Constructive Solid Geometry | 108 |
| MCTILL | Medial Correspondences To Infer Landmark Location | 118 |
| LCM | Landmark Correspondence Matrix | 122 |
| LDF | Landmark Displacement Function | 124 |
| BFO | Brute Force Optimization | 129 |
| GALO | Gradient Ascent-Like Optimization | 139 |
| ITK | Insight Toolkit | 159 |
| NLM | National Library of Medicine | 159 |
| FLTK | Fast Light Toolkit | 159 |
| VTK | Visualization Toolkit | 159 |

LIST OF NOTATION (VARIABLES)

| <u>VARIABLE</u> | <u>DESCRIPTION</u> | <u>PAGE</u> |
|--------------------|--|-------------|
| M_α | number of medial nodes extracted from image α | 26 |
| $n_{\alpha,i}$ | node i from image α | 26 |
| $C_{\alpha,i}$ | the i^{th} clique of nodes from image α | 26 |
| \mathbf{C}_m | correspondence matrix calculated from metric m | 26 |
| Ψ_α | set of medial nodes extracted from image α | 26 |
| N | image dimension | 26 |
| R | label to denote the reference image | 26 |
| Q | label to denote the query image | 26 |
| α | variable for image designation | 26 |
| $\mu(j, k)$ | unary metric calculated for singlets j and k | 27 |
| ω | normalization constant for the unary metric | 27 |
| μ_T | threshold for determining unary correspondences | 29 |
| \mathbf{p}_i | spatial coordinate of node i | 30 |
| \mathbf{d} | location of a node in the coordinate system of another ... | 31 |
| d_i | i^{th} component of \mathbf{d} | 31 |
| $\hat{\mathbf{d}}$ | normalized \mathbf{d} | 31 |
| $\ \mathbf{d}\ $ | magnitude of \mathbf{d} | 31 |
| $(\hat{d}_i)^2$ | i^{th} NCS of \mathbf{d} | 31 |
| \mathbf{v} | displacement vector between medial nodes | 31 |
| v_j | j^{th} component of \mathbf{v} | 31 |
| $s(\lambda)$ | signature of a doublet | 31 |
| $u(\cdot)$ | the unit step function | 31 |

| | | |
|---------------------------|---|----|
| $s(\lambda)_{\alpha,i,j}$ | the signature for a doublet consisting of nodes i and j , extracted from image α | 35 |
| $S_{i,j}$ | signature metric evaluated for doublets consisting of nodes i and j | 35 |
| κ | takes on the value 1 or -1 for computing discrete signature metric | 36 |
| $\mathbf{p}_{\alpha,i}$ | spatial coordinate of node i from image α | 36 |
| $d_{\alpha,i,j}$ | Euclidean norm between nodes i and j from image α | 36 |
| β_o | orientation term of the binary metric | 37 |
| β_d | distance term of the binary metric | 37 |
| β | the binary metric for a given pair of doublets | 37 |
| \mathbf{C}_β | binary correspondence matrix | 37 |
| β_T | threshold for determining binary correspondences | 38 |
| θ_α | angle between 2 doublets of a triplet | 40 |
| L_α | distance between a doublet nodes in image α | 40 |
| \mathbf{C}_τ | trinary correspondence matrix | 41 |
| τ_T | threshold for trinary metric | 42 |
| $D_{a,i}$ | correspondence ground-truth for nodes a and i | 44 |
| $k_{a,i}$ | Euclidean distance between nodes a and i | 44 |
| k_{max} | maximum distance computed between a set of nodes | 44 |
| ϕ_D | parameter for correspondence ground-truth | 44 |
| C_T | threshold for a correspondence value | 44 |
| D_T | threshold for the ground-truth | 44 |
| N_{FN} | the number of false negatives | 45 |
| N_{TP} | the number of true positives | 46 |
| N_{TN} | the number of true negatives | 46 |
| N_{FP} | the number of false positives | 46 |
| A_{ROC} | area under an ROC curve | 49 |
| $f(x_i)$ | a function of x_i | 49 |
| \mathbf{r}_i | 3D point in image R | 92 |

| | | |
|------------------------|--|-----|
| \mathbf{q}_j | 3D point in image Q | 92 |
| \mathbf{R}_i | homogeneous coordinates of \mathbf{r}_i | 92 |
| \mathbf{Q}_j | homogeneous coordinates of \mathbf{q}_j | 92 |
| t_i | translation along cardinal axis i | 92 |
| $\hat{\mathbf{n}}$ | vector orthogonal to the plane of a triplet triangle | 93 |
| \mathbf{c}_m | center of mass of a triplet triangle | 93 |
| \bar{L} | average length of a triplet triangle's sides | 95 |
| $\ \mathbf{L}_{i,j}\ $ | magnitude of the vector between points i and j | 95 |
| χ | triplet transform metric | 97 |
| $L_{\alpha,i}$ | The i^{th} landmark from the landmark set from image α ... | 115 |
| Γ_J | The collection of landmarks from the J^{th} landmark set ... | 115 |
| \mathbf{K}_L | landmark correspondence matrix from the correspondence matrix calculated with the L metric | 123 |
| $E_{L,i}$ | location error for the i^{th} estimated landmark in Q | 125 |
| \bar{E}_L | overall location error measure for the set of nodes in Q ... | 125 |
| \mathbf{A} | matrix whose columns are eigenvectors, $\hat{\mathbf{a}}_i$ for $i = 1, 2, \dots, N$ | 169 |
| \mathbf{E} | matrix whose columns are rotated eigenvectors $\hat{\mathbf{e}}_i$ for $i =$ $1, 2, \dots, N$ | 169 |
| \mathbf{T}_R | rotation transform matrix | 169 |
| V^3 | number of voxels in an image | 171 |

PREFACE

A number of people have played a key role in my life during the completion of my graduate studies. I am truly grateful to my advisor and mentor, Dr. Stetten... I mean George. It was his guidance, encouragement, and constant support that certainly contributed to the completion of my dissertation research. He taught me to find the nugget. I thank C. C. Li, Fernando Boada, and Bob Boston for taking the time to serve on my committee and providing insightful comments and suggestions. I am very appreciative of all past and present members of VIA lab for their assistance through the years. Once upon a time, it was just me in the lab; sitting at a desk and surrounded by boxes. Eventually, Aaron Cois joined the lab and has been “there” ever since. Thanks for the work on the early version of the correspondence algorithm. Thanks to Vikram Chib and our tapeball games that were always a source of relief. Thanks to Damion Shelton for developing Chimera, which became the basis of my programming applications. Thanks to Wilson Chang for being a motivation. Thanks to John Galeotti for being evil, in a good way. Many thanks to the Silvestri’s for opening their arms to me. Erica, thanks for being in my life. Most importantly, I thank my small family for their unconditional love and support. Mom, thanks for being you. You are truly the best mom in the entire world. Dad, thanks for being so encouraging throughout our years together.

1.0 INTRODUCTION

1.1 GOALS AND MOTIVATION

Medical imaging has permitted clinicians and researchers to look beneath the surface of the skin to obtain critical diagnostic information and scientific insight. Computational techniques such as segmentation (separating objects from their neighbors) and registration (aligning images to each other) have been developed to aid with the analysis of these images, but have not advanced at the same rate as the imaging technologies themselves. Many such methods are manual or semi-automated requiring varying degrees of user interaction, making them tedious and time-consuming, especially for three-dimensional (3D) images. Additionally, these methods require users to be trained, and results are often subjective. Complete automation has been difficult to achieve due to inherent anatomical variation, image noise, and discontinuous structure boundaries. Nonetheless, the potential benefits to the medical and research communities are motivation for the continued development of automated image analysis techniques. For example, a method for automatically and accurately segmenting anatomical structures would provide a means to calculate quantitative measures of the underlying anatomy to assist clinicians in identifying pathologies, making diagnoses, and treatment planning.

An area of image analysis that deserves attention is the *correspondence problem* because it forms the basis of many image analysis methods. The correspondence problem can be stated as matching a specified element in an image to its corresponding element in another image. If such a correspondence can be established, these elements share the same identity or label. Determining correct correspondences is nontrivial for several reasons. Typically, correspondences can be determined once the images have been aligned with a spatial trans-

form. However, solving for rigid or nonrigid transforms with unknown correspondence is a difficult problem. Moreover, the correspondence problem is usually ill-posed since there may be several possibilities for correct correspondences. There are two main approaches for establishing correspondences: those based on intensity correlation and those based on feature similarity.

Correlation-based methods establish correspondences by sampling a local neighborhood of voxels (window) in the first image, and shifting the window across the second image until the correlation between the window and the underlying voxel intensities is maximum. This approach relies on a semi-uniform intensity mapping between the two images making it sensitive to image noise, structure distortion, scanner settings, etc. Correlation-based methods tend to be computationally expensive since each window may have to be shifted across the entire second image. The accuracy of such methods can be moderated by utilizing different window sizes. In the presence of image noise and distortion, smaller windows can localize the corresponding point more accurately than larger windows.

Alternatively, *feature-based methods* establish correspondences by matching features extracted from the images (e.g., points, lines, ridges, corners) by means of a similarity measure. The set of features is sparse compared to the set of image voxels, and thus establishing correspondences between features is less demanding than establishing correspondences between voxels or patches of voxels. The success of feature-based methods is dependent on how well the similarity measure can differentiate between features.

Both approaches to the correspondence problem are commonly used and suffer from similar difficulties in the presence of image noise and structure deformation. The feature-based approach is more attractive because only sparse sets of features are considered, and the method, depending on the features used, can be flexible to variations in intensity and shape. In this light, a correspondence hierarchy based on measuring the similarity between *cliques* of features has been developed in this dissertation. Each type of clique consists of a *singlet*, *doublet*, or *triplet* of features. Cliques of multiple features describe the geometric relationship between the comprising features. Inter-image clique correspondences are determined via a hierarchy of similarity metrics. The metric for identifying correspondences between singlets, doublets, and triplets are the *unary*, *binary*, and *trinary* metrics, respectively.

By means of Receiver Operator Characteristic (ROC) curve analysis, the accuracy in determining correct clique correspondences can be quantified for each metric. As will be shown, correspondence accuracy improves with each successive metric. Although the trinary metric has greater accuracy over the unary and binary metrics, incorrect correspondences still exist. A second goal of this dissertation is to extend the metric hierarchy to further reduce the number of incorrect correspondences by introducing a global analysis of the cliques based on implied transforms.

Another goal of this dissertation is to explore the use of correspondences to automatically place labeled, anatomical landmarks in an image. Anatomical landmarks identify locations of interest to a human user. Landmarks differ from image features in that they are directly related to underlying anatomy and have unique identities, whereas image features are locally processed from the image itself and are not necessarily unique. Analogous to landmark-based registration, a *reference image* containing predefined landmarks is used to place those landmarks in a new *query image*. By adding landmarks to cliques containing features, landmark location can be inferred from the correspondences. For example, consider a clique from the reference image that consists of features from the thalamus, amygdala, and pons. A clique with a similar geometric relationship between structures is sought in the query image. Once that correspondence is established, a landmark can be automatically placed in the pituitary gland based on its relationship to the other structures given that such a landmark was previously placed in the pituitary gland of the reference image.

It is envisioned that automatic landmark placement could be employed for labeling anatomical regions, define regions of interest, initialize segmentation or registration algorithms, or build atlases or models. In this dissertation, estimating landmark location will play a key role in validating correspondences. In practice, a lack of ground-truth for correspondences is a problem for evaluating methods because surfaces are complex and variable, resulting in subjective interpretation. Correspondences will be validated by how accurately they can infer the location of specified landmarks under various geometric perturbations. Ground-truth will be established from a parametric model of a human brain. The parametric model will deliver exact landmark locations, and allow for systematic variation of the model geometry. Images generated from the parametric model will permit the extrac-

tion of sets of image features for testing the robustness of the correspondence and landmark placement methods. The features used in this dissertation are called *medial nodes*, which encapsulate the local medial properties of shape providing rotationally invariant shape properties of dimensionality and scale localized on the medial manifold of an object. The method for extracting medial nodes is discussed in Section 2.3.1.

Independent of anatomical target and imaging modality, it is envisioned that determining feature correspondences and estimating the location of anatomical landmarks could be accomplished by an automated system that would serve as a tool for recognizing anatomical shapes or foreign bodies, as well as making critical diagnostic measurements. The system would alleviate tedious work by clinicians and researchers, and could also be useful to medical students and other healthcare professionals as an interactive teaching utility.

1.2 SUMMARY OF GOALS AND CLAIMS

The goals of this dissertation are outlined below with each associated claim:

1. Automatically determine feature-based correspondences:

As opposed to approaches that determine correspondences solely by intensity, correspondences will be determined between cliques of image features by utilizing geometric constraints and a hierarchy of similarity metrics. Receiver Operator Characteristic curve analysis will be employed to initially assess the accuracy of each correspondence metric.

Claim 1: *By using a hierarchy of similarity metrics, inter-image correspondences between cliques of features can be determined.*

2. Improve correspondences with implied triplet transforms:

The accuracy of the correspondence framework will be improved upon by introducing a global analysis of triplet configurations. Each triplet correspondence implies a transform that maps a triplet in the query image to a corresponding triplet in the reference image. From these transforms, a *correspondence likelihood* will be calculated for every possible

triplet correspondence. The correspondence likelihood will strengthen true correspondences and weaken false correspondences.

Claim 2: *By incorporating global geometric constraints into the correspondence hierarchy, correspondence accuracy can be improved.*

3. Generate validation data from a parameterized model:

Images will be computer generated from a parametric model to approximately emulate a Computed Tomography image of a human brain. Brain structures will be modeled by parametrically defined shape primitives, and combinations of such shapes. The use of parametrically defined data will permit systematic perturbation of structure geometry and establishing ground-truth for landmark location towards the goal of assessing metric performance (Goal 5).

Claim 3: *By defining a parametric model, it is possible to establish ground-truth towards the goal of validating correspondences in the presence of variations in structure geometry.*

4. Automatically estimate anatomical landmark locations:

Given that landmarks have been manually or parametrically placed in a reference image, the same landmarks will be automatically placed in a query image by incorporating landmarks into cliques and the correspondence framework. Landmark locations will be inferred from correspondences and geometric relationships between cliques that consist of nodes and landmarks by means of an optimization procedure that maximizes landmark correspondences.

Claim 4: *By incorporating landmarks into the correspondence framework, their locations can be inferred from feature-correspondences and automatically placed in a query image to validate the correspondence method, as well as provide anatomical information to the user.*

5. Assess metric performance in the presence of geometric variation:

Metric performance will be studied as a function of geometric variations between structures. The parametric model (Goal 3) allows for systematic perturbation of the data to

create query images that differ from the reference image and establishes ground-truth for landmark locations. The anatomical landmarks discussed in Goal 4 will establish a ground-truth independent of these variations. Correspondence metrics will be validated by their accuracy in estimating landmark locations in the query image.

Claim 5: *By computing the accuracy of inferring the location of anatomical landmarks in a query image, correspondences can be validated for images that differ by structure geometry.*

1.3 DESCRIPTION OF CHAPTERS

Including the current chapter, this dissertation consists of 9 chapters and a bibliography.

| | |
|------------------|--|
| Chapter 2 | An overview of medical imaging, medical image segmentation and registration, landmark extraction, and medial-based image analysis is provided. |
| Chapter 3 | The framework for determining clique correspondences and an initial method for quantifying and validating correspondence accuracy is detailed. |
| Chapter 4 | A method for improving correspondences by incorporating global transform constraints is described. |
| Chapter 5 | The parametric brain model and validation data are defined. |
| Chapter 6 | Anatomical landmarks are defined and the optimization procedure for inferring landmark locations from clique correspondences is developed. |
| Chapter 7 | Results of inferring anatomical landmark location in the presence of geometric variations of the underlying anatomy are reported. |
| Chapter 8 | Programming techniques utilized in this dissertation are described. |
| Chapter 9 | Conclusions of this dissertation, claims are reviewed, and potential directions of future research are explored. |

2.0 BACKGROUND

2.1 OVERVIEW OF MEDICAL IMAGING

After Dr. Roentgen inadvertently discovered X-rays in 1895, they were almost immediately applied to medicine when Dr. Edwin Frost performed the first clinical X-ray scan in early 1896. These projected images dramatically affected the medical community by providing a means to non-invasively look into the body of a patient. The newly acquired visual information assisted doctors and other healthcare professionals in making diagnostic decisions. The clinical value of such images was immediate and spawned the invention of other modalities of medical imaging.

High frequency sound, called ultrasound, was first generated in 1876 by Francis Galton, but would not be applied to medicine until 1937 when Karl Dussik began investigating its use as a diagnostic tool by visualizing brain structures and making ventricular measurements. Dussik *et al.* presented their initial findings in 1942 [1] and later published ultrasound images of a human brain in 1947 [2]. In 1965, engineers at Siemens created the world's first real-time ultrasound machine, generating live two-dimensional (2D) images and displaying them on a video screen.

Nuclear Magnetic Resonance (NMR) spectroscopy was first described by Felix Bloch and Edward Mills Purcell in 1946. Bloch and Purcell observed that nuclei in a magnetic field absorbed energy if a Radio Frequency (RF) pulse at the nuclei's resonant frequency was applied. A spectrum of the magnitude of the NMR resonance signal for a molecule was later found to describe its structure. For 26 years, NMR was primarily used to study molecular properties until Paul Lauterbur extended NMR to produce images of internal body structures. This new technique, called Magnetic Resonance Imaging (MRI), operates

by introducing gradients into the magnetic field allowing for spatial localization of emitted RF waves, and thus 2D tomographic images can be constructed and stacked sequentially to form 3D images [3].

In 1972, Godfrey Hounsfield utilized X-ray technology and built the first Computed Tomography (CT) prototype. In a CT scanner, an X-ray source and X-ray sensors are positioned on opposite sides of a circular gantry. The gantry rotates 360° to collect a single slice of data. The subject is then moved forward and the gantry rotates 360° in the opposite direction to collect another scan. Once the entire set of scans is complete, the data is reconstructed to form, like MRI, 2D tomographic images that can be stacked to form 3D images [4].

Each of these imaging techniques offers a unique insight into the human body. For instance, ultrasound produces real-time 2D images that are useful for observing the behavior and condition of dynamic objects, such as the heart and a fetus. Two-dimensional ultrasound images can be reconstructed to 3D after data acquisition, but are no longer real-time and have poor image quality compared to MRI and CT images. MRI and CT produce high-resolution images that are effective for visualizing objects in the body, but require significant time for image reconstruction following data acquisition [5]. Dynamic objects can also be studied with CT and MRI by gating the scans to the electrocardiogram (ECG). Unfortunately, gated scans are averaged over many cardiac cycles degrading the resolution of the image.

These imaging techniques continue to rapidly evolve, offering faster data acquisition, quicker image reconstruction, and higher image resolution. For example, at Duke University, in the early 1990s Real-Time 3-Dimensional (RT3D) ultrasound was developed. RT3D utilizes a matrix array transducers to capture 3D ultrasound data at a scan rate fast enough to acquire cardiac data throughout its cycle [6, 7].

In the mid-1980s, the power slip ring was developed, which transfers electrical power from a stationary power source to a continuously rotating gantry. This new technology lead to the development of spiral CT in which the gantry continuously rotates around the patient resulting in much faster scan times as compared to conventional CT. An even faster method of acquiring CT images called Cine Computerized Tomography (cine-CT) was later developed, which does not require any mechanical motion. In cine-CT an electron beam is

swept across a large, stationary anode that surrounds the patient. When electrons strike the anode, X-rays are emitted, pass through the patient, and get measured by a stationary array of detectors. Images from a cine-CT system are acquired very rapidly, permitting non-gated studies of cardiac function, though at a lower resolution than conventional CT.

In the field of MRI, much research has been conducted in manipulating pulse sequences and other scanner parameters to produce MRI images that contrast tissue differently. An example of such structural MRI images are T_1 -weighted, T_2 -weighted, proton density, FLuid Attenuated Inversion Recovery (FLAIR), and SPoiled Gradient Recalled (SPGR) acquisition [8]. In addition to imaging structure, functional information can be imaged via the hemodynamic response related to neural activity in the brain. This form of MRI is called Functional MRI (fMRI) and utilizes an MRI pulse sequence to detect blood oxygenation, which can be an indication of activation areas in the brain.

Physiological processes can also be imaged with nuclear imaging, which involves the injection of radioactive tracers into the patient for diagnosis and therapy. Tracers can be designed to tag molecules for targeting specific sites. After the tracer is administered to the patient, the patient is positioned in front of a detector array, and the radiation emitted from the body is measured and constructed into an image. These images are fairly low resolution with a high amount of noise due to the inability of using high radiation doses. The detector array can be stationary, collecting a single projection image, or it can be made to spin around the patient as in CT to collect tomographic images. The two main tomographic methods in nuclear medicine imaging are Single Photon Emission Computed Tomography (SPECT) and Positron Emission Tomography (PET). In SPECT the tracers used emit gamma radiation while decaying and in PET the tracers used produce positron-electron pairs, which in turn create high energy photon pairs [4].

During the development of conventional medical imaging techniques, computational methods of image reconstruction have been essential to image visualization. In recent years, medical imaging has moved beyond the role of visualizing anatomic structure and physiologic function, becoming a tool for surgical planning and simulation [9, 10] and intra-operative navigation [11, 12]. Doctors use these images to extract information critical to diagnosis and treatment by identifying, measuring, functionally assessing, and locating structures.

2.2 MEDICAL IMAGE ANALYSIS

To further assist doctors with the diagnosis and treatment of disease, the medical image analysis community has faced the challenge of extracting clinically useful information from images of various modalities. For example, advances in cardiac and brain imaging have lead to a deeper understanding of various pathologies and physiological anomalies. In cardiac imaging, abnormal cardiac contractions can be observed by tracking the cardiac surfaces through the cardiac cycle, thereby identifying valve disease or arrhythmia. A measure of left ventricular volume can be utilized to calculate cardiac function parameters such as stroke volume and ejection fraction. If myocardial volume and thickness can be measured and healthy valves are assumed, stroke volume and cardiac output can then be calculated to determine the health of heart tissue. Each of these measured quantities can be useful in the diagnoses of cardiovascular disease, including cardiomyopathy, ischemia, myocardial infarction, and congestive heart failure. In brain imaging, Alzheimer’s disease can be detected by measuring ventricular volume, subarachnoid cerebrospinal fluid space, or hippocampus size. Schizophrenia can be diagnosed by making measures of the corpus callosum and thalamus. Strokes, as well as the presence of brain tumors, can also be detected through the use of image analysis.

Unfortunately, image analysis as a means to assist doctors in extracting such information has evolved much slower than imaging technology itself. For instance, it is still generally not feasible to automatically analyze medical images with accuracy and efficiency even though the images provide a high resolution survey of internal anatomy.

The field of medical image analysis covers a large number of areas of interest such as segmentation, registration, classification, object recognition, and shape representation. In general, image analysis methods can be classified as *manual*, *semi-automated*, or *automated*.

Manual methods require the user to directly interact with the image to complete the task. Manual methods tend to be very subjective and dependent on inter- and intra-user variability. They tend to be time consuming and tedious, but accurate, assuming that the user is a reliable expert. For this reason, manual methods are often used to validate the accuracy of semi-automated or fully automated algorithms. With semi-automated methods,

the user can initially interact with the image to provide image information to the algorithm, or make modifications to the method during run-time based on feedback provided. Ideally, semi-automated methods are faster than manual methods, minimally affected by user variability and error, and provide reproducible results. Automated methods do not require user interaction, and instead utilize a training or learning procedure. While automated methods completely remove the user from the process, some are so computationally intensive that they may be actually slower than manual methods. Additionally, without the assistance of a user they can fail when dealing with the difficulties common to medical images such as image noise, weak boundaries, and structure complexity and variation.

This section begins with a brief overview of two common areas of image analysis: the *segmentation* and *registration* of structural, static images. This is by no means a complete survey of segmentation and registration in medical image analysis, but provided to illustrate the motivation for this research. This section concludes with a review of semi-automated and automated landmark extraction methods.

2.2.1 Medical Image Segmentation

The goal of medical image segmentation is to isolate anatomical structures of interest from the background and surrounding tissues. This process is typically governed by labeling voxels to indicate what each voxel corresponds to in the real world. For example consider an algorithm for segmenting the thalamus from an MRI brain image. Such an algorithm would label all voxels that correspond to the thalamus as belonging to the thalamus, and every other voxel as not belonging to the thalamus. Such a segmentation would thus produce an image with voxels labeled as either: thalamus or not thalamus. A wide variety of segmentation applications exist in medical research, such as volumetric visualization, quantitative measures of anatomical structures, construction of anatomical atlases, and tracking of anatomical change over time. Although this dissertation does not directly develop a segmentation method, it is still reviewed here because methods developed in this dissertation could potentially improve existing segmentation techniques.

As intuitive as object segmentation is to people, it translates to a very difficult problem and elaborate procedure for computers. Medical image segmentation is such a prevalent and difficult task that it has been an active area of research for several decades. Yet, a general segmentation procedure that is truly robust for all imaging modalities and anatomical targets is still a much desired goal. Some difficulties of image segmentation arise from the imaging process itself. During data acquisition and image reconstruction, boundaries between objects are often blurred and most supposedly homogenous regions are not truly homogeneous due to image noise. Although the human visual system is able to deal with these issues, even the most advanced algorithms often falter. The complexity and variability of anatomical structures also makes it difficult to segment objects without knowledge provided by an expert user or knowledge built into the system. Some common approaches can be categorized as follows: threshold-, histogram-, region-, edge-, contour-, model-, texture-, and hybrid-based approaches [13, 14]. Presented here are some common approaches to semi-automated and automated based methods. For a more thorough review of segmentation in medical imaging, refer to [15, 16]. Before delving into these segmentation approaches, manual segmentation is described.

Manual segmentation consists of either hand labeling voxels or tracing the boundary of the structure of interest. For clinical use, an expert with knowledge of the anatomy utilizes a mouse to perform the segmentation while looking at the computer screen. This is a relatively straightforward task with 2D images, but becomes problematic with 3D images because of the limits of effectively visualizing 3D images on a computer screen. Thus, the user is required to perform the segmentation on each 2D slice in the image volume, having to mentally reconstruct the structure of interest while viewing adjacent slices in the volume to correctly reconstruct the structure in 3D. As previously mentioned, manual segmentation is a time-consuming process, subject to inter- and intra-user variability. Consequently, it is generally avoided if a more autonomous method exists for the structure of interest. However, despite its shortcomings, manual segmentation is frequently used as a gold standard for validating newly developed segmentation algorithms.

A compromise between manual and automated methods are semi-automated methods. Semi-automated segmentation methods depend on user interaction to provide information to

the system. Among the simplest of these methods are threshold-based methods. The most primitive threshold-based method is to simply label voxels that are within a user-specified intensity range, which is usually determined by trial and error. More advanced threshold-based methods utilize the image’s intensity histogram to decide on a threshold that would best segment the object from the background. These conventional threshold methods use a single global threshold [17]. The success of these methods depends on histograms with nicely separable peaks. For images whose histograms have more than two modes, multiple thresholds can be decided upon. A method called adaptive thresholding utilizes one or more thresholds based on local image characteristics [18]. One such method, developed by Otsu, selects threshold levels to maximize the ratio of the between-class variance to the within-class variance [19].

Another semi-automated segmentation method based on voxel intensity is region growing. With region growing methods, a seed voxel is selected by the user to initialize the region within the structure of interest. Neighboring voxels that satisfy a homogeneity criteria are then added to the region until a convergence criteria is satisfied, such that the region fills the structure of interest [20]. Similarly, mathematical morphology involves dilation of regions in addition to erosion, opening, closing, etc [21, 22]. The most often used morphology operator for segmentation is the watershed operator, which has led to a method of segmentation called watershed segmentation [23]. In watershed segmentation, pixels are found at the midpoints between features by treating pixel values as topographic elevations. The areas between these features become catchment basins that segment objects.

Structures are ideally delineated by a closed boundary. Many segmentation methods first employ a scheme that identifies boundaries by measuring rapid changes in intensity. Since boundaries are typically discontinuous, gaps are filled in some manner, and thus the set of voxels contained by the boundary is considered the segmented structure. Structure boundaries may be detected by identifying areas of high intensity gradient magnitude. Classical gradient operators such as the Prewitt, Sobel, and Roberts Cross convolution kernels detect a boundary by taking an image derivative. These kernels deliver gradient magnitude, but also amplify high-frequency image noise. The Canny detector addresses this problem by first convolving the image with a Gaussian kernel, finding putative edges using the gradient

and connecting them to form edges. Another classical boundary detector is the Laplacian of Gaussian filter, which locates boundaries by taking the second image derivative and locating zero crossings [13, 18].

In addition to utilizing image intensity and intensity gradient for structure segmentation, global shape assumptions can be made. One such assumption is that anatomical structures will have smooth boundaries without high curvature. Methods that may enforce such constraints include snakes (active contours) [24], deformable models [25], and level sets [26, 27, 28]. These methods formulate the segmentation problem as an energy-minimization problem, where a curve evolves in the image until it reaches the lowest energy state, i.e., the curve is close to the structure boundary. These types of models subject the curve to external and internal forces which control its evolution. External forces are derived from image information, causing the curve to stop evolving, for example, when the curve reaches a high-gradient boundary. Internal forces control the shape of the curve causing it to, for example, have a speed proportional to its curvature, which would quickly smooth sharp curves while keeping low curvature regions the same.

Automated segmentation algorithms generally provide an anatomical model. Most semi-automated methods can be automated by providing an anatomical model rather than relying on user supplied information. Two examples of anatomical models are atlases and shape models. Atlases are often created by a manual or semi-automated segmentation method of delineating the structure of interest. Shape models may be derived from a training set of images with Principal Component Analysis (PCA) or other methods of dimensionality reduction. Shape models encapsulate more information than atlases because they capture the anatomic variation of the training set [29, 30, 31]. Originally, the actual construction of the shape model required manual identification of anatomical landmarks. Since the conception of shape models, there has been research into automating landmark extraction. A review of these methods is presented at the end of this chapter. Segmentation is usually performed by iteratively repositioning and transforming the model in the image to minimize a distance or energy function such that the model best fits the underlying image data. This transformation process is the subject of medical image registration, which is discussed in the following section.

2.2.2 Medical Image Registration

Medical image registration is a correspondence problem with the goal of spatially aligning two images such that correspondences are established between image voxels. Registration is a necessary process for comparing or integrating information from multiple images. Image registration has many uses in the medical field such as 1) combining information from multiple imaging modalities, 2) relating individual patient anatomy to standardized atlases, 3) monitoring changes in structure size, shape, or intensity, and 4) relating pre-operative images and surgical plans to the patient during image guided surgery. The remainder of this section briefly reviews three basic approaches of intrinsic registration, that is, methods that rely only on patient data. For a more thorough review on medical image registration, the reader is referred to [16, 32, 33].

The three approaches of registration reviewed here are based on aligning segmented structures, voxels, and landmarks; these are called, respectively, *segmentation-*, *voxel-*, and *landmark-based* registration.

Segmentation-based registration methods either rely on a rigid or deformable model. With a rigid model based method, structure segmentations from both images are used to align the images via a global rigid transform [34, 35, 36, 37, 38]. The registration accuracy of segmentation-based methods is typically limited to the accuracy of the provided segmentations. As previously discussed, the difficulty of the segmentation task depends on the structure of interest, and a wide array of segmentation methods are available. Rigid models are typically used for intra-subject registration and deformable approaches are necessary for inter-subject and atlas registration. With deformable models, an extracted structure from one image is deformed to fit the second image. The extracted structure is commonly represented as snakes or active contours. Similar to the contour-based methods discussed for segmentation, the deformable model is fit while minimizing an energy function [39, 40, 25]. Often a template generated from an atlas image is registered to a segmented or unsegmented image with global rigid or affine transformations. If there are local deformations in the template that drastically differ from the target structure, the registration will typically fail. To address such problematic issues, local transformations are permitted.

Voxel-based registration methods operate directly on the intensity values of the image voxels. There are two distinct approaches to voxel-based registration methods. One approach immediately reduces the number of voxels to be considered. These methods compute the image center of gravity and principal axes from the zeroth and first order moments. Registration is then performed by aligning the center of gravity and principal axes. Researchers have attempted to improve results by computing higher order moments with some success. Resulting registrations are not usually very accurate, but are automated and very fast to compute, which is well-suited for applications such as image re-alignment and pre-registration [41, 42, 43, 44]. The second approach utilizes the entire image. Since all voxels are considered throughout the registration process, the method is very flexible. Computational costs are considerable, but manageable with modern computer technology. The goal of this approach is to optimize transform parameters while minimizing a cost function or maximizing a reward function that is based on voxel properties. There are many cost functions and reward functions available such as those that 1) minimize variance of intensity ratios [45], variance of intensity values within segments [46, 47], histogram entropy difference [48], histogram dispersion [49, 45], or squared intensity differences [50, 51, 44], or 2) maximize cross-correlation [52, 42, 53], mutual information [54, 55], zero crossings in difference images [56, 57].

Landmark-based registration methods attempt to identify a global transform that maximizes the mapping of landmarks to each other. The transform can then be applied to an image to spatially align the voxels between images. Landmarks can be anatomical [58, 59, 60, 61] or geometrical [62, 63, 64, 65, 66]. Classically, anatomical landmarks are salient and reliably locatable points on anatomical structures, i.e., the tip of the nose, that are manually identified by the user. Geometrical landmarks are typically features extracted from the image such as boundary points, local curvature, corners, and edges. The optimum transform may be determined by minimizing a distance measure such as the Procrustes metric, which is defined as the least-squares distance between each landmark and its corresponding landmark. Standard optimization methods such as Powell’s method, Levenberg-Marquardt, gradient descent, and the downhill simplex method are often utilized to minimize the distance measure. Non-standard optimizers such as the Iterative Closest Point (ICP)

algorithm have also been developed for this particular problem. The ICP algorithm is popular because it is computationally fast and can be used for curves, surfaces, and volumes in addition to specifically corresponding points [67]. Since the set of landmarks is sparse compared to the number of image voxels, landmark-based registration is typically very fast and efficient. Transforms are typically limited to rigid or affine unless a large set of landmarks is available. A disadvantage of these methods is that extracting landmarks is typically a manual, time-consuming process. Consequently, there has been investigation into partially and fully automating geometric landmark estimation methods.

A majority of landmark estimation methods operate by detecting specific features (geometric landmarks) from the images then determining correspondences between them. One type of landmark can be detected with first-, second-, or third-order differential operators to locate regions of high intensity gradient [68, 69, 70]. Pielot, *et. al.*, determine correspondences between landmarks detected with a first-order differential operator by pre-aligning and gridding the images, then looking for maximum gradient in each grid [71]. Another type of landmark can be detected by locating points on a contour of high curvature (critical points) with an angle or corner detection scheme [72, 73, 74]. Correspondences between extracted landmarks can be determined by any of the methods previously mentioned. Hill *et. al.*, have developed a method of landmark estimation without requiring landmarks to be extracted from both reference and query images. Their method relies on first generating a shape model from a set of training images. Landmarks are chosen to best approximate the mean shape as the vertices of a sparse polygon. Landmarks can then be projected into each of the images in the training set from the shape model [75].

Each of these approaches requires prior knowledge about the type of landmarks to be detected. Geometric landmarks may be useful, but do not necessarily relate to manually defined landmarks used by a clinician. In this dissertation landmarks are not restricted to underlying image features. The landmarks can be placed anywhere in the image that a user desires. The correspondence problem and landmark estimation problem are separate such that once correspondences between image features are established landmarks can be inferred.

Ultimately, the goal of image registration is the determination of correspondences between inter-image voxels for comparing the underlying anatomy between images. This dis-

sertation addresses the correspondence problem, not for individual voxels, but for image features. Grimson proposed that correspondences between image features can be found based on geometric constraints [76]. The correspondence method developed in this dissertation utilizes geometric constraints between *cliques* containing image features to automatically identify inter-image correspondences. Analogous to landmark-based registration, the correspondence method developed in this dissertation attempts to identify correspondences between image features (the term landmark is reserved for anatomical landmarks in the remainder of this dissertation). However, rather than utilizing feature correspondences to register image voxels, the locations of anatomical landmarks are inferred from feature correspondences.

2.3 MEDIAL-BASED IMAGE ANALYSIS

There are many different approaches to image analysis. A bottom-up approach based on the medial properties of shapes has been chosen as the foundation of the proposed research. The medial axis, as defined by Blum for 2D binary images, is the locus of centers of circles that are at least bi-tangent to an object boundary and fit entirely within the object [77] (Fig. 2.1). Similarly in 3D, the centers of medial spheres define a *medial manifold* that is at most 2D. Pizer *et al.* developed a measure called *medialness* to extend the medial manifold to three-dimensional gray-level images [78, 79]. Medialness links the aperture of a boundary measurement to the radius of the medial manifold. The ridge of medialness is called a core. Methods using medialness have proven robust against image noise and shape variation [80, 81, 82, 83, 84, 85, 86, 87, 88]. These methods include shocks [89, 90, 91, 92], skeletons [93, 94, 95], m-reps [85, 96], and core atoms [80, 84, 86]. *Core atoms* have been chosen for the features to be used in this dissertation because they are a bottom-up approach that is computationally efficient and have proven successful in identifying (with the use of a pre-determined model) the left ventricle in RT3D ultrasound.

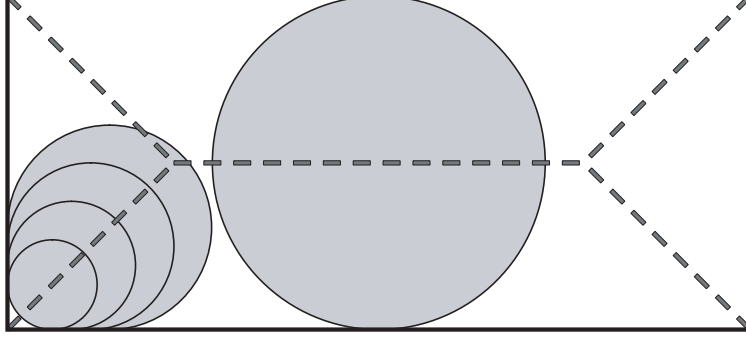


Figure 2.1: A homogeneous rectangle on a uniform background. Circles are bi-tangent and the dashed line indicates the medial axis of a rectangle as defined by Blum.

2.3.1 Core Atoms

Core atoms are formed from a collection of boundary points that consist of a local measurement of gradient magnitude and orientation, such as those delivered by a *Difference of Offset Gaussians* (DoOG) gradient detector [97, 98]. A core atom is constructed from a pair of boundary points if the following two requirements are satisfied:

1. The distance between the boundary points \mathbf{b}_1 and \mathbf{b}_2 is within a specified range. This constraint permits computational savings if there is *a priori* information about the expected width of the structure. This distance is termed the *scale* of the resulting core atom and is defined as the L2-Norm of the vector $\mathbf{s}_{1,2}$ that extends from \mathbf{b}_1 to \mathbf{b}_2 as

$$\|\mathbf{s}_{1,2}\| = \|\mathbf{b}_2 - \mathbf{b}_1\|, \quad (2.1)$$

where $s_{min} \leq \|\mathbf{s}_{1,2}\| \leq s_{max}$ and the expected width is between s_{min} and s_{max} . The vector $\mathbf{s}_{1,2}$ indicates the direction from \mathbf{b}_1 to \mathbf{b}_2 . The core atom is defined to be located at the midpoint between the two boundary points.

2. There is sufficient *face-to-faceness*, i.e., the boundary point gradients face each other. Face-to-faceness is defined as

$$F(\mathbf{b}_1, \mathbf{b}_2) = \left| \left(\frac{\mathbf{s}_{1,2}}{\|\mathbf{s}_{1,2}\|} \cdot \mathbf{n}_1 \right) \left(\frac{\mathbf{s}_{2,1}}{\|\mathbf{s}_{2,1}\|} \cdot \mathbf{n}_2 \right) \right|, \quad (2.2)$$

where \mathbf{n}_i for $i = 1, 2$ is the normalized gradient of the \mathbf{b}_i boundary point. There is sufficient face-to-faceness if $F(\mathbf{b}_1, \mathbf{b}_2)$ is close to 1, which allows some variation in the orientation of the boundary points relative to $s_{1,2}$.

2.3.2 Medial Nodes

A local population of core atoms whose centers fall within a certain region can be analyzed in terms of their orientations. For 3D data, there are three basic core atom configurations; “koosh-ball,” “spokes-of-a-wheel,” and “bed-of-nails.” The “koosh-ball” configuration occurs when core atoms are formed within a spherical surface. The core atom vectors form the koosh spikes and the centers of the core atoms cluster near the center of the sphere. Core atoms formed on a cylinder result in a “spokes-of-a-wheel” arrangement, with core atoms located along the cylindrical axis. Core atoms that form within the surfaces of a slab will yield a “bed-of-nails” configuration across the interior of the slab with core atom centers clustered on a 2D plane between the surfaces of the slab. Each of these basic core atom configurations are shown in Fig. 2.2.

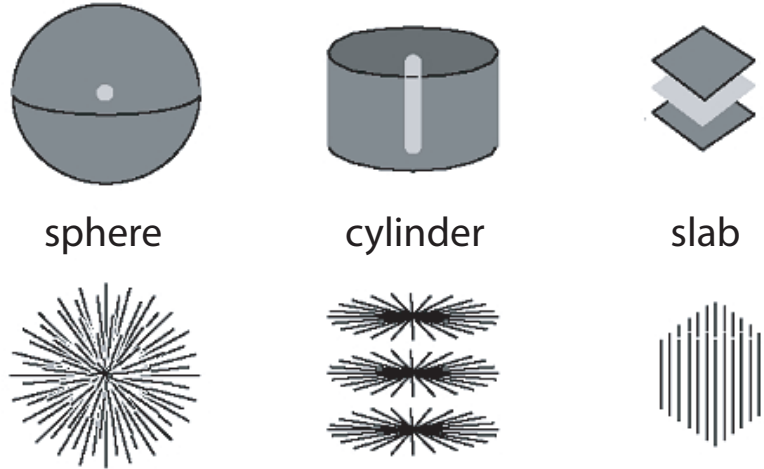


Figure 2.2: Top shows basic shapes in dark gray with corresponding core in light gray. Below these are shown the corresponding basic core atom configurations (from left to right): “koosh-ball,” “spokes-of-a-wheel,” and “bed-of-nails” (with core atoms shown as line segments).

For a given population of core atoms, their orientation vectors can be statistically processed using eigenanalysis of the covariance matrix, to yield a measure of *dimensionality*

and an overall estimate of *orientation*. The resulting eigenvalues $\lambda_1 < \lambda_2 < \lambda_3$ define the local dimensionality of the core and the corresponding eigenvectors $\hat{\mathbf{a}}_1$, $\hat{\mathbf{a}}_2$, $\hat{\mathbf{a}}_3$ define a coordinate system specific to the orientation of the core atom population. The eigenvectors are ordered such that $\hat{\mathbf{a}}_1$ is most orthogonal to the population of core atom vectors and $\hat{\mathbf{a}}_3$ is least orthogonal to the core atoms, i.e., most aligned with the core atoms. The eigenvalues are greater than or equal to zero and sum to 1. A value of 0 indicates that the corresponding eigenvector is completely orthogonal to every core atom in the population. This is true for $\hat{\mathbf{a}}_1$ in a cylinder and $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{a}}_2$ in a slab. For a sphere, no eigenvectors are orthogonal to every core atom since the core is a point. Since the eigenvalues sum to 1, they can be viewed as a system with only 2 independent variables defining a triangular domain called the lambda triangle (Fig. 2.3).

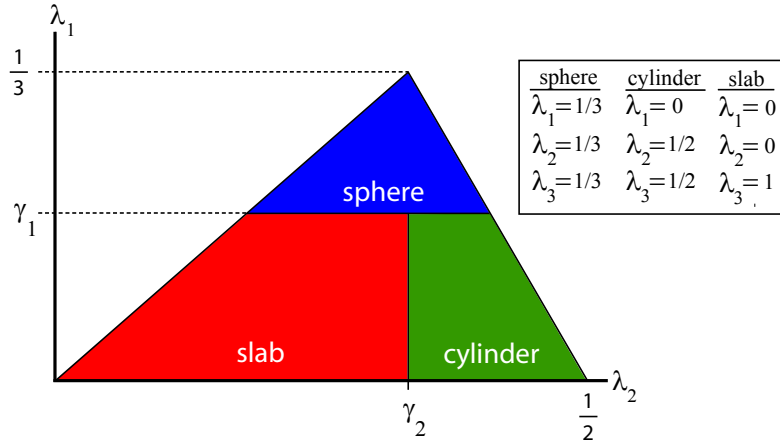


Figure 2.3: The lambda triangle. The eigenvalues in the box describe the perfect sphere, cylinder, and slab, i.e., the vertices of the lambda triangle.

The lambda triangle’s vertices represent the three basic medial shapes in 3D (sphere, cylinder, and slab). All possible sets of eigenvalues are bound by the triangle. Arbitrary thresholds, γ_1 and γ_2 , placed on λ_1 and λ_2 divide the triangle into 3 compartments, which designate dimensionality for visualization purposes. Sampling core atoms on a regular grid is convenient, but medial properties measured in this manner are prone to displacement from the true core, a result of the face-to-faceness threshold and sampling artifact. The

displacement from the true core may cause a misclassification of local dimensionality for core atom populations that are under-sampled. Clustering neighboring samples of core atoms overcomes this displacement problem, using methods described by Stetten [80]. A clustered core atom population is analyzed as a single *medial node*. The location of a medial node in image space is the center of mass of the corresponding core atom population, and the scale of the medial node, σ is the average scale of that population.

3.0 AUTOMATED FEATURE-BASED CORRESPONDENCES

In this chapter, the method for automatically determining correspondences between medial nodes (often referred to simply as *nodes*) is detailed. Node correspondences are determined by first determining correspondences between *cliques* consisting of medial nodes. Three different types of cliques are considered: singlets, doublets, and triplets. *Singlets* consist of one node, *doublets* of two nodes, and *triplets* of three nodes. Determining correspondences between each type of clique are governed by a unique correspondence metric. The metrics for evaluating correspondences between singlets, doublets, and triplets are the *unary*, *binary*, and *trinary* metrics, respectively.

The unary metric identifies corresponding nodes by their similarity in dimensionality and scale. Doublets and triplets incorporate geometric relationships between the constituent nodes, and correspondences are determined from metrics invariant to global rotation and translation. Identifying correspondences relies on using these three metrics in succession. Although medial nodes are employed in this dissertation, the correspondence method developed here provides a general framework for determining correspondences between cliques containing any feature. The framework, based on identifying similar cliques relies on a hierarchy of similarity metrics inherent to feature properties and geometrical relationship between clique constituents. Other features can be utilized by reformulating the similarity metrics to incorporate information delivered by the type of feature. For example, consider boundary points calculated with a gradient-based convolution kernel. The unary metric could be formulated to evaluate for similarity in intensity and gradient magnitude. The relative orientation between boundary points in a doublet can be calculated from the gradient vectors.

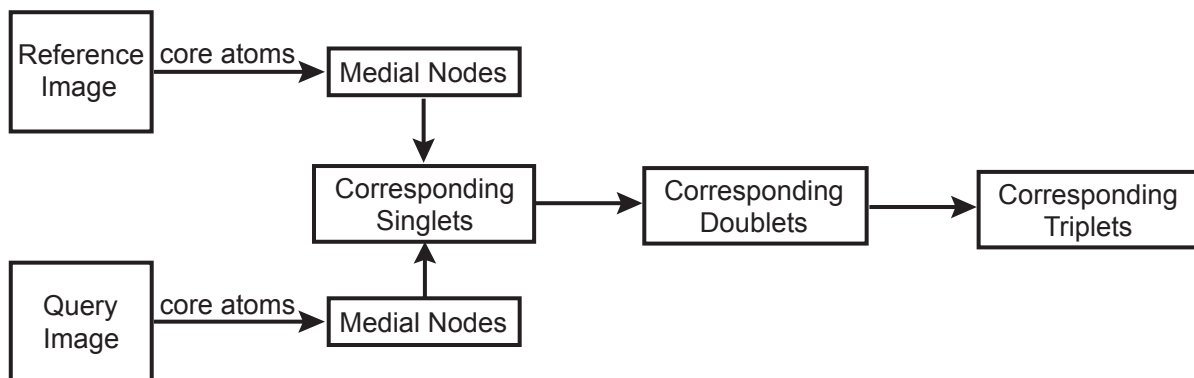


Figure 3.1: Framework for identifying correspondences between cliques of medial nodes.

The framework for identifying correspondences with this metric hierarchy is shown in Fig. 3.1. After each step in the metric hierarchy, correspondences become more accurate, as will be shown with Receiver Operator Characteristic (ROC) curve analysis at the end of this chapter.

The chapter concludes with several experiments that demonstrate the correspondence framework and ROC curve analysis.

3.1 PRELIMINARIES

Presented here is the notation used to derive the methodology of the correspondence framework. The reference image is denoted as R and the query image is denoted as Q . Let Ψ_α be the set of all medial nodes extracted from an N -dimensional image denoted as α , e.g., Ψ_R is the set of nodes extracted from image R . The number of nodes extracted from image α is denoted as M_α , i.e. $M_\alpha = |\Psi_\alpha|$.

The i^{th} medial node in Ψ_α is denoted as $n_{\alpha,i}$. The i^{th} clique from image α is generally represented by its set of nodes $C_{\alpha,i}$ such that $|C_{\alpha,i}| = k$, where k is the number of nodes in the clique. For example, consider the i^{th} doublet in image R consisting of nodes $n_{R,a}$ and $n_{R,b}$. This doublet is denoted as $C_{R,i} = \{n_{R,a}, n_{R,b}\}$ with $k = 2$. When describing any clique in image α , the subscript is omitted and the set of nodes is simply written C_R .

The unary, binary, trinary, and triplet transform metrics are respectively denoted as μ , β , τ , and χ . A correspondence matrix constructed via metric m is denoted as \mathbf{C}_m , where $m = \mu, \beta, \tau$, or χ . The correspondence value calculated for nodes $n_{R,i}$ and $n_{Q,j}$ is accessed from the correspondence matrix by $\mathbf{C}_m(n_{R,i}, n_{Q,j})$ or simply $\mathbf{C}_m(i, j)$. Each correspondence metric takes on values in the continuous interval $[0, 1]$, with 0 signifying perfect correspondence. The correspondence methods may be applied to any $N > 1$, but results are presented for only 3D images in this dissertation.

3.2 UNARY CORRESPONDENCES

Identifying node correspondences begins with the *unary metric* by identifying similar singlets, i.e., cliques that each consist of one node. Given that core atoms have been constructed and clustered into medial nodes, the unary metric is calculated between each singlet (node) in Ψ_R and each singlet in Ψ_Q . Note that since singlets consist of only one node, singlets are sometimes referred to as nodes within this section. The unary metric quantifies the similarity between a pair of singlets by their medial dimensionality and scale with a value of 0 indicating perfect correspondence. For example, the unary metric evaluated for two singlets consisting

of spherical (dimensionality) nodes of similar diameter (scale) yields a value close to 0 since they are nearly identical. Recall that each medial node has $N - 1$ independent eigenvalues λ_i for $i = 1 \dots N - 1$ that describe dimensionality, and an average diameter of the core atom population that defines its scale σ .

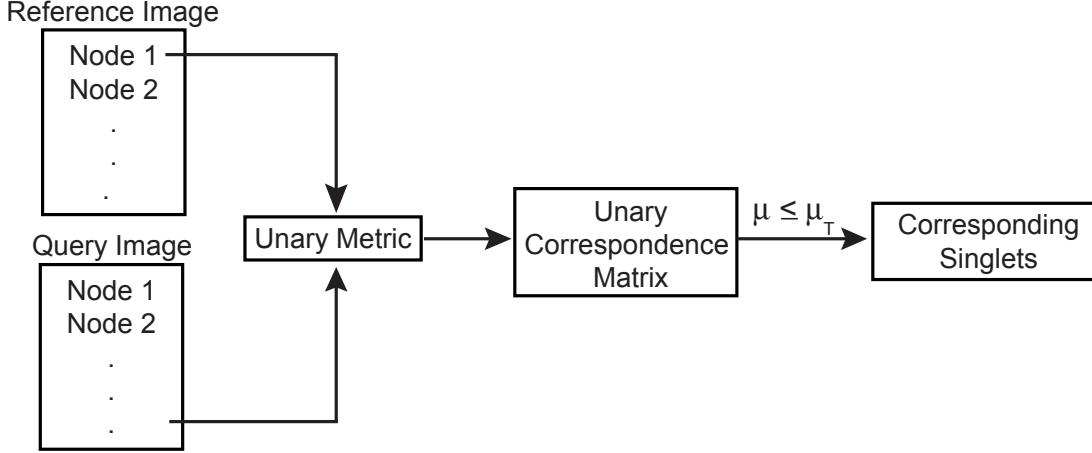


Figure 3.2: Framework for identifying singlet correspondences with the unary metric.

The steps for identifying corresponding singlets with the unary metric are shown in Fig. 3.2. The unary metric is calculated as the sum of a normalized dimensionality and normalized scale term. For node j in image R, $n_{R,j}$, and node k in image Q, $n_{Q,k}$, the unary metric $\mu(j, k)$ is defined as

$$\mu(j, k) = a_\mu \omega \sum_{i=1}^{N-1} (\lambda_{j,i} - \lambda_{k,i})^2 + b_\mu \left| \frac{\sigma_j - \sigma_k}{\sigma_j + \sigma_k} \right|, \quad (3.1)$$

where $\lambda_{j,i}$ and $\lambda_{k,i}$ are the i^{th} eigenvalues for nodes $n_{R,j}$ and $n_{Q,k}$, respectively, and σ_j and σ_k are the respective scales of $n_{R,j}$ and $n_{Q,k}$. Parameters a_μ and b_μ can be used in applications where it may be beneficial to weight the dimensionality and scale terms differently. To enforce $0 \leq \mu(j, k) \leq 1$, these weight parameters must satisfy $a_\mu + b_\mu = 1$. The normalization constant ω is necessary for keeping $\mu(j, k)$ distributed between 0 and 1.

Recall that for $N = 3$, the lambda triangle is bound by $\lambda_1 = \frac{1}{3}$ and $\lambda_2 = \frac{1}{2}$. Thus, the maximum value for the dimensionality term $\sum_{i=1}^{N-1} (\lambda_{j,i} - \lambda_{k,i})^2$ occurs when the eigenvalues for

node j are $\lambda_{j,1} = \frac{1}{3}$ and $\lambda_{j,2} = \frac{1}{2}$ and the eigenvalues for node k are 0 (or vice versa), i.e.,

$$\sum_{i=1}^{N-1} (\lambda_{j,i} - \lambda_{k,i})^2 = \left(\frac{1}{3} - 0\right)^2 + \left(\frac{1}{2} - 0\right)^2 \approx 0.36. \quad (3.2)$$

Therefore, ω is calculated as

$$\omega = \frac{1}{\sum_{i=1}^{N-1} (\lambda_{j,i} - \lambda_{k,i})^2} = \frac{1}{0.36} \approx 2.77. \quad (3.3)$$

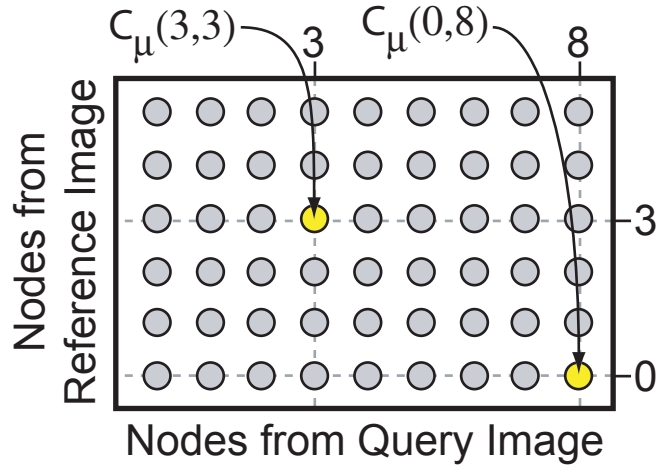


Figure 3.3: Unary correspondence matrix for organizing the unary correspondences between the 6 nodes from the reference image and 9 nodes from the query image. Origin (0,0) is the lower-left corner of the matrix. Unary correspondence values for nodes indexed at (3,3) and (0,8) are indicated as $\mathbf{C}_\mu(3,3)$ and $\mathbf{C}_\mu(0,8)$, respectively.

The unary metric evaluated between all nodes in Ψ_R and Ψ_Q are organized in a *unary correspondence matrix*, denoted as \mathbf{C}_μ . The unary correspondence matrix is $M_R \times M_Q$ in size, where M_R and M_Q are the numbers of medial nodes in Ψ_R and Ψ_Q , respectively. The matrix is organized such that nodes from the reference image are along the matrix rows and nodes from the query image are along the matrix columns. The unary correspondence matrix therefore stores the correspondence value calculated by the unary metric for every node in Ψ_R and Ψ_Q . The unary correspondence value between $n_{R,j}$ and node $n_{Q,k}$ from the query image can be retrieved by indexing \mathbf{C}_μ by (j,k) . An example \mathbf{C}_μ is illustrated in Fig. 3.3. From the diagram, it can be deduced that 6 nodes were extracted from the reference

image and 9 nodes were extracted from the query image. The unary metric evaluated for nodes $n_{R,3}$ and $n_{Q,3}$ is indicated as $\mathbf{C}_\mu(3,3)$, and the unary metric evaluated for $n_{R,0}$ and $n_{Q,8}$ is indicated as $\mathbf{C}_\mu(0,8)$.

If the order and number of nodes in Ψ_R and Ψ_Q are the same, the unary correspondence matrix, as well as other correspondence matrices, is symmetric about the diagonal, i.e., $\mathbf{C}_\mu(j,k) = \mathbf{C}_\mu(k,j)$ since $\mu(j,k) = \mu(k,j)$. Two medial nodes are said to correspond if their unary correspondence value is within a pre-determined threshold μ_T , i.e., nodes j and k correspond if $\mathbf{C}_\mu(j,k) \leq \mu_T$. The threshold on μ permits for flexibility in determining correspondences and offsetting computational round-off error.

Since the unary metric determines correspondences only by dimensionality and scale, it is expected that there will be a large number of false correspondences. Correspondences could be improved by incorporating the orientation of the nodes into the metric. However, the orientation of a node is within the reference frame of an image's coordinate system, and since it is not guaranteed that images will have the same orientation, an absolute coordinate system would need to be established. As stated earlier, accurately aligning images without first establishing correspondence is a difficult problem, so it is common in practice to manually align images during pre-processing, but this would no longer be totally automatic. Instead of taking this manual approach, orientation is incorporated into the correspondence framework by means of the binary metric. Within the correspondence framework, no assumptions of image coordinate systems are made, allowing for the development of metrics that are invariant to image orientation.

3.3 BINARY CORRESPONDENCES

A *doublet* is a clique that consists of two medial nodes from the same image. Doublets are restricted to only include nodes that have a correspondence in the other image as determined by thresholding the evaluated unary metric. Correspondences between doublets are determined with the *binary metric*, which is the second metric in the hierarchy of similarity metrics, as illustrated in Fig. 3.1. Since doublets are formed from nodes with unary cor-

respondences, the binary metric inherently encapsulates similarities in dimensionality and scale. More notably, the binary metric has been formulated to identify similar doublets based on particular geometric properties, namely orientation and distance between the two nodes in the doublet. *Doublet orientation* describes the relationship between a node's local coordinate system in relation to the other node's coordinate system by producing a unique *doublet signature*. For each doublet, two such doublet signatures are produced, one from the perspective of each node. *Doublet distance* is calculated as the Euclidean distance between the constituent of nodes in the doublet. Shown in 3.4 is the process for determining doublet correspondences.

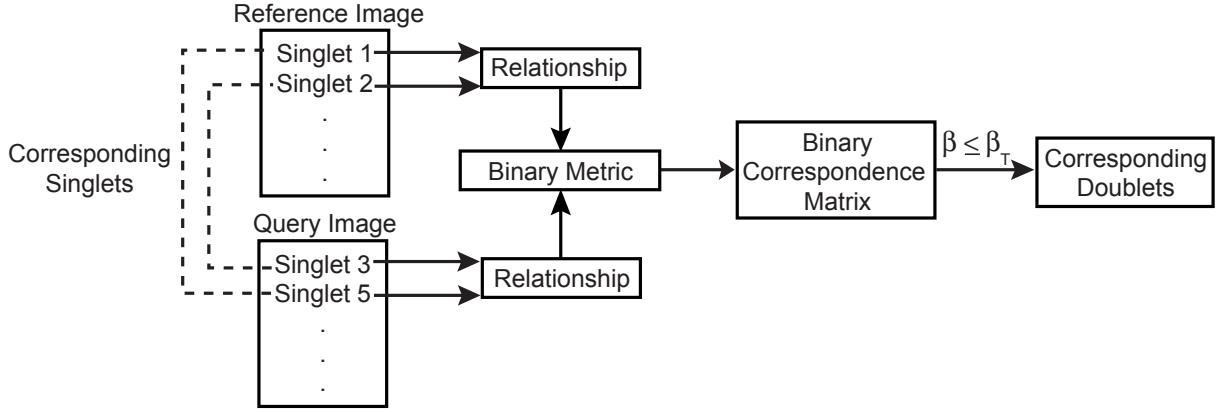


Figure 3.4: Identifying corresponding doublets with the binary metric.

3.3.1 The Doublet Signature

Consider the i^{th} doublet in image α consisting of two nodes $n_{\alpha,j}$ and $n_{\alpha,k}$. This doublet is denoted as $C_{\alpha,i} = \{n_{R,j}, n_{R,k}\}$. For the purpose of defining the doublet signature, notation will be simplified such that $n_{\alpha,j}$ and $n_{\alpha,k}$ will be referred to as node 1 and node 2, respectively. The doublet signature is now derived from the perspective of node 1. The signature from the perspective of node 2 can be computed with the same equations by interchanging the node 1 variables for node 2 variables. The displacement vector from node 1 with image coordinate \mathbf{p}_1 to node 2 with image coordinate \mathbf{p}_2 is defined as

$$\mathbf{v} = \mathbf{p}_2 - \mathbf{p}_1. \quad (3.4)$$

Given the eigenvectors $\hat{\mathbf{a}}_i$ for $i = 1, 2, \dots, N$ of node 1, the relative location of node 2 can be expressed in the coordinate system of node 1 as

$$d_i = \sum_{j=1}^N v_j (\hat{\mathbf{a}}_i)_j \text{ for } i = 1, 2, \dots, N, \quad (3.5)$$

where $(\hat{\mathbf{a}}_i)_j$ is the j^{th} component of the i^{th} eigenvector. It is further defined that $\hat{\mathbf{d}}$ is the normalized vector

$$\hat{\mathbf{d}} = \frac{\mathbf{d}}{\|\mathbf{d}\|}, \quad (3.6)$$

which expresses the direction to node 2 from node 1 in the coordinate system defined by the eigenvectors of node 1. The square of the normalized component $(\hat{d}_i)^2$ is referred to as the i^{th} *Normalized Component Squared* (NCS) of \mathbf{d} .

The signature for node 1 is defined as the cumulative NCS for all i as a function of λ . Each component of the NCS along a given eigenvector $\hat{\mathbf{a}}_i$ is added at its corresponding eigenvalue,

$$s(\lambda) = \sum_{i=1}^N (\hat{d}_i)^2 u(\lambda), \quad (3.7)$$

where $u(\lambda)$ is the unit step function. Since eigenvalues are always positive, $s(\lambda) = 0$ for $\lambda < 0$. In effect, $s(\lambda)$ is a series of steps. Each step is located at an eigenvalue with a height equal to the corresponding NCS.

A metric to express the similarity between doublets should ignore symmetry in node 1 in terms of the location of node 2. Such symmetry results only when two of node 1's eigenvalues are approximately equal. If two or more eigenvalues are equal, their steps superimpose into a single NCS in the degenerate subspace defined by the corresponding eigenvectors. By the Pythagorean Theorem, the total step size in such a case is independent of arbitrary rotation within the degenerate subspace and the total of all the steps is 1. A proof is provided in Appendix A to show that the signature is invariant to symmetry in the degenerate subspace of a node.

Typical signatures, from the perspective of node 1, are shown in Fig. 3.5 for doublets with an almost perfectly cylindrical node 1 (recall that $\lambda_1 = 0$, $\lambda_2 = 0.5$, and $\lambda_3 = 0.5$ for a perfectly cylindrical node). Such signatures could hypothetically have been computed

from the doublet illustrated in Fig. 3.6, which shows a cylindrical node 1 paired with a cylindrical node 2 at the “equator” of node 1’s coordinate system. Comparing the two signatures for the doublet in Fig. 3.5 there is an arbitrary rotation of node 1’s eigenvectors within the degenerate subspace. Referring back to Fig. 3.6, this arbitrary rotation can be conceptualized as rotation around the $\hat{\mathbf{a}}_1$ axis of the node 1 cylinder. The doublets have similar signatures, which only vary along a short segment of the λ -axis. The first NCS for these signatures is 0 since its corresponding eigenvector is orthogonal to \mathbf{v} .

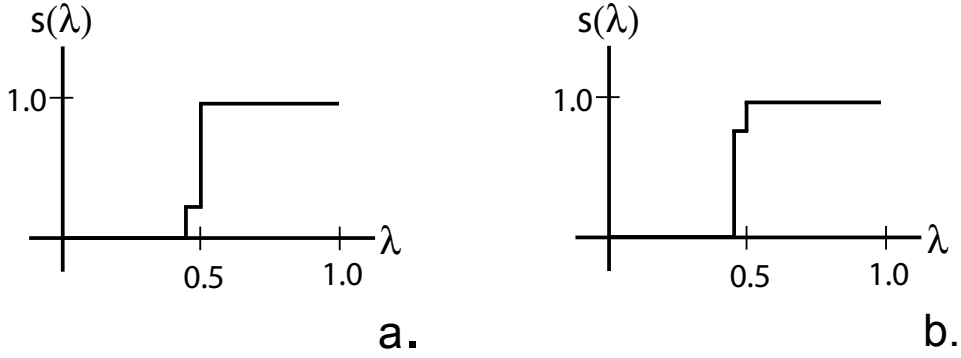


Figure 3.5: a) Signature of a cylindrical node 1 paired with node 2 at the “equator” of node 1’s coordinate system. b) Another doublet with an “equatorial” node 2, but a different orientation of node 1’s eigenvectors within the degenerate subspace.

In Figure 3.7, signatures from the perspective of node 1 are shown with the same cylindrical node 1 as before and a cylindrical node 2 located not at the equator, but at 45° “latitude” in node 1’s coordinate system. The signature is shifted above the λ -axis at a value of $\mathbf{v} \cdot \hat{\mathbf{a}}_1$ since the first eigenvector is no longer orthogonal to \mathbf{v} . The minor difference in step sizes at $\lambda = \frac{1}{2}$ is still a result of rotation within node 1’s degenerate subspace, which can be conceptualized as rotation around the $\hat{\mathbf{a}}_1$ axis of the node 1 cylinder in Fig. 3.8.

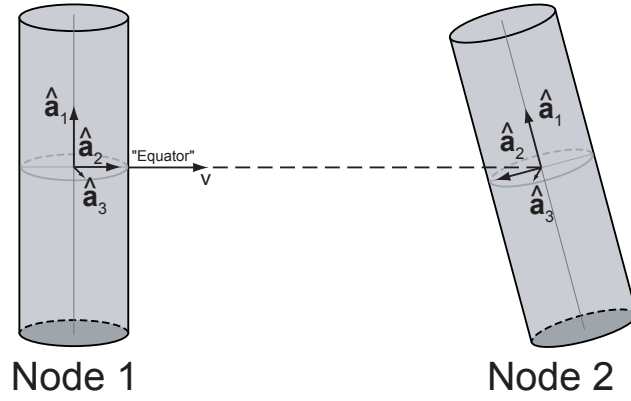


Figure 3.6: Cylindrical node 1 paired with cylindrical node 2 at the “equator” of node 1’s coordinate system. Rotation within node 1’s degenerate subspace can be conceptualized as rotation around the $\hat{\mathbf{a}}_1$ axis of node 1.

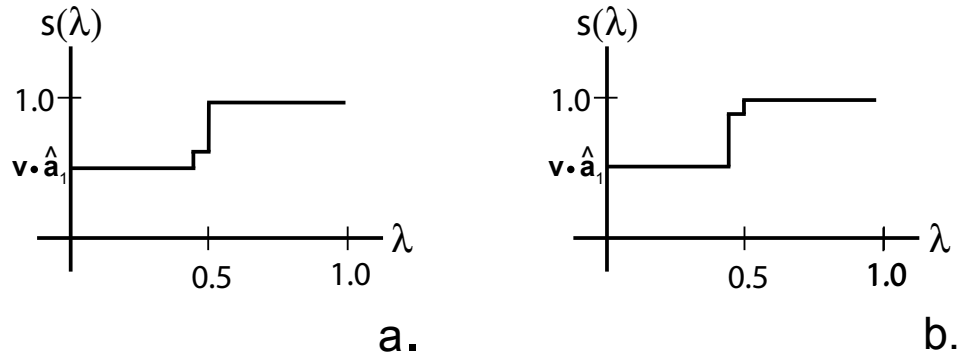


Figure 3.7: **a)** Signature (from node 1’s point of view) of cylindrical node 1 paired with node 2 at 45° “latitude” of node 1’s coordinate system, as shown in Fig. 3.8. **b)** Another doublet with node 2 at the 45° “latitude” of node 1’s coordinate system, but with an arbitrary rotation of node 1’s eigenvectors within the degenerate subspace.

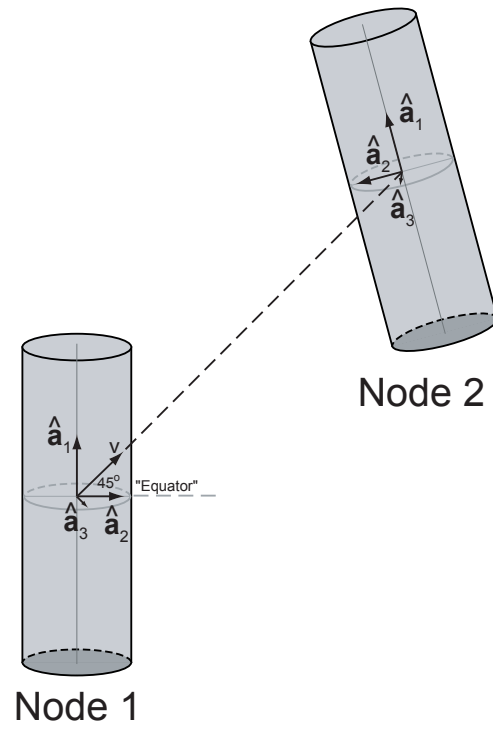


Figure 3.8: Cylindrical node 1 paired with cylindrical node 2 at 45° “latitude” of node 1’s coordinate system.

3.3.2 The Binary Metric

The binary metric operates by comparing doublet signatures. Within an image, two signatures are computed for each doublet, one signature from the perspective of each node. For these four signatures, only two signature comparisons are necessary since node correspondences within the doublet are known from the unary metric. As part of the binary metric, signature metrics are calculated for signatures between doublets as the integral of the absolute signature differences, i.e., signatures from images R and Q are subtracted, and the average absolute difference is calculated. For identical signatures, the signature metric evaluates to 0.

Consider a doublet in image R consisting of nodes $n_{R,1}$ and $n_{R,2}$. The signature of this doublet from the perspective of node $n_{R,1}$ is denoted as $s(\lambda)_{R,1,2}$ and the signature of this doublet from the perspective of $n_{R,2}$ is denoted as $s(\lambda)_{R,2,1}$. Following the same notation for image Q, a doublet in image Q consists of nodes $n_{Q,1}$ and $n_{Q,2}$ with signatures $s(\lambda)_{Q,1,2}$ and $s(\lambda)_{Q,2,1}$. It is known that according to the unary metric $n_{R,1}$ and $n_{Q,1}$ correspond, and $n_{R,2}$ and $n_{Q,2}$ correspond. Therefore, the signature metric evaluated for signatures from the perspective of node 1 to node 2 is defined as

$$S_{1,2} = \int_0^1 |s(\lambda)_{R,1,2} - s(\lambda)_{Q,1,2}| d\lambda \quad (3.8)$$

and the signature metric evaluated for signatures from the perspective of node 2 to node 1 is defined as

$$S_{2,1} = \int_0^1 |s(\lambda)_{R,2,1} - s(\lambda)_{Q,2,1}| d\lambda, \quad (3.9)$$

where $0 \leq S \leq 1$ and a value of 0 indicates identical signatures. In the special case that singlet correspondences are ambiguous, the signature metrics must be also calculated between $s(\lambda)_{R,1,2}$ and $s(\lambda)_{Q,2,1}$, and between $s(\lambda)_{R,2,1}$ and $s(\lambda)_{Q,1,2}$. Shown in Fig. 3.9 are two superimposed signatures illustrating the signature metric from node 1's perspective. The gray region is the difference in signatures, the area of which is equal to $S_{1,2}$.

The integrals in Equations 3.8 and 3.9 can be computed efficiently as the area of a series of rectangles by sorting the union of the two sets of eigenvalues from least to greatest

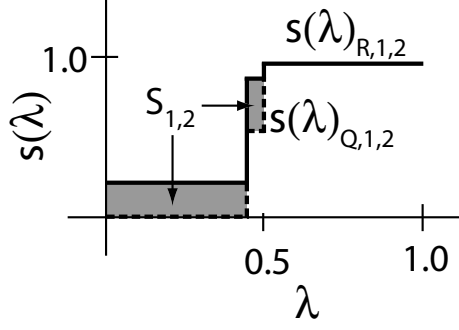


Figure 3.9: Two superimposed signatures illustrating the signature metric from node 1's perspective. A signature from image R, $s(\lambda)_{R,1,2}$, is shown as a solid line with a signature from image Q, $s(\lambda)_{Q,1,2}$, as a dashed line. The gray region is their difference, the area of which is equal to $S(1, 2)$.

and assigning a new index from $0 \leq k < 2N$ to each eigenvalue. The signature metric is calculated from the set of $2N$ eigenvalues by

$$S = \sum_{k=1}^{2N} \kappa(\hat{\mathbf{d}}_k)^2 (\lambda_k - \lambda_{k-1}), \quad (3.10)$$

where $\kappa = 1$ for eigenvalues from node 1 and $\kappa = -1$ for eigenvalues from node 2, and vice versa for $S(2, 1)$. The orientation term of the binary metric is calculated from the signature metric as the average of the signature metrics,

$$\beta_o = \frac{S_{1,2} + S_{2,1}}{2}, \quad (3.11)$$

where $0 \leq \beta_o \leq 1$.

The distance term of the binary metric is calculated as a normalized difference between doublet distances. The doublet distance between $n_{\alpha,1}$ and $n_{\alpha,2}$ is calculated as the Euclidean norm

$$d_{\alpha,1,2} = \|\mathbf{p}_{\alpha,1} - \mathbf{p}_{\alpha,2}\|, \quad (3.12)$$

where $\mathbf{p}_{\alpha,1}$ is the spatial coordinate of $n_{\alpha,1}$ and $\mathbf{p}_{\alpha,2}$ is the spatial coordinate of $n_{\alpha,2}$. Thus, the distance term of the binary metric can be calculated for a pair of doublets in images R and Q by

$$\beta_d = \left| \frac{d_{R,1,2} - d_{Q,1,2}}{d_{R,1,2} + d_{Q,1,2}} \right|, \quad (3.13)$$

where $0 \leq \beta_d \leq 1$. The binary metric can now be defined as the weighted sum of Equations 3.11 and 3.13,

$$\beta = a_\beta \beta_o + b_\beta \beta_d = a_\beta \left(\frac{S_{1,2} + S_{2,1}}{2} \right) + b_\beta \left| \frac{d_{R,1,2} - d_{Q,1,2}}{d_{R,1,2} + d_{Q,1,2}} \right|, \quad (3.14)$$

where a_β and b_β are weight parameters that must satisfy $a_\beta + b_\beta = 1$, and $0 \leq \beta \leq 1$.

In practice, computation times can be drastically reduced by using distance similarity as a constraint prior to computing the signature metrics. In this case, if $|d_{R,1,2} - d_{Q,1,2}| \leq d_T$, where an allowable tolerance d_T is not satisfied, then the doublets are no longer considered for binary correspondence and signatures are not even computed. Computation times are also reduced by using data structures specifically designed to manage the combinatorial complexity of searching for potentially corresponding cliques. These data structures are further discussed in Appendix B.

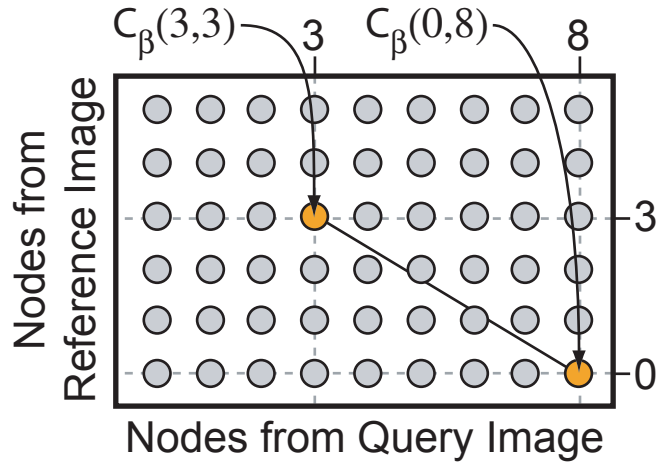


Figure 3.10: Binary correspondence matrix for organizing the binary correspondences between the 6 nodes from the reference image and 9 nodes from the query image. Shown in orange at $C_\beta(0,8)$ and $C_\beta(3,3)$ is the binary metric evaluated between doublets $C_R = \{n_{R,0}, n_{R,3}\}$ and $C_Q = \{n_{Q,3}, n_{Q,8}\}$.

The binary metric evaluations for all doublets in images R and Q are organized in a *binary correspondence matrix* that is denoted as C_β . The binary correspondence matrix is of the same size as the unary matrix, and organized in the same manner with nodes from the reference image along the rows and nodes from the query image along the columns.

Correspondences values are stored differently for doublets however. Consider a doublet in the reference image consisting of nodes $n_{R,a}$ and $n_{R,b}$ and a doublet in the query image consisting of nodes $n_{Q,i}$ and $n_{Q,j}$. By the unary metric, node $n_{R,a}$ corresponds to $n_{Q,i}$ and node $n_{R,b}$ corresponds to $n_{Q,j}$. The binary metric evaluated for these doublets is stored at indices (a, i) and (b, j) of the matrix. An example of a binary correspondence matrix is shown in Fig. 3.10. The binary metric evaluated between doublets $C_R = \{n_{R,0}, n_{R,3}\}$ and $C_Q = \{n_{Q,3}, n_{Q,8}\}$ is highlighted in orange at $\mathbf{C}_\beta(0, 8)$ and $\mathbf{C}_\beta(3, 3)$.

It is possible that a number of nodes will comprise multiple doublets and therefore multiple results from the binary metric may need to be accumulated at certain nodes in \mathbf{C}_β . After the binary metric has been evaluated for all inter-image pairs of doublets, the binary matrix is normalized by the largest accumulated value in the matrix keeping values in the range $[0, 1]$. Thus, the binary correspondence stores the similarity between every node in Ψ_R and Ψ_Q . Nodes are said to correspond if their binary correspondence value is within a pre-determined threshold β_T , i.e., nodes a and i correspond if $\mathbf{C}_\beta(a, i) \leq \beta_T$. Similar to the threshold for the unary metric, the threshold on β permits for flexibility in determining correspondences.

3.4 TRINARY CORRESPONDENCES

Correspondences between medial node *triplets* are introduced to improve specificity over doublet correspondences by considering the geometric configuration between three medial nodes in a clique. Greater specificity is expected since a collection of 3 nodes in 3D image space accounts for more degrees of freedom than 2 nodes in 3D image space. Even forgetting, for the moment, the orientation of the individual nodal eigensystems and treating each node simply as a point location, the degrees of freedom in terms of the affine transform can be analyzed to map a clique in one image to a clique in the other image. A 3D affine transform has 9 independent degrees of freedom to account for translation, rotation, and anisotropic scale. Doublets have 6 degrees of freedom: 3 translations, 2 rotations, and 1 scale. Triplets have 8 degrees of freedom: 3 translations, 3 rotations, and 2 scales. A method to account for the remaining degree of freedom (scale orthogonal to the plane of the triplet) is the subject of Chapter 4.

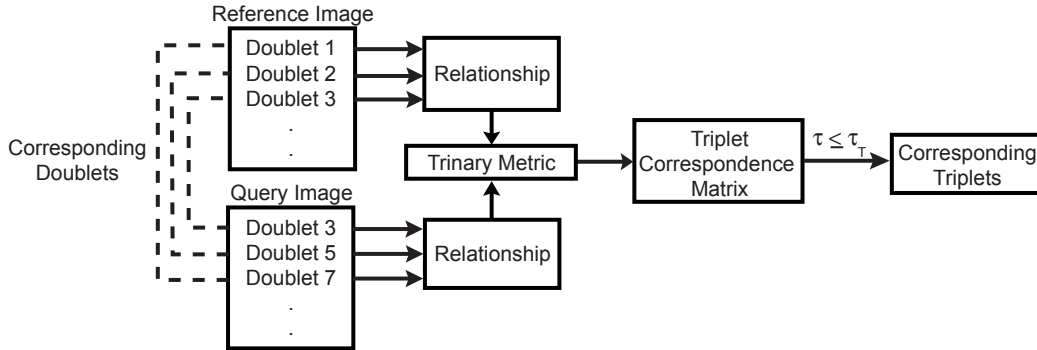


Figure 3.11: Identifying corresponding triplets from corresponding doublets with the trinary metric. Note that corresponding doublets do not necessarily have the same identification number.

An overview for determining triplet correspondences is diagrammed in Fig. 3.11. As previously stated, a triplet is a clique consisting of three medial nodes, as shown in Fig. 3.12. Each vertex of a triplet is a medial node, where $n_{\alpha,j}$ denotes the j^{th} node of the triplet in image α . Similar to doublets, triplets are formed from a pair of cliques that consist of nodes with correspondences in the other image. This is illustrated in Fig. 3.12. For a reference image triplet, nodes $n_{R,1}$ and $n_{R,2}$ make-up a doublet, and the corresponding doublet in the

query image triplet consists of nodes $n_{Q,1}$ and $n_{Q,2}$. Nodes $n_{R,1}$ and $n_{R,3}$ make-up the other doublet in the reference image triplet with its corresponding doublet in the query image consisting of nodes $n_{Q,1}$ and $n_{Q,3}$. Also note that the doublets comprising a triplet share a *base node*, i.e., $n_{R,1}$ and $n_{Q,1}$ for the triplets in the reference and query images, respectively. Within a triplet, “twist” of the third “implied” doublet is accounted for by calculating the binary metric between the first two doublets, e.g. between, $n_{R,2}$ and $n_{R,3}$ and $n_{Q,2}$ and $n_{Q,3}$ in Fig. 3.12.

The *triplet angle* is defined as the angle between the first two doublets, denoted as θ_α .

The *triplet length* is defined as the Euclidean norm between the non-base nodes, e.g., the length of the third, implied doublet, and is denoted as L_α .

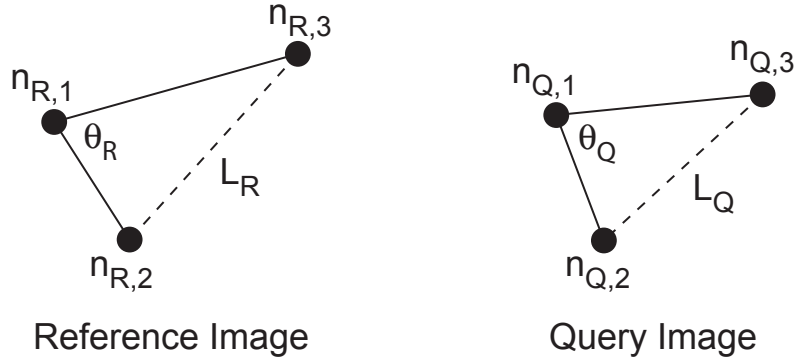


Figure 3.12: Triplets from the reference and query images are shown. Each vertex is a medial node, where $n_{\alpha,j}$ denotes the j^{th} node of the triplet in image α . For clarity, corresponding nodes have the same subscript of j . A triplet can be defined by two doublets (solid lines) that share a base node with the third “implied” doublet (dashed line). The triplet angle is θ_α and the triplet distance is L_α as defined relative to the base node and the third “implied” doublet.

Triplet correspondences are implied when the triplets share similar geometry. For example, in Fig. 3.12, suppose doublets $(n_{R,1}, n_{R,2})$ and $(n_{R,1}, n_{R,3})$ make-up a triplet in image R and doublets $(n_{Q,1}, n_{Q,2})$ and $(n_{Q,1}, n_{Q,3})$ make-up a triplet in image Q. These two triplets correspond if the triplet angles are similar, since constituent doublets were already shown to correspond. Correspondences between triplets is quantifiable by the trinary metric, which is defined for angles θ_R and θ_Q as

$$\tau_\theta = \left| \frac{\theta_R - \theta_Q}{\theta_R + \theta_Q} \right|. \quad (3.15)$$

Alternatively, distances L_R and L_Q may be used since angle calculations may be computationally difficult. Triplet distance is also more sensitive to differences in doublet distance, whereas angle is independent of doublet distances, i.e., if the doublet distances are changed for two identical triplets, the triplet lengths will differ yet the angles will remain the same. The trinary metric is defined for lengths L_R and L_Q as

$$\tau_L = \left| \frac{L_R - L_Q}{L_R + L_Q} \right|. \quad (3.16)$$

As with the unary and binary metrics, both forms of the trinary metric evaluate to a value between 0 and 1, where 0 indicates a perfect correspondence. Results reporting triplet correspondences in this dissertation made use of the trinary metric defined in Eq. 3.16 for reasons of computational stability.

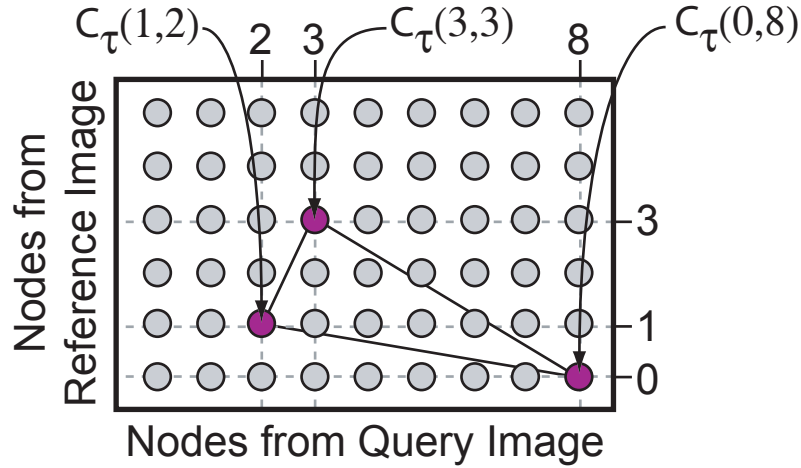


Figure 3.13: Trinary correspondence matrix for organizing the trinary correspondences between the 6 nodes from the reference image and 9 nodes from the query image. The trinary metric evaluated for a triplet consisting of nodes (3,3), (0,8), and (1,2) is highlighted in purple.

Evaluations using the trinary metric for all triplets in images R and Q are organized in a *trinary correspondence matrix*, denoted as \mathbf{C}_τ . The trinary correspondence matrix is the same size as the unary and binary matrices, and organized in the same manner. An example trinary correspondence matrix is shown in Fig. 3.13 with two triplets: one from image R and one from image Q. The triplet in image R consists of doublets $(n_{R,0}, n_{R,1})$ and $(n_{R,0}, n_{R,3})$, where $n_{R,0}$ is the base node. The triplet in image Q consists of doublets

$(n_{Q,8}, n_{Q,2})$ and $(n_{Q,8}, n_{Q,3})$, where $n_{Q,8}$ is the base node. By means of the binary metric, $(n_{R,0}, n_{R,1})$ corresponds to $(n_{Q,8}, n_{Q,2})$, and $(n_{R,0}, n_{R,1})$ corresponds to $(n_{Q,3}, n_{Q,3})$. The trinary metric evaluated for these two triplets is stored at indices $(1, 2)$, $(3, 3)$, and $(0, 8)$ of \mathbf{C}_τ .

Similar to the binary correspondence matrix, the similarity between every node in Ψ_R and Ψ_Q is stored by accumulating trinary metric values in \mathbf{C}_τ , and normalizing them by the largest accumulated value in the matrix, which keeps values in the range $[0, 1]$.

Threshold τ_T is used when computing triplet correspondences to allow for flexibility in determining triplet correspondences. Triplets are said to correspond if $\tau < \tau_T$, where τ_T is a threshold. Individual nodes correspond if their values in the correspondence matrix are within threshold τ_T , i.e., nodes a and i correspond if $\mathbf{C}_\tau(a, i) \leq \tau_T$.

3.5 MEASURING CORRESPONDENCE SENSITIVITY AND SPECIFICITY

A Receiver Operating Characteristic (ROC) curve assesses the ability of a test to discriminate between correct and incorrect results, delineating the trade-off for a chosen test parameter [99, 100, 101]. The methodology of ROC curve analysis was developed in 1954 by Tanner, *et al.* [102] to measure the detectability of airplanes by radar. In 1968, Lusted saw the utility in ROC curves and used them to study medical decision making and the efficacy of medical imaging devices [103, 104].

In this dissertation, ROC curves are analyzed to evaluate the accuracy of each correspondence metric in determining correct medial correspondences. For the remainder of this chapter, ROC curve analysis will be presented in the context of testing the correspondence metrics. Consider medial nodes extracted from two images with the objective of identifying corresponding medial nodes. A correspondence test is performed with a specified metric. The test, as described above, has a binary outcome based on a threshold applied to the metric. A *positive test* indicates that the nodes likely correspond, and a *negative test* indicates that the nodes likely do not correspond. The test is repeated with a change in parameter setting, namely the metric threshold. All test results are then compared to a ground-truth for correct correspondences, and an ROC curve is generated and analyzed.

The ground-truth for node correspondence is established by using two identical images, so that nodes that are known to correspond will have identical image coordinates. For this section, notation is simplified from previous sections. A node in the reference and query images is denoted simply as a and i , as opposed to $n_{R,a}$ and $n_{Q,i}$, respectively. The ground-truth correspondence between two given nodes a and i is defined as

$$D_{a,i} = \left(\frac{k_{a,i}}{k_{max}} \right)^{\phi_D}, \quad (3.17)$$

where $k_{a,i}$ is the Euclidean distance between the locations of nodes a and i , as if their coordinates were projected into the same image space, k_{max} is the maximum $k_{a,i}$ computed between the sets of nodes, and ϕ_D is a parameter for controlling the effective threshold

between correct and incorrect correspondences. The ground-truth is always within the range $[0, 1]$, where 0 implies perfect correspondence.

Table 3.1: All possible classifications for ROC analysis of node correspondences. Positive/Negative indicates correspondences as determined by test results and Correct/Incorrect indicates correspondences known via the ground-truth.

| Node Correspondence Classifications | | |
|-------------------------------------|----------------|------------------|
| | Correct | Incorrect |
| Positive | True Positive | False Positive |
| Negative | False Negative | True Negative |

The ground-truth permits test results to be separated into one of four groups, as shown in Table 3.1. As determined by the ground-truth, correspondence can be classified as *correct* or *incorrect*, where $D_{a,i} = 0$ indicates correct correspondence, and $D_{a,i} = 1$ indicates incorrect correspondence. Thresholds are established for the ground-truth (D_T) and correspondence values (C_T) to classify the “correctness” of the correspondence test. Each test result is classified as True Positive (TP), False Positive (FP), True Negative (TN), or False Negative (FN). Each classification is defined as follows:

- **True Positive:** the ground-truth indicates a *correct* correspondence, and the correspondence test is *positive* for correspondence.
- **False Positive:** the ground-truth indicates an *incorrect* correspondence, but the correspondence test is *positive* for correspondence.
- **True Negative:** the ground-truth indicates an *incorrect* correspondence, and the correspondence test is *negative* for correspondence.
- **False Negative:** the ground-truth indicates a *correct* correspondence, but the correspondence test is *negative* for correspondence.

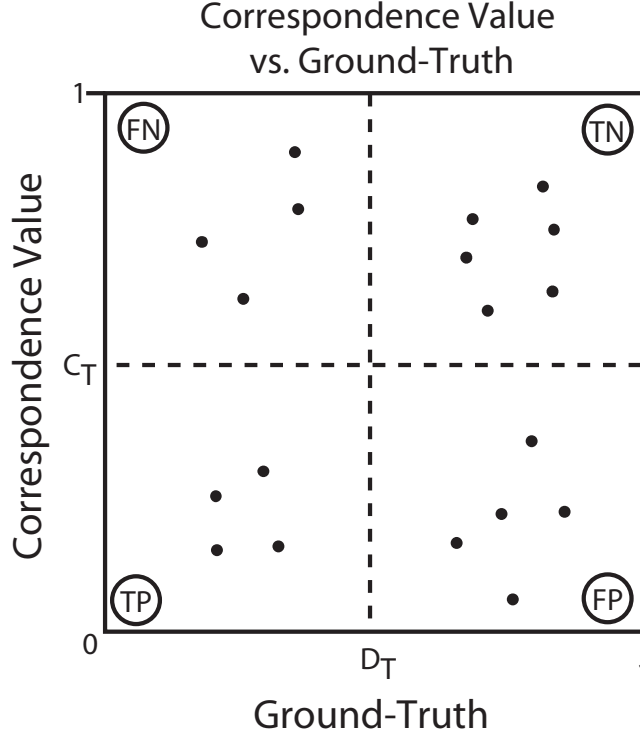


Figure 3.14: Scatter plot of correspondence values vs. ground-truth. Also shown are the test classifications (circled) and thresholds for classifying the test results.

A scatter plot of correspondence values versus ground-truth is shown in Fig. 3.14. Applying the thresholds on correspondence values and ground-truth will partition the plot into the 4 possible test classifications.

An ROC curve is generated by calculating the true positive fraction (TPF) versus false positive fraction (FPF) over a range of values for C_T . The TPF is a measure of the test's *sensitivity* in detecting true positives, i.e., in calling a correct case positive. The TPF is calculated as the ratio of true positives to total correct classifications:

$$TPF = Sensitivity = \frac{N_{TP}}{N_{FN} + N_{TP}}, \quad (3.18)$$

where N_{TP} is the number of true positives and N_{FN} is the number of false negatives. If $TPF = 1$, the correspondences test achieves 100% sensitivity since all test results were true

positive and none were false negative. *Specificity* is the probability of classifying an incorrect case negative, i.e.,

$$Specificity = \frac{N_{TN}}{N_{TN} + N_{FP}}, \quad (3.19)$$

where N_{TN} is the number of true negatives and N_{FP} is the number of false positives.

The false positive fraction is simply one minus the specificity or the ratio of false positives to total incorrect classifications:

$$FPF = 1 - Specificity = \frac{N_{FP}}{N_{TN} + N_{FP}}. \quad (3.20)$$

If $FPF = 0$, the correspondences test achieves 100% specificity since all test results were true negative and none were false positive.

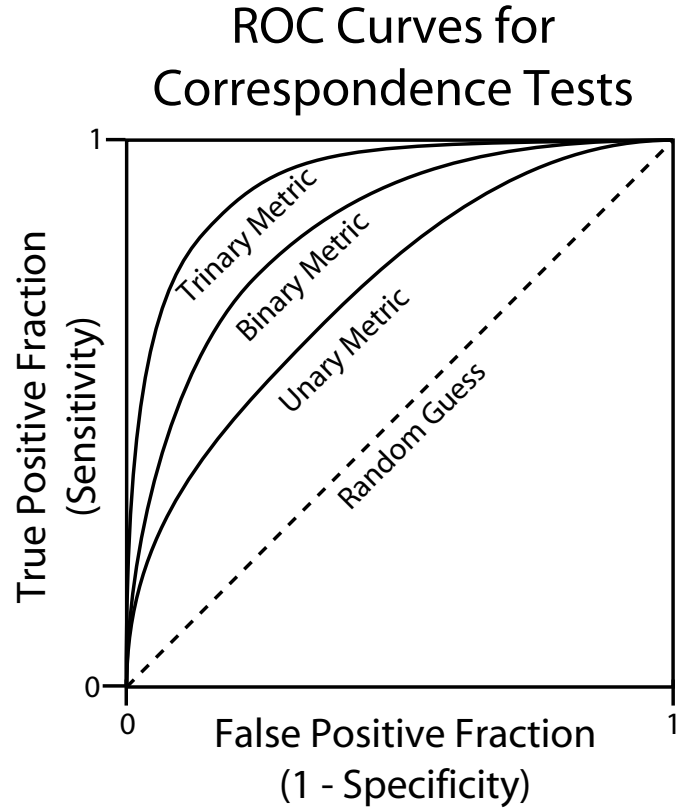


Figure 3.15: Theoretical ROC curves for the medial correspondence metrics. The dashed 45° line indicates a test that involves random guessing.

Plotting TPF vs. FPF for a number of trials with varying values for C_T results in an ROC curve. Three ROC curves are shown together in Fig. 3.15 illustrating theoretical expectations for the three different correspondence metrics. It is expected that the unary metric will be somewhat more accurate than randomly guessing (just above the 45° line) while the trinary metric will be the most accurate correspondence test, approaching vertical on the left and horizontal on the top. The binary metric is expected to perform somewhere in between the unary and trinary metrics.

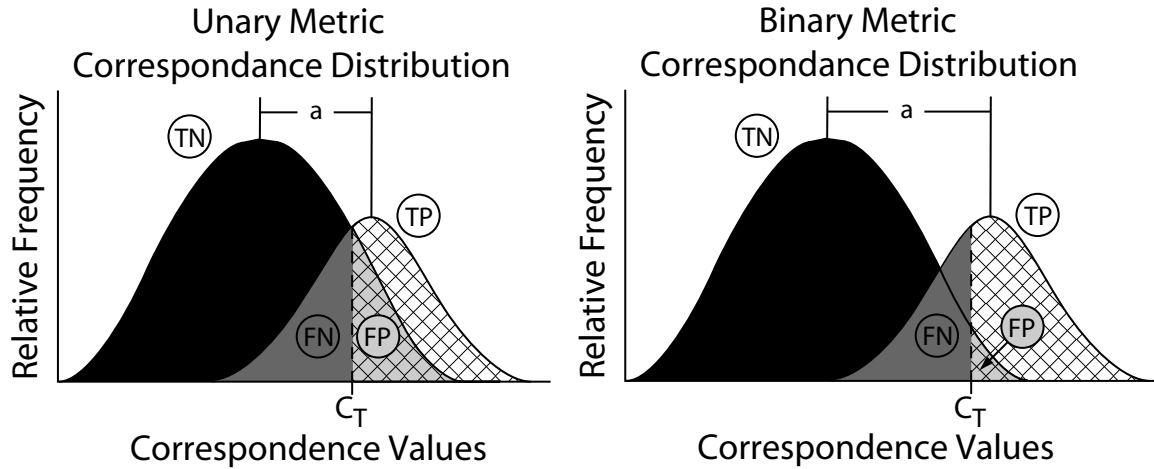


Figure 3.16: Theoretical histograms for the unary and binary correspondence metrics. Left shows a histogram of unary correspondence values and right shows histogram of binary correspondence values. The increase in curve separation a , for the binary metric indicates increased correspondence specificity.

Theoretical histograms, created by binning True Negatives and True Positives, are shown in Fig. 3.16 for the unary and binary metrics. Peak separation distance and curve width are indicative of a metric test's specificity. An increase in separation distance and decrease in peak width results in a reduction in the number of false negatives and false positives, thereby increasing specificity. As shown in Fig. 3.16, the separation distance for the binary test is expected to be larger than the separation distance for the unary test.

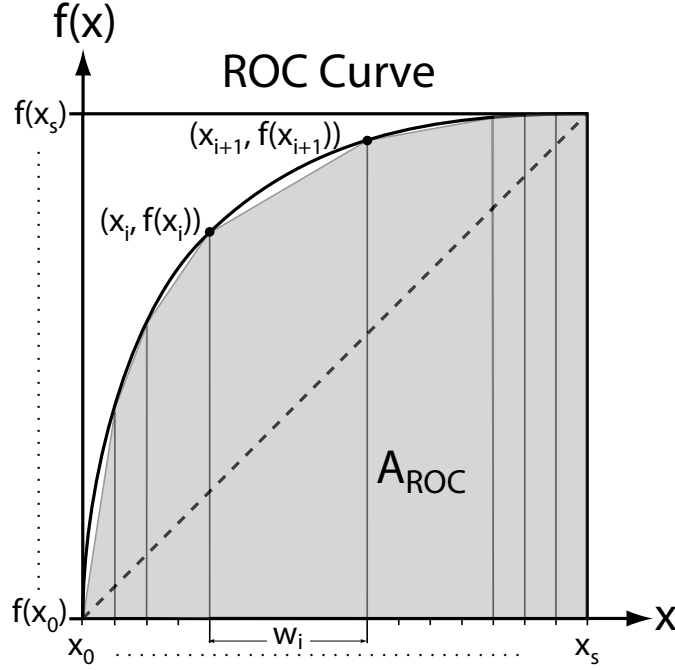


Figure 3.17: Computing the area under the ROC curve with trapezoidal rule. The area of each trapezoidal region is summed for the approximated area under the ROC curve.

A measure of accuracy that encompasses both sensitivity and specificity can be quantified by calculating the area under the ROC curve, A_{ROC} . The area under the ROC curve makes no assumptions about the underlying distribution of the underlying histograms, e.g., Fig. 3.16. The A_{ROC} for randomly guessing whether a correct correspondence exists is 0.5 and the A_{ROC} for a test that is always correct is 1.0. There are many parametric and non-parametric techniques for estimating A_{ROC} . The trapezoidal rule is used in this dissertation to estimate A_{ROC} , as illustrated in Fig. 3.17 and described below.

Let $f(x)$ represent the ROC curve. The ROC curve evaluated at a series of s points x_i is denoted as $f(x_i)$. Trapezoids of width $w_i = x_{i+1} - x_i$ are thus formed along the ROC curve such that a chord connects the points $(x_i, f(x_i))$ and $(x_{i+1}, f(x_{i+1}))$. The more trapezoids that are utilized the more accurately A_{ROC} can be approximated as the sum of all trapezoid areas,

$$A_{ROC} \approx \sum_{i=0}^{s-1} \frac{x_{i+1} - x_i}{2} (f_{x_i} + f_{x_{i+1}}). \quad (3.21)$$

3.6 EMPIRICAL RESULTS

In these following sections, experimental results are presented for several 3D computer generated data sets to demonstrate 1) the framework for automatically determining feature-based correspondences and 2) correspondence accuracy via ROC curve analysis.

The voxel intensity of the test images used were all 8-bit with a dynamic range of 0 - 255. The first data set consisted of two concentric ellipsoids of various orientations. A sample of various images from this data set is shown in Fig. 3.18. Two concentric ellipsoids were chosen to yield a population of nodes differing in dimensionality and scale. It was expected that cylindrical nodes would be extracted from the inner ellipsoid and slab-like nodes would be extracted from the “shell” between the ellipsoids. Each image from this data set was $100 \times 100 \times 100$ voxels in size with origin (0, 0, 0) and center (50, 50, 50). The orientation of each ellipsoid is with respect to the image coordinate system (shown in Fig. 3.18). The inner ellipsoid axes were 25, 50, and 15 voxels in length along the x-axis, y-axis, and z-axis, respectively. The outer ellipsoid axes were 45, 70, and 35 voxels in length along the x-axis, y-axis, and z-axis, respectively. The space between the ellipsoids constitute a 20 voxel-wide “shell.” Voxels within the inner ellipsoid had an intensity of 255 and voxels within the shell had an intensity of 128. Voxels outside both ellipsoids had an intensity of 0.

The second data set consisted of images obtained from a Simulated Brain Database (SBD) called BrainWeb. BrainWeb images are generated using the MRI simulator developed at the McConnell Brain Imaging Centre at the Montréal Neurological Institute. From a webpage interface, the MRI simulator allows users to control various acquisition parameters, and to realistically model noise from the data acquisition and image reconstruction processes [105, 106, 107, 108]. Several cross-sections of an image obtained from BrainWeb are shown in Fig. 3.19.

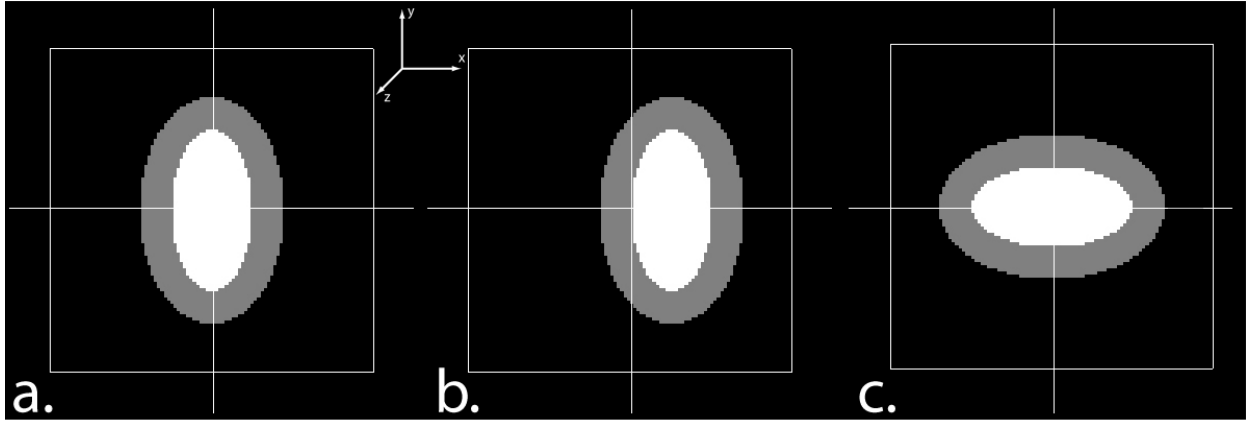


Figure 3.18: Cross-sections of various images containing two concentric ellipsoids. The image coordinate system is shown between **a** and **b**. **a**). Two concentric ellipsoids centered in the image. **b**). Two concentric ellipsoids centered in the image and translated along the x-axis. **c**). Concentric ellipsoids centered in the image and rotated by 90° about the z-axis.

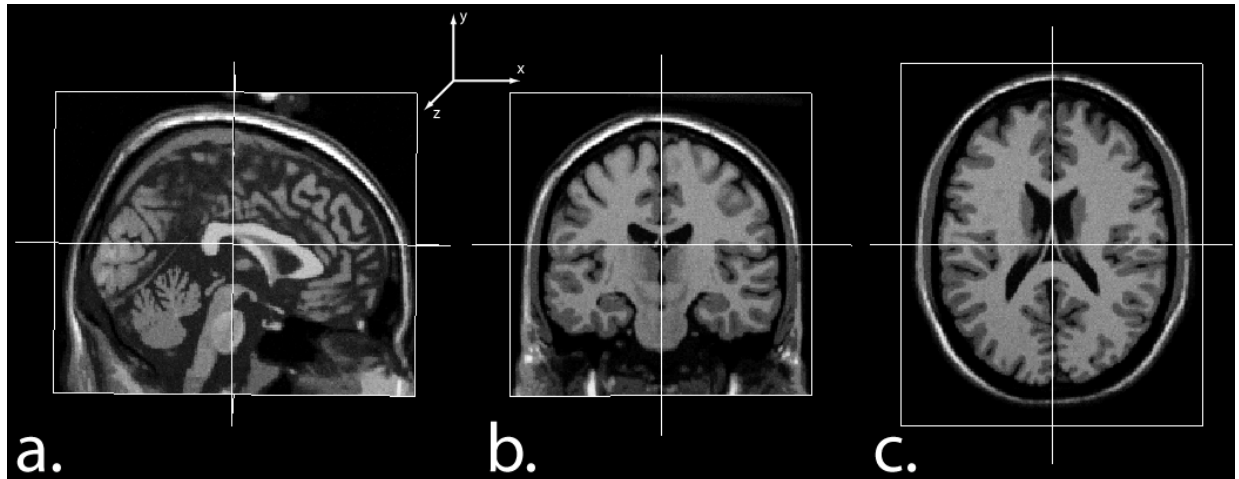


Figure 3.19: **a**). Sagittal, **b**). coronal, and **c**). transverse cross-sections of a 3D image obtained from the BrainWeb database.

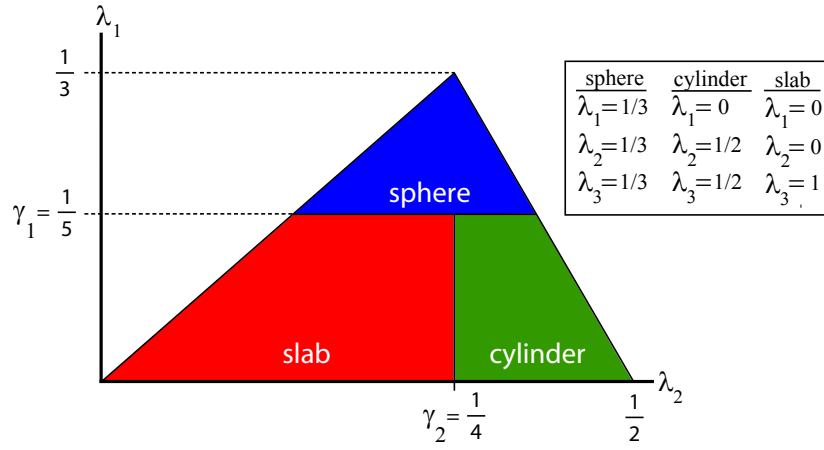


Figure 3.20: The lambda triangle with eigenvalue thresholds defined. The eigenvalues in the box describe the perfect sphere, cylinder, and slab, i.e., the vertices of the lambda triangle.

A medial node is visualized as one or more lines orientated in the predominate direction(s) of the corresponding eigenvector(s). A spherical node is visualized as 3 intersecting lines, a cylindrical node as 2 intersecting lines, and a slab-like node as 1 line. The length of the lines are equal to the node's scale. The color of a medial node is determined from the lambda triangle that was briefly described in Section 2.3.2. For the results presented in the remainder of this dissertation, thresholds on λ_1 and λ_2 were set to $\gamma_1 = \frac{1}{5}$ and $\gamma_2 = \frac{1}{4}$, respectively. As shown in Fig. 3.20, these thresholds divide the lambda triangle into 3 regions. Each region designates dimensionality and color as follows:

- When $\frac{1}{5} < \lambda_1 \leq \frac{1}{3}$ for all λ_2 , the medial node is classified as spherical and displayed in blue.
- When $0 \leq \lambda_1 \leq \frac{1}{5}$ and $\frac{1}{4} < \lambda_2 \leq \frac{1}{2}$, the medial node is classified as cylindrical and displayed in green.
- When $0 \leq \lambda_1 \leq \frac{1}{5}$ and $0 \leq \lambda_2 \leq \frac{1}{4}$, the medial node is classified as slab-like and displayed in red.

Since the lambda triangle bounds all possible eigenvalues, any medial node is assigned to one of the three colors.

3.6.1 Automated Feature-Based Correspondences

3.6.1.1 Identical Synthetic Images For this first experiment, the concentric ellipsoids image shown in Fig 3.18a was used for both the reference and query images. Medial nodes were extracted from each image with the same core atom parameter settings to ensure identical sets of nodes. These parameter settings are shown in Table 3.2. Given identical core atom parameters and images, nodes from each image are stored in their respective lists in the same order, which allows correspondences to be known based on the position of the nodes in the lists, i.e., $n_{R,i}$ corresponds to $n_{Q,j}$ for $i = j$. Consequently, known correspondences will be on the diagonal of a given correspondence matrix. Qualitative observation of the correspondence matrix diagonal is the basis of validation for this preliminary experiment. The remainder of this chapter presents 1) preliminary results to demonstrate global rotational and translational invariance of the correspondence methods, and 2) the results of ROC curve analysis for simulated MRI brain images.

Table 3.2: Parameter settings used to extract medial nodes from images containing concentric ellipsoids.

| Parameter | Description | Value |
|----------------------------|---|-------------------------------|
| Blur Repetitions | Number of applications of a 2×2 binomial kernel | 2 |
| Boundary Point Threshold | Intensity threshold for determining boundary points | 32 |
| Distance Minimum | Minimum search distance for finding opposing boundary point | 10 |
| Distance Maximum | Maximum search distance for finding opposing boundary point | 40 |
| Face-to-Faceness Tolerance | Allowable tolerance | 0.10 |
| Polarity | Direction to search for opposing boundary point | 0 (direction of the gradient) |

A total of 42 medial nodes were extracted from each of the reference and query images. The set of nodes is shown with the image in Fig. 3.21. As expected, cylindrical nodes were found within the inner ellipsoid and slab-like nodes were found within the shell. Note that the viewpoint was rotated to provide clarity for observing the medial nodes. The green intersecting lines indicate cylindrical nodes from the inner ellipsoid, and the red lines indicate slab-like nodes from the shell.

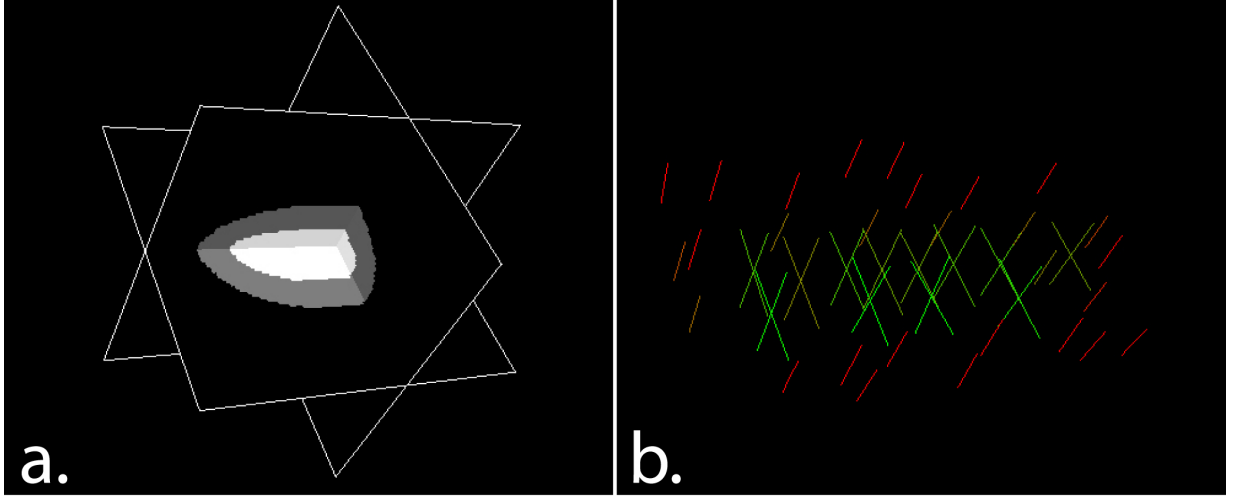


Figure 3.21: a). An image consisting of two concentric ellipsoids centered at the image origin. Note that voxel intensities are uniform as described in the text, but are displayed here with a shadowing effect for clarity. b). The set of extracted medial nodes. Green intersecting lines indicate cylindrical nodes from the inner ellipsoid and red lines indicate slab-like nodes from the shell between the ellipsoids.

Singlet correspondences were calculated via the unary metric using a threshold of $\mu_T = 0.10$. An example of unary correspondences is shown in Fig. 3.22. In Fig. 3.22a, slab-like node $n_{R,9}$ is selected and the corresponding nodes are highlighted in Fig. 3.22b. The correct node correspondence is labeled $n_{Q,9}$. In Fig. 3.22c, cylindrical node $n_{R,24}$ is selected and the corresponding nodes are highlighted in Fig. 3.22d. The correct node correspondence is labeled $n_{Q,24}$. By qualitative assessment, multiple nodes of similar dimensionality and scale are highlighted, including the actual correct correspondence.

The unary correspondence matrix is shown in Fig. 3.23. White indicates a perfect unary correspondence value of 0, and white values off-diagonal indicate incorrect correspondences.

Correct correspondences for the selected nodes in Figs. 3.22a and 3.22c are highlighted by red boxes along the matrix diagonal (thin, red dashed-line), i.e., the nodes indexed by $\mathbf{C}_\mu(9, 9)$ is the correct correspondence for the slab-like node, and nodes indexed by $\mathbf{C}_\mu(24, 24)$ is the correct correspondence for the cylindrical node. The highlighted correspondences in Figs. 3.22b and 3.22d can be seen in the correspondence matrix by observing the row of a selected node. The values in this row that are above the correspondence threshold have sufficient correspondence and are highlighted in the query image. For example, the row labeled n_{R_9} , defined as $\mathbf{C}_\mu(9, k)$, contains the unary metric evaluated between n_{R_9} for all k nodes in the query image. For each node in the query image, if $\mathbf{C}_\mu(9, k) \leq 0.10$, node k is highlighted in Fig. 3.22b. Likewise, the same can be observed in the unary correspondence matrix for node $n_{R_{24}}$ and its highlighted correspondences shown in Fig. 3.22d.

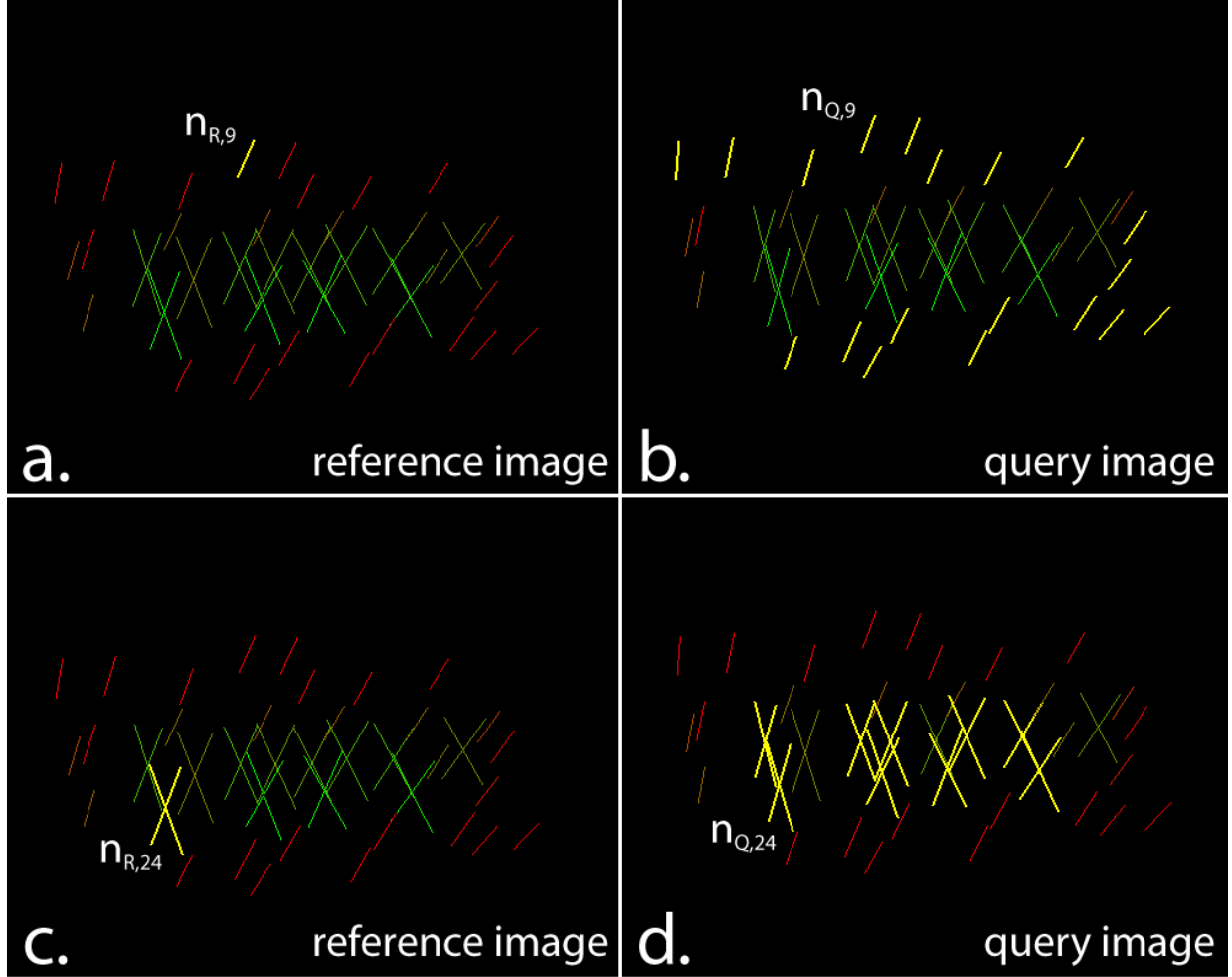


Figure 3.22: Example of unary correspondences between nodes extracted from identical images containing two concentric ellipsoids. **a).** A slab-like node in the shell of the reference image is selected (yellow) and the corresponding nodes in the query image are highlighted in **b.** **c).** A cylindrical node within the inner ellipsoid of the reference image is selected (yellow) and the corresponding nodes in the query image are highlighted (yellow) in **d).**

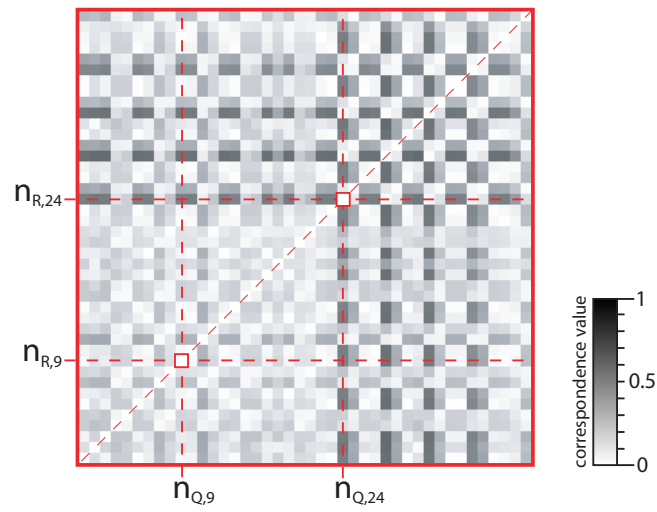


Figure 3.23: Unary correspondence matrix calculated from identical images containing two concentric ellipsoids. White indicates perfect correspondence and nodes along the diagonal dashed-line are known correspondences. The red squares are the known correspondences for the selected nodes shown in Figs. 3.22a. and 3.22c.

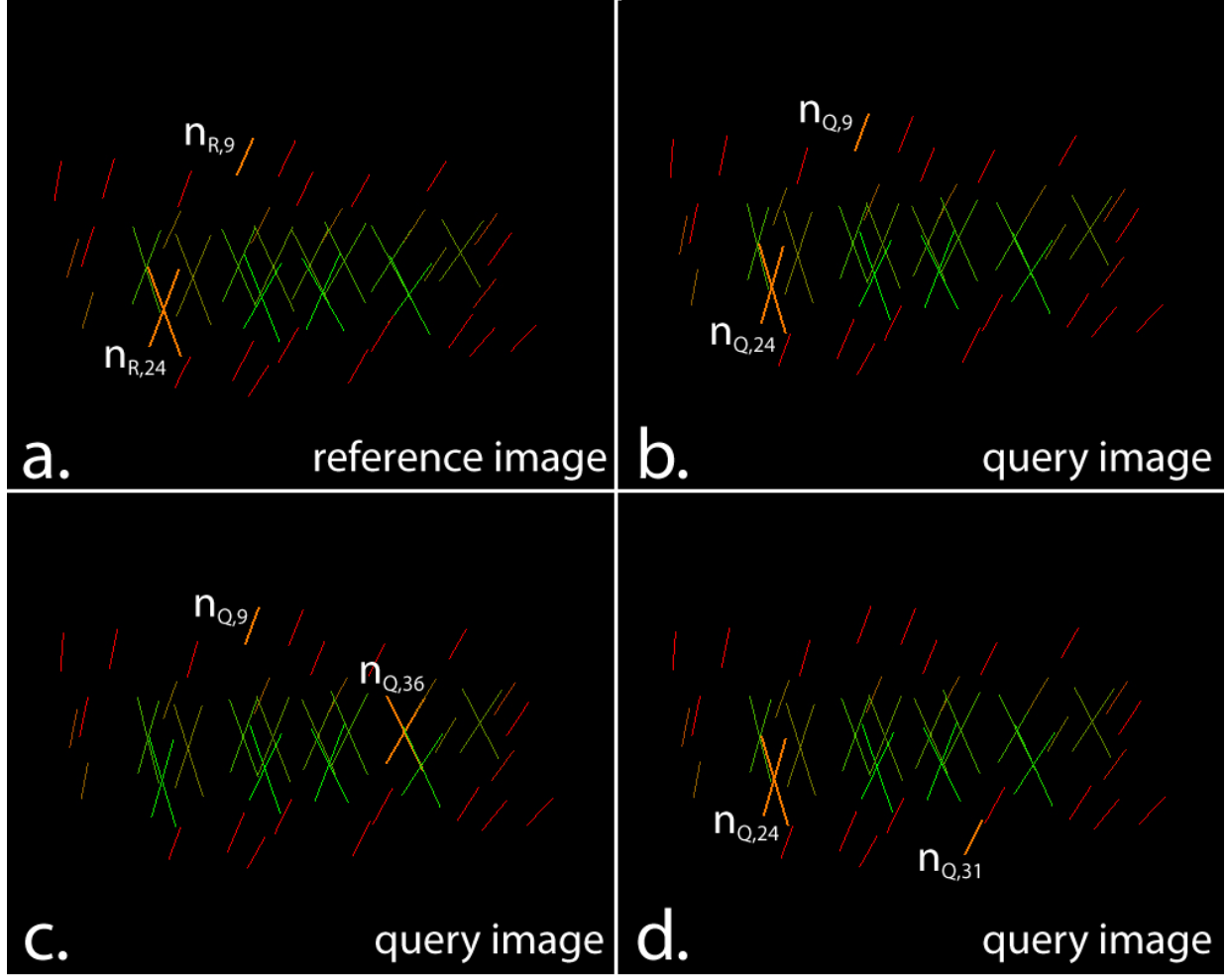


Figure 3.24: Three examples of binary correspondences between doublets extracted from identical images containing two concentric ellipsoids. **a).** A doublet consisting of a cylindrical and slab-like node is selected (orange) in the reference image. **b).** The correctly corresponding doublet is highlighted (orange). Two incorrect correspondences are highlighted (orange) in **c.** and **d.**

Doublet correspondences were determined from corresponding singlets via the binary metric with $\beta_T = 0.10$. A doublet consisting of the same nodes previously described for the unary metric is shown in Fig. 3.24a and its corresponding doublet is shown in Fig. 3.24b. The binary correspondence matrix is shown in Fig. 3.25. Note how correspondence values off the diagonal are lower as compared to the unary correspondence matrix, signifying greater specificity (fewer false positives). The reference image doublet $(n_{R,9}, n_{R,24})$ and its corresponding doublet $(n_{Q,9}, n_{Q,24})$ are highlighted by the orange line labeled *a-b*. Individually, nodes $n_{R,9}$ and $n_{Q,9}$ appear to correspond, and nodes $n_{Q,24}$ and $n_{Q,24}$ appear to correspond

since their correspondence values are fairly white. Since these pairs of nodes are on the matrix diagonal and have the same index, they do indeed correspond.

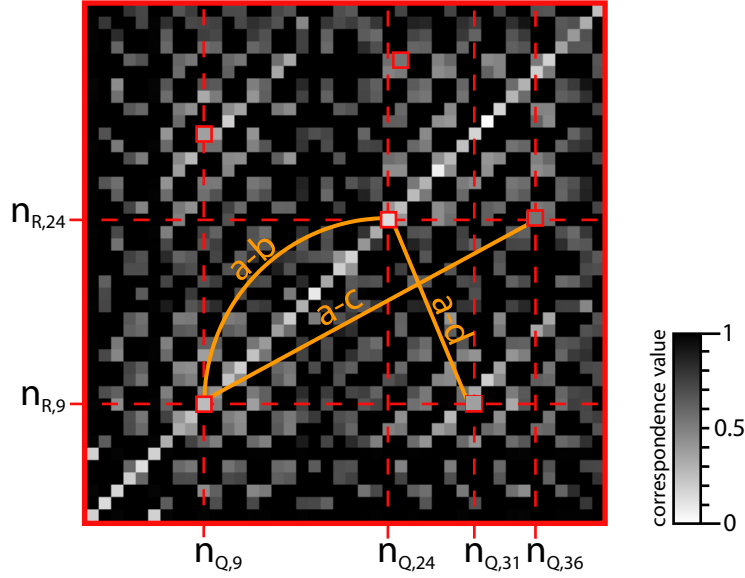


Figure 3.25: Binary correspondence matrix calculated from identical images containing two concentric ellipsoids. Red squares connected by an orange line indicate the pair of doublets from Fig. 3.24.

Shown in Figs. 3.24c and 3.24d are examples of incorrect correspondences (false positives). It is clear that these correspondences are incorrect by observing the location of the cylindrical nodes relative to each other. However, since the doublet distance and the doublet orientation appear to be similar, the doublets satisfy the binary correspondence criteria. These incorrect correspondences can be observed in the binary correspondence matrix (Fig. 3.25). The pair of doublets shown in Figs. 3.24a and 3.24c is connected by the orange line labeled *a-c*. The slab-like nodes $n_{R,9}$ and $n_{Q,9}$ are on the matrix diagonal, but the cylindrical nodes $n_{R,24}$ and $n_{Q,36}$ are off-diagonal. Since the correspondence values for the cylindrical nodes are “more gray than white,” they could be overlooked when considering individual node correspondences by using a proper threshold. For the example shown in Fig. 3.24d, the pair of doublets is connected by the orange line labeled *a-d* in the binary correspondence matrix. In this case, the cylindrical nodes $n_{R,24}$ and $n_{Q,24}$ are on the matrix diagonal, but the slab-like nodes $n_{R,9}$ and $n_{Q,31}$ are off-diagonal. Again, a proper threshold could distinguish between true positives on the diagonal and these false positives, but greater specificity can

be achieved by using the trinary metric, described below. It is the goal of the trinary metric to remove these incorrect correspondences by incorporating more geometric information into the clique.

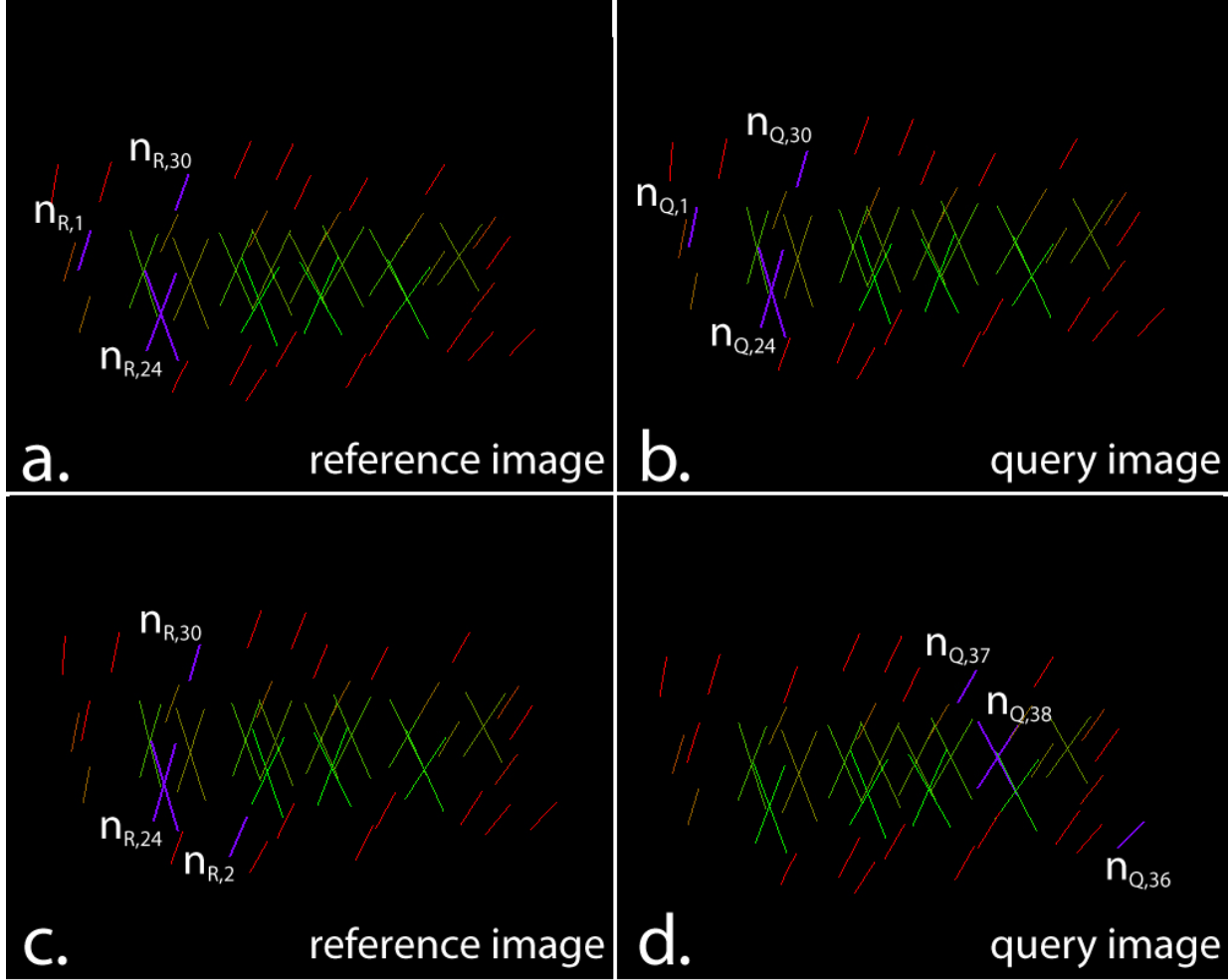


Figure 3.26: Two examples of trinary correspondences between triplets extracted from identical images containing concentric ellipsoids. **a).** A selected triplet (purple) consisting of 2 slab-like nodes and a cylindrical node. **b).** The correctly corresponding triplet is highlighted (purple). **c).** A selected triplet (purple) consisting of 2 slab-like nodes and a cylindrical node. **d).** An incorrectly corresponding triplet is highlighted (purple).

Triplet correspondences were calculated via the trinary metric with $\tau_T = 0.10$. In Fig. 3.26a, a triplet consisting of 2 slab-like nodes and 1 cylindrical node is selected and the correctly corresponding triplet is highlighted in Fig. 3.26b. This correspondence is shown in the trinary correspondence matrix (Fig. 3.27) as 3 red squares connected by the purple line

labeled “a-b.” All 3 nodes are on the diagonal and the indices of the individual nodes are the same and therefore it is known that this is a correct correspondence. An incorrect triplet correspondence is shown in Figs. 3.27c and Fig. 3.27d. Geometrically, the triplets appear similar, but clearly do not correspond based on their spatial location in the image. Incorrect correspondences can be further verified by the triplet correspondence matrix in Fig. 3.27. The incorrect triplet correspondence illustrated in Figs. 3.27c and Fig. 3.27d is shown as red

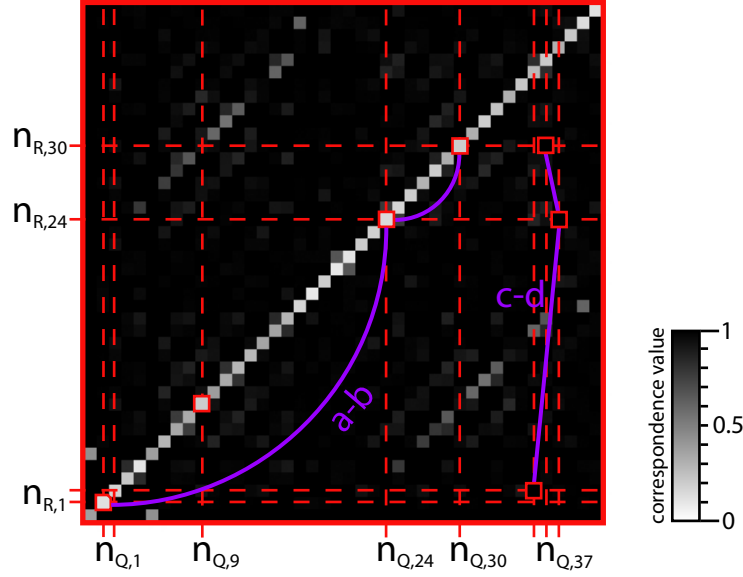


Figure 3.27: Trinary correspondence matrix calculated from identical images containing two concentric ellipsoids. Red squares connected by a purple line indicates the trinary metric evaluated between a pair of triplets from Fig. 3.26.

squares connected by the purple line labeled “c-d.” Note that none of the nodes are on the matrix diagonal. Moreover, the correspondence values are very high as indicated by the very dark red squares. Each of the correspondence matrices are shown side-by-side in Fig. 3.28. From left to right, are the unary, binary, and trinary correspondence matrices. With each successive matrix, the high correspondence values off the diagonal decrease, which indicates that incorrect correspondences are being eliminated and accuracy is being improved.

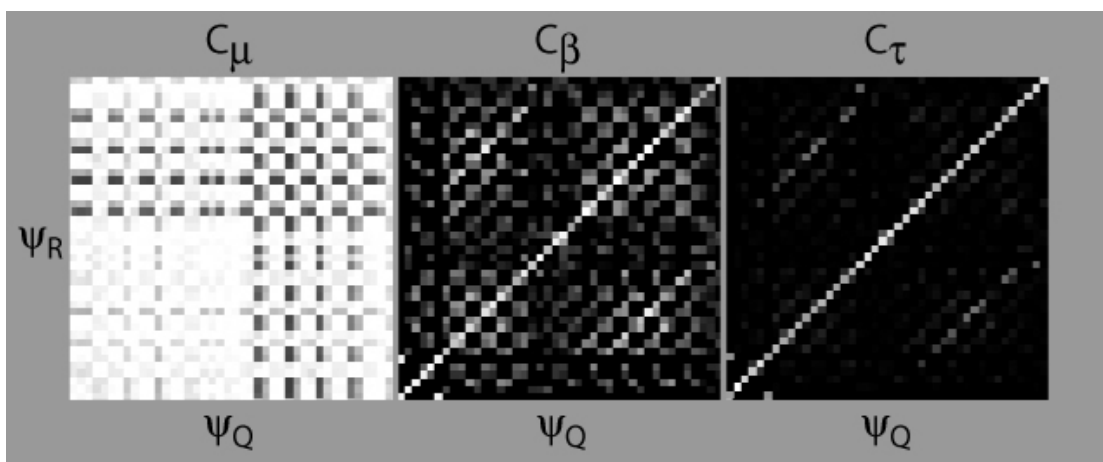


Figure 3.28: Correspondence matrices calculated from the concentric ellipsoids data set. From left to right, are the unary, binary, and trinary correspondence matrices. Note the decreasing number of false correspondences as indicated by a decrease in correspondence values off the matrix diagonal.

3.6.1.2 Rotational and Translational Invariance The global rotational and translational invariance of the correspondence metrics is now demonstrated. Validation for this experiment is purely qualitative. Quantitative validation of the correspondence metrics is the subject of Chapter 7. For this experiment, the reference image consisted of two concentric ellipsoids centered in the image and rotated by 90° (Fig. 3.29a). The query image consisted of two concentric ellipsoids centered in the image and translated along the x-axis by 13 voxels (Fig. 3.29b). Medial nodes were extracted using the same parameters as tabulated above. The nodes extracted from the reference and query images are shown in Figs. 3.29c and 3.29d, respectively. Due to discretization errors in medial node calculations (a result of rotating and translating the images), the set of nodes differ despite identical ellipsoid and core atom parameters. Example correspondences for selected nodes are shown in Fig. 3.30. The “correctness” of the correspondences can be inferred by mentally rotating the reference image nodes 90° clock-wise and recognizing that the selected nodes appear to superimpose on the highlighted nodes. The effects of rotation and translation are now independently investigated with greater detail.

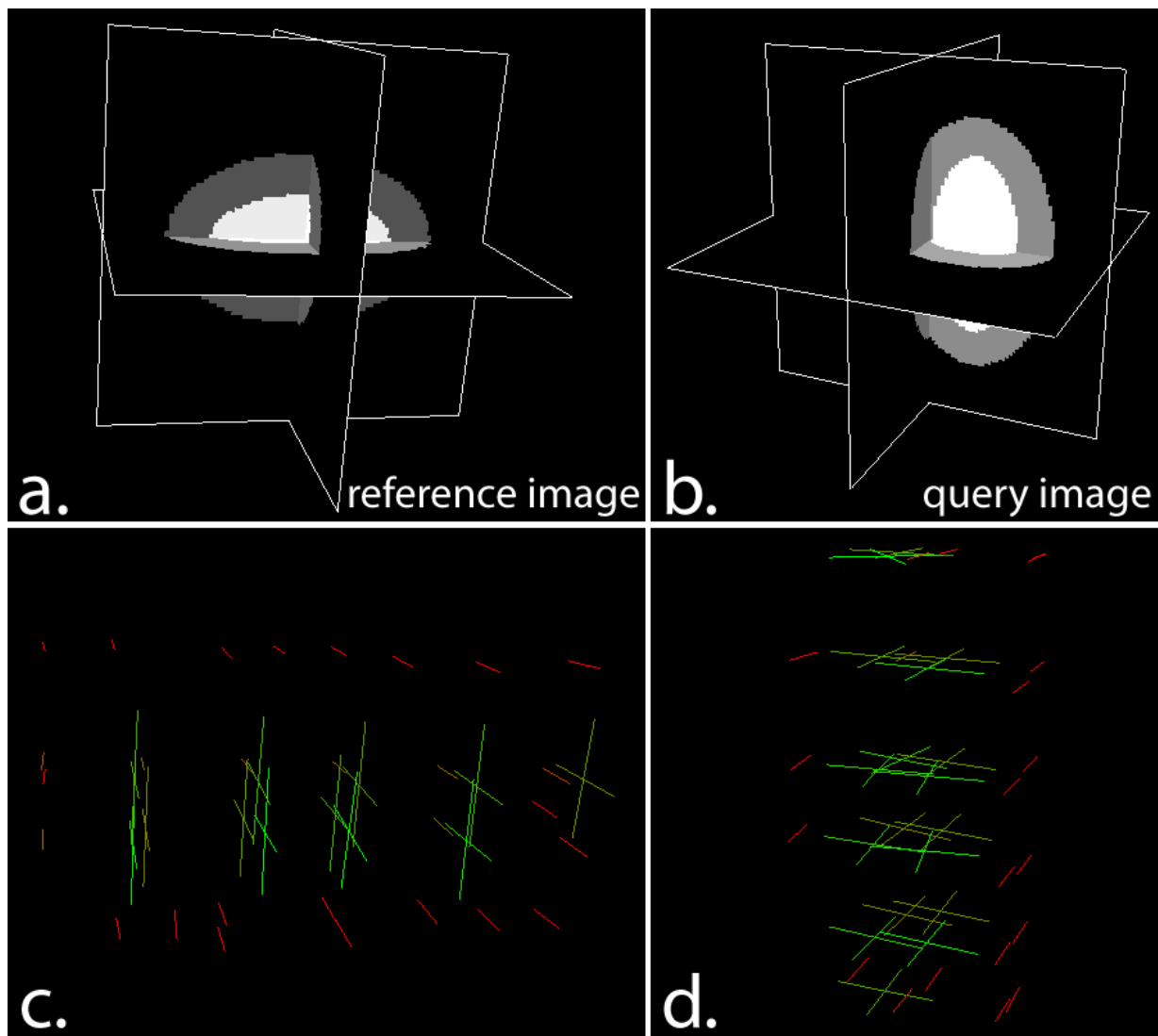


Figure 3.29: **a).** Reference image consisting of 2 concentric ellipsoids centered in the image and rotated by 90° . **b).** Query image consisting of 2 concentric ellipsoids centered in the image and translated along the x-axis by 13 voxels. **c).** Medial nodes extracted from the reference image. **d).** Medial nodes extracted from the query image.

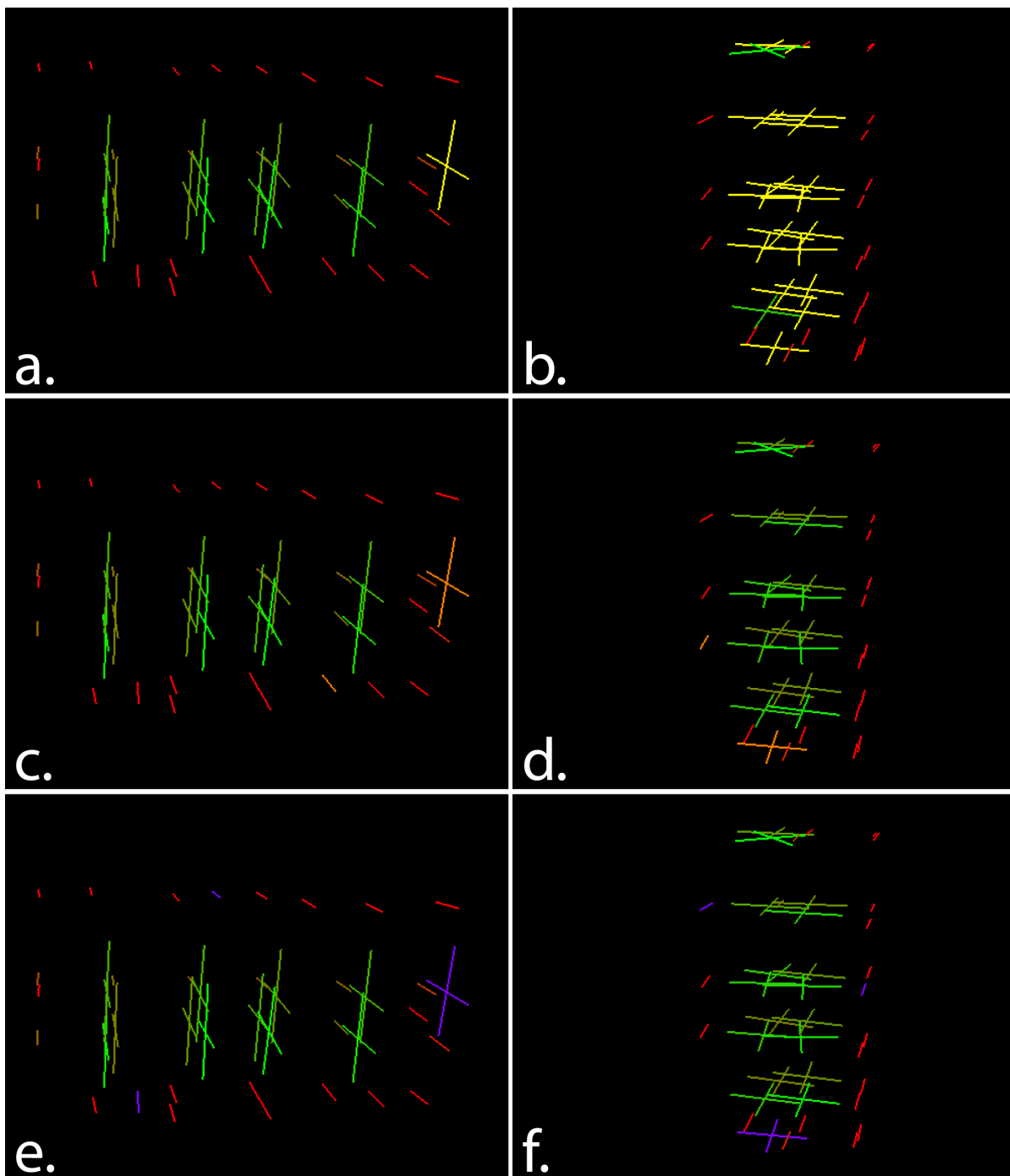


Figure 3.30: Correspondences determined for the reference and query images shown in Fig. 3.29. **a).** A selected cylindrical node (yellow) in the reference image and **b).** its corresponding nodes. **c).** A selected doublet (orange) consisting of a cylindrical and slab-like node, and its **d).** corresponding doublet. **e).** A selected triplet (purple) consisting of a cylindrical node and 2 slab-like nodes, and its **f).** corresponding triplet.

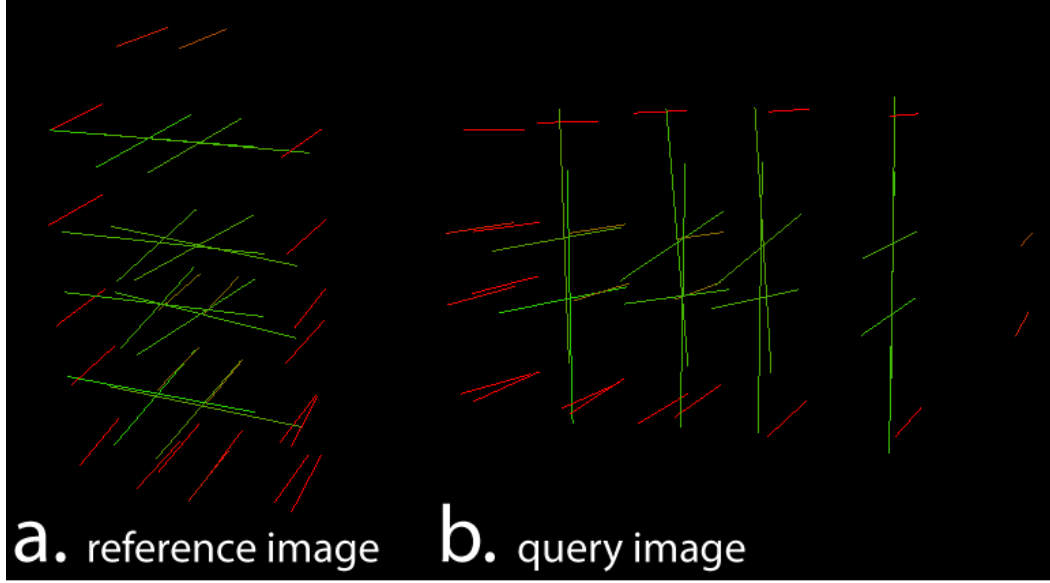


Figure 3.31: **a).** The set of nodes extracted from the reference image. **b).** The set of nodes extracted from the query image.

The rotational invariance of the correspondence framework was investigated further, as follows. The reference image consisted of two concentric ellipsoids centered in the image, and the query image consisted of two concentric ellipsoids centered in the image and rotated 90° about the z-axis. The same ellipsoid parameters described in the beginning of this chapter were used, and these reference and query images are shown in Figs. 3.18a and 3.18c, respectively. To establish ground-truth the set of query image nodes was rotated by 90° and compared to the location of the reference image nodes. A query node within the same proximity (nearest node to within 10 voxels) of a reference node was chosen manually and reordered to have the same corresponding index as the reference node. Unlike previous experiments, the correspondence matrices would not have been expected to favor the diagonal since the node lists were not identical. By reordering the query nodes, correct correspondences could be expected to fall on the diagonal of a given correspondence matrix. Nodes without correspondence were removed from the data set leaving a total of 31 nodes to be examined. The sets of medial nodes for both images are shown in Fig. 3.31.

Singlet correspondences were calculated via the unary metric with $\mu_T = 0.10$. In Fig. 3.32a, a slab-like node between the ellipsoids is selected (yellow) and the corresponding

nodes are highlighted in Fig. 3.32b (yellow). In Fig. 3.32c, a cylindrical node within the inner ellipsoid is selected (yellow) and the corresponding nodes are highlighted in 3.32d. As expected, by qualitative assessment, multiple nodes of similar dimensionality and scale are highlighted.

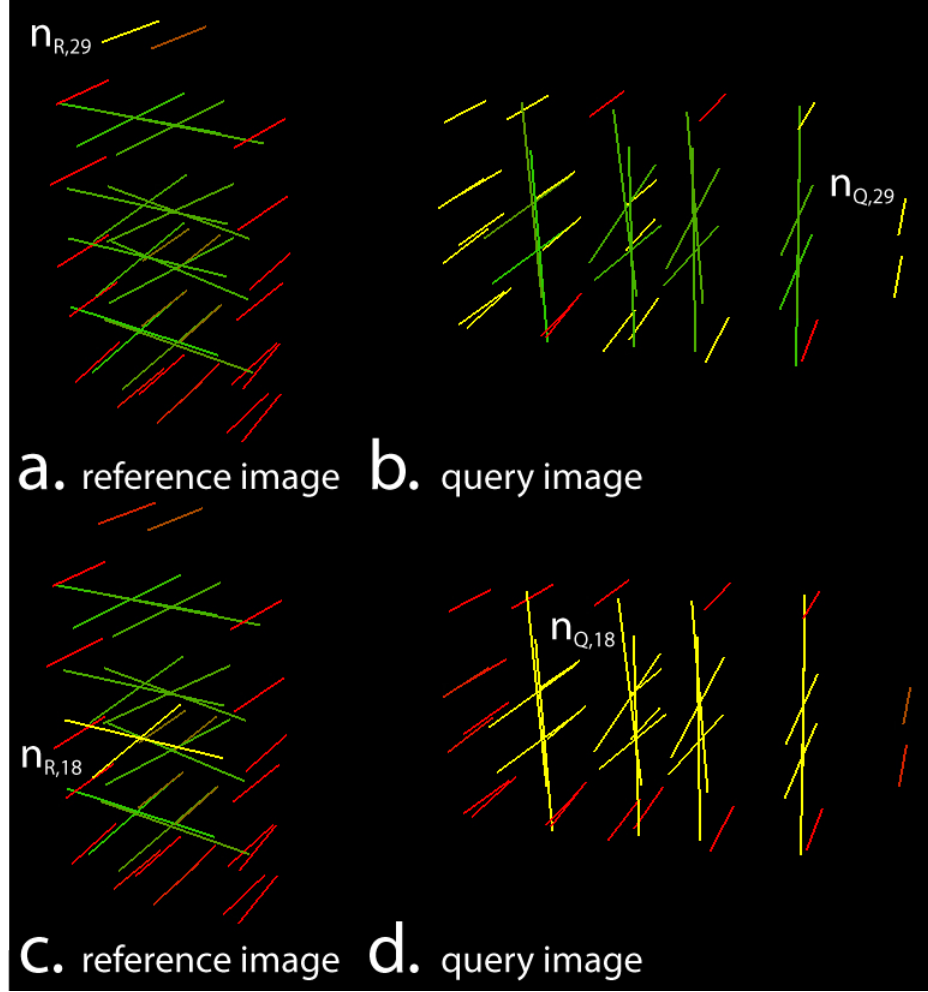


Figure 3.32: Two examples of unary correspondences between nodes extracted from images containing two concentric ellipsoids, one of which is rotated. **a).** A slab-like node in the shell of the reference image is selected (yellow) and **b).** the corresponding nodes in the query image are highlighted. **c).** A cylindrical node within the inner ellipsoid of reference image is selected (yellow) and **d).** the corresponding nodes in the query image are highlighted

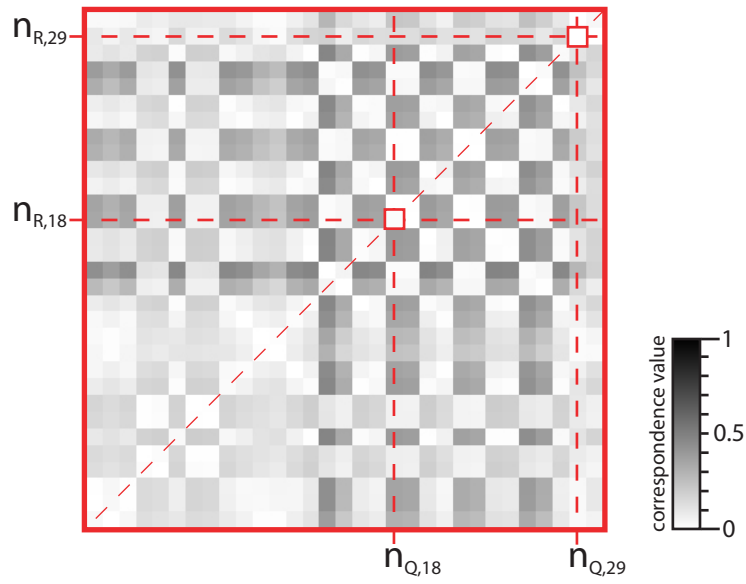


Figure 3.33: Unary correspondence matrix calculated from images containing two concentric ellipsoids, one of which is rotated. White indicates perfect correspondence. Nodes along the diagonal dashed-line are known correspondences. The red squares indicate the correct correspondences for the nodes selected in Figs. 3.32a. and 3.32c.

The unary correspondence matrix is shown in Fig. 3.33. The selected node in Fig. 3.32a is labeled as $n_{R,29}$. The correspondence value at each element k in this row is equal to the unary metric evaluated between node $n_{R,29}$ and node k from the query image. White indicates a perfect correspondence value of 0. The white off-diagonal elements indicate a large number of incorrect correspondences. The highlighted nodes in Fig. 3.32b correspond to those elements in the $n_{R,29}$ row with $\mathbf{C}_\mu(29, k) \leq 0.10$. The known correspondence for node $n_{R,29}$ is indicated as the red square along the matrix diagonal (thin, red dashed-line) at $(n_{R,29}, n_{Q,29})$. Similarly, the selected cylindrical node in Fig 3.32c is labeled as $n_{R,18}$ in Fig. 3.33 with its known correspondence as the red square at $(n_{R,18}, n_{Q,18})$.

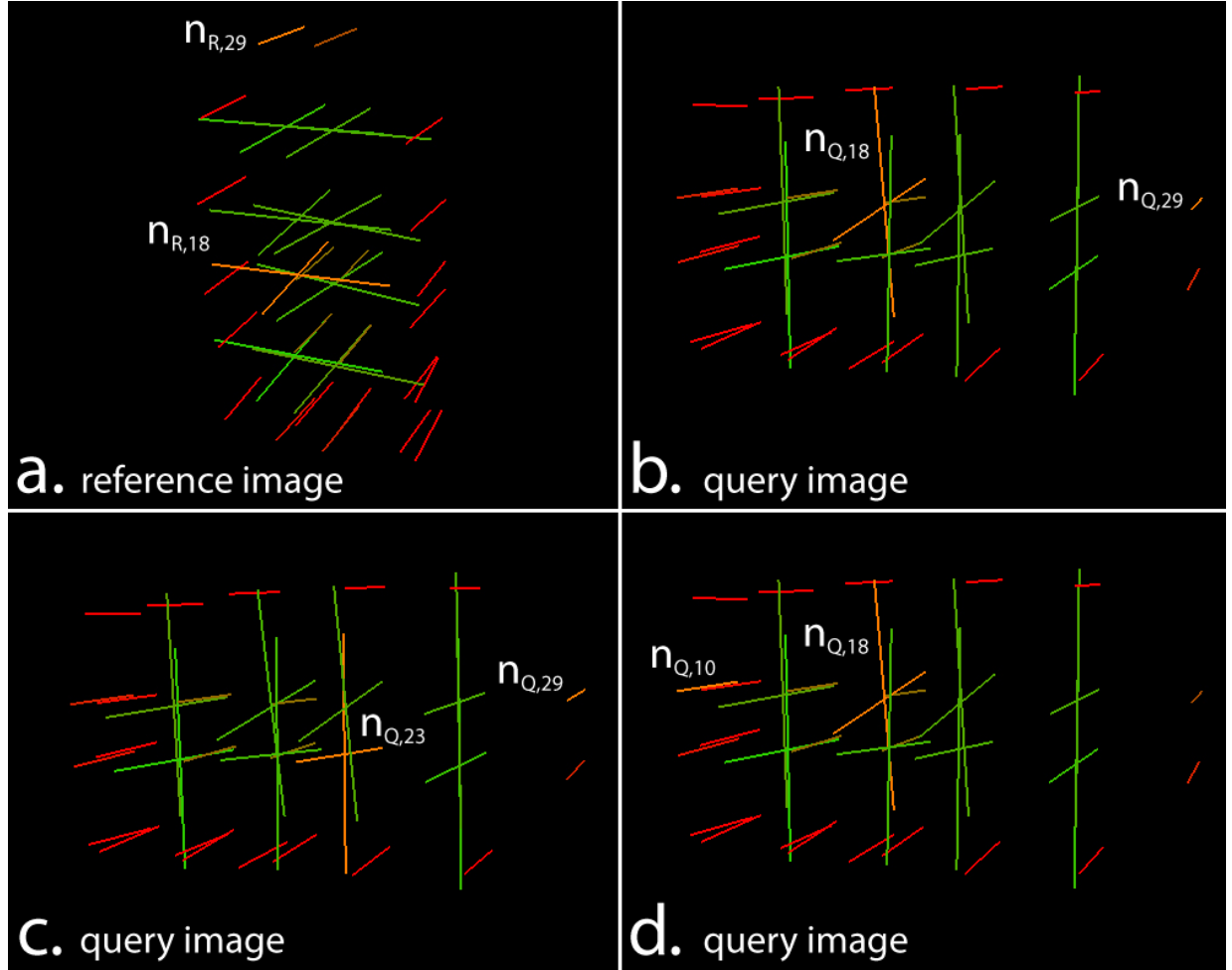


Figure 3.34: Three examples of binary correspondences between doublets extracted from identical images containing concentric ellipsoids. **a).** A doublet consisting of a cylindrical and slab-like node is selected (orange) in the reference image. **b).** The correctly corresponding doublet to that in **a** is highlighted (orange). **c).** and **d).** Two false correspondences to that in **a** are highlighted (orange).

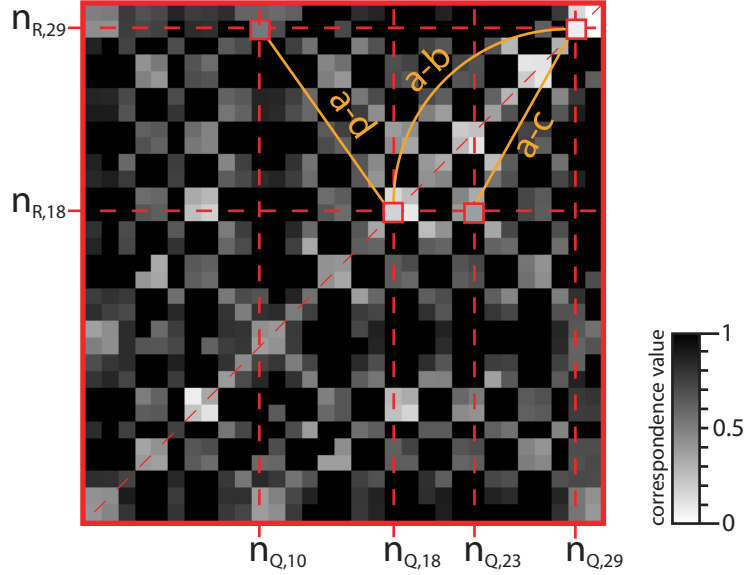


Figure 3.35: Binary correspondence matrix calculated from identical images containing two concentric ellipsoids. Red squares connected by an orange line indicates the binary metric evaluated between a doublet from Fig. 3.34a and a doublet from the query image labeled with orange letters. The letters corresponds to the panes in Fig. 3.34.

Doublet correspondences were calculated via the binary metric with $\beta_T = 0.10$. A selected doublet is shown in Fig. 3.34a and the corresponding doublet is shown in Fig. 3.34b. The binary correspondence matrix is shown in Fig. 3.35. Note that there are less low correspondence values off the diagonal as compared to the unary correspondence matrix. The reference image doublet $(n_{R,18}, n_{R,29})$ and its corresponding doublet $(n_{Q,18}, n_{Q,29})$ is illustrated by the orange line labeled “a-b.” Individually, nodes $n_{R,18}$ and $n_{Q,18}$ correspond, and $n_{R,29}$ and $n_{Q,29}$ correspond. Note that these node pairs are on the diagonal, which indicates that they are indeed true correspondences.

A doublet consisting of a cylindrical and slab-like node is selected in Fig. 3.34a, and a potentially corresponding doublet is highlighted in Fig. 3.34c. By observing the location of the cylindrical nodes relative to each other, it is clear that this correspondence is incorrect. However, since the nodes within the doublets, the doublet distance, and the doublet orientation were sufficiently similar, the doublet correspondence criteria was satisfied. This incorrect correspondence can be observed in the correspondence matrix as the orange line labeled “a-c”. The slab-like nodes $n_{R,29}$ and $n_{Q,29}$ are on the diagonal, but the cylindrical

nodes $n_{R,18}$ and $n_{Q,23}$ are off-diagonal. Since the indices were reordered to maintain actual correspondence, the differing indices for the cylindrical nodes also indicate incorrect correspondence. Another example of an incorrect correspondence is shown in Fig. 3.34d. The pair of doublets are labeled “a-d” in the binary correspondence matrix. The cylindrical nodes $n_{R,18}$ and $n_{Q,18}$ are on the diagonal, but the slab-like nodes $n_{R,29}$ and $n_{Q,10}$ are off-diagonal.

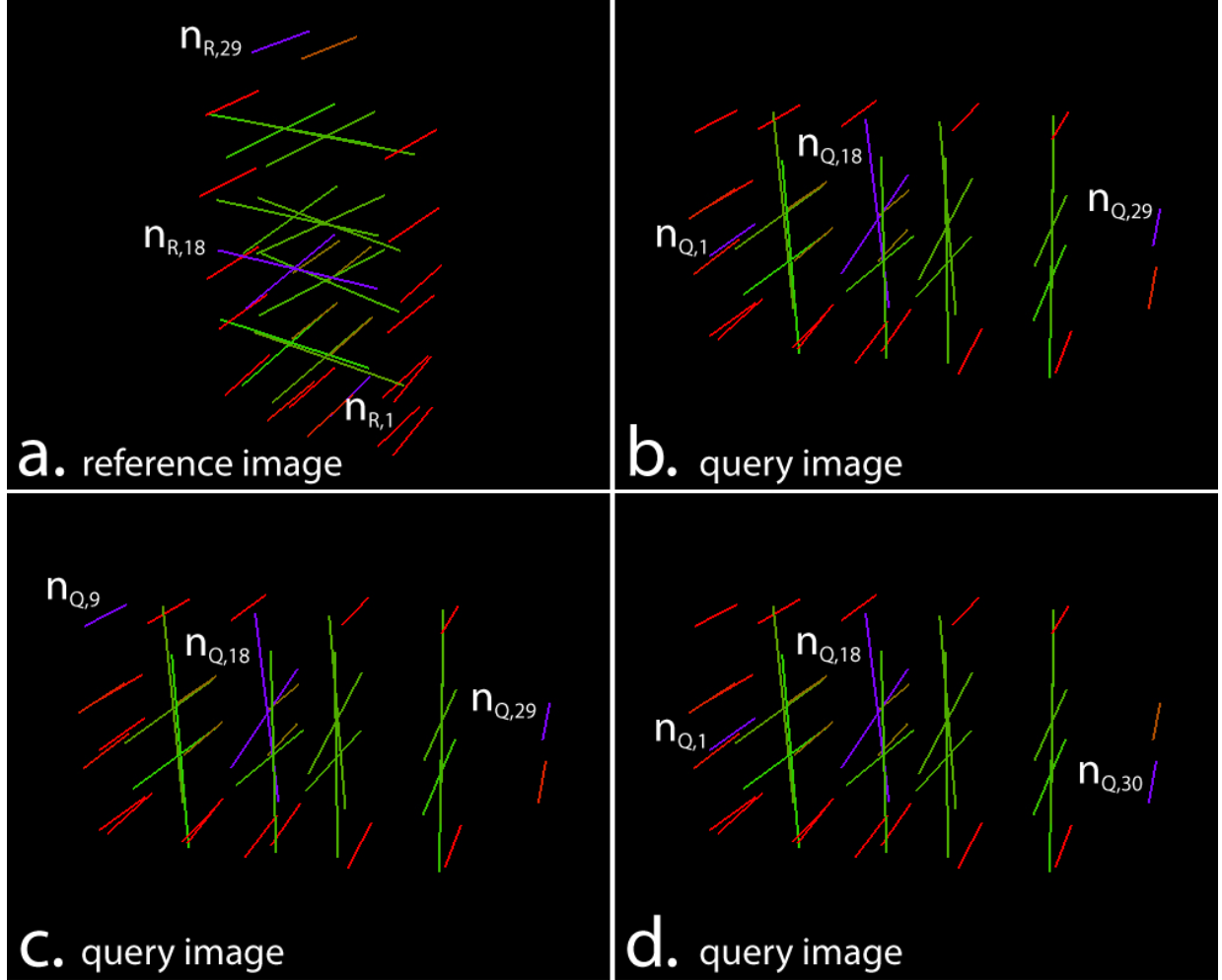


Figure 3.36: Two examples of trinary correspondences between triplets extracted from identical images containing concentric ellipsoids. **a).** A selected triplet (purple) in the reference image consisting of 2 slab-like nodes and a cylindrical node. **b).** The correctly corresponding triplet in the rotated query image is highlighted (purple). **c).** An incorrectly corresponding triplet (purple) consisting of 2 slab-like nodes and a cylindrical node. **d).** Another incorrectly corresponding triplet in the query image (purple).

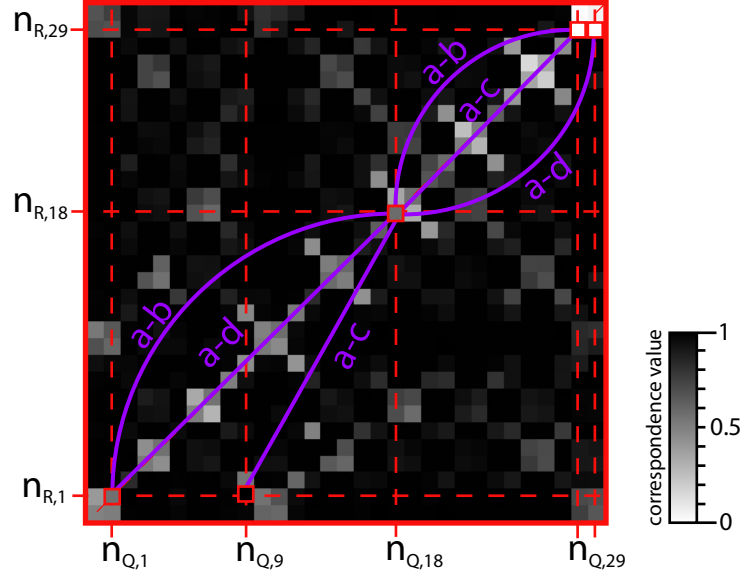


Figure 3.37: Trinary correspondence matrix calculated from identical images containing two concentric ellipsoids. Red squares connected by a purple line indicate the trinary metric evaluated between the triplet from the reference image in Fig. 3.26a and various triplets from the query image as identified by “a-b”, “a-c”, and “a-d”, corresponding to the triplets in Figs. 3.26a, b and c.

Triplet correspondences were calculated via the trinary metric with $\tau_T = 0.10$. Shown in Fig. 3.36a is a selected triplet (purple) in the reference image consisting of 2 slab-like nodes and 1 cylindrical node. A correctly corresponding triplet in the query image is highlighted in Fig. 3.36b. This correspondence is shown in the trinary correspondence matrix (Fig. 3.37) as 3 red squares connected by the purple line labeled “a-b.” Since all 3 nodes are on the diagonal, the indices of the individual nodes are the same and thus it is known that this is a correct correspondence. An incorrect triplet correspondence is shown in Figs. 3.36c and 3.36d. In terms of the dimensionality and scale of the individual nodes, the triplets appear similar, but clearly do not correspond based on relative spatial configuration of the nodes within each triplet. The fact that this particular triplet correspondence is incorrect is verified by observing the triplet correspondence matrix in Fig. 3.37. The triplet correspondence is shown as red squares connected by the purple line labeled “a-d.” Note that none of the nodes are exactly on the matrix diagonal. Moreover, the correspondence values are very high as highlighted by the red squares.

Each of these “experimental” correspondence matrices are shown side-by-side in Fig. 3.38c. From left to right, are the unary, binary, and trinary correspondence matrices. With each successive matrix, correspondence values off the diagonal decrease, indicating that incorrect correspondences are being eliminated.

Shown in Figs. 3.38a and 3.38b are two sets of correspondence matrices computed for comparison with the above results while introducing rotation. First, to establish “ideal” correspondence matrices, both the reference and query images were rotated by 90° about the z-axis. Therefore, if the correspondence metrics are truly independent of rotation, the matrices shown in Fig. 3.38c should be identical to those computed for these images (Fig. 3.38a). Second, to establish a “theoretical” set of nodes that should be extracted from the rotated query image, the reference nodes were manually rotated and used as the query nodes. Correspondence matrices computed from this theoretical set of nodes are shown in Fig. 3.38b. As expected, these matrices are nearly identical to those in Fig. 3.38a since the relative geometric relationship between nodes is independent of global rotation. However, in comparing the “experimental” and “theoretical” results, there is a noticeable difference between the matrices. The differences between the matrices can be attributed to sampling artifact when extracting medial nodes, notably deviations in node properties and node location. The absolute difference in scale and dimensionality, and the magnitude of location difference between the “theoretical” and “experimental” nodes are tabulated in Table 3.3.

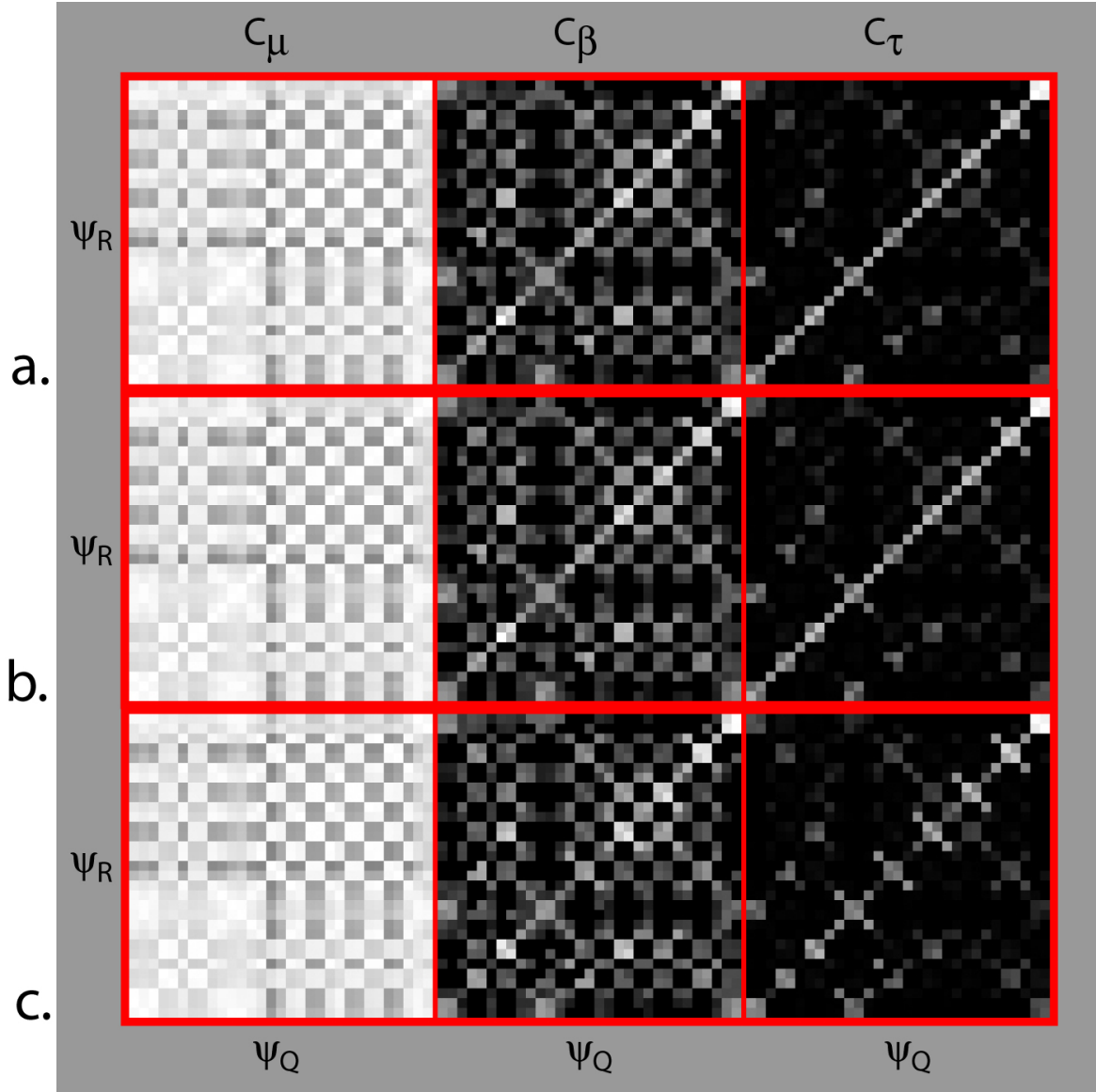


Figure 3.38: Correspondence matrices calculated to demonstrate the effects of global image rotation on the correspondence metrics. From left to right, are the unary, binary, and trinary correspondence matrices. **a).** Calculated correspondence matrices when using the reference image as the query image. **b).** Correspondence matrices calculated with query nodes rotated by 90° about the z-axis. **c).** Correspondence matrices calculated with query nodes extracted from the rotated concentric ellipsoids image.

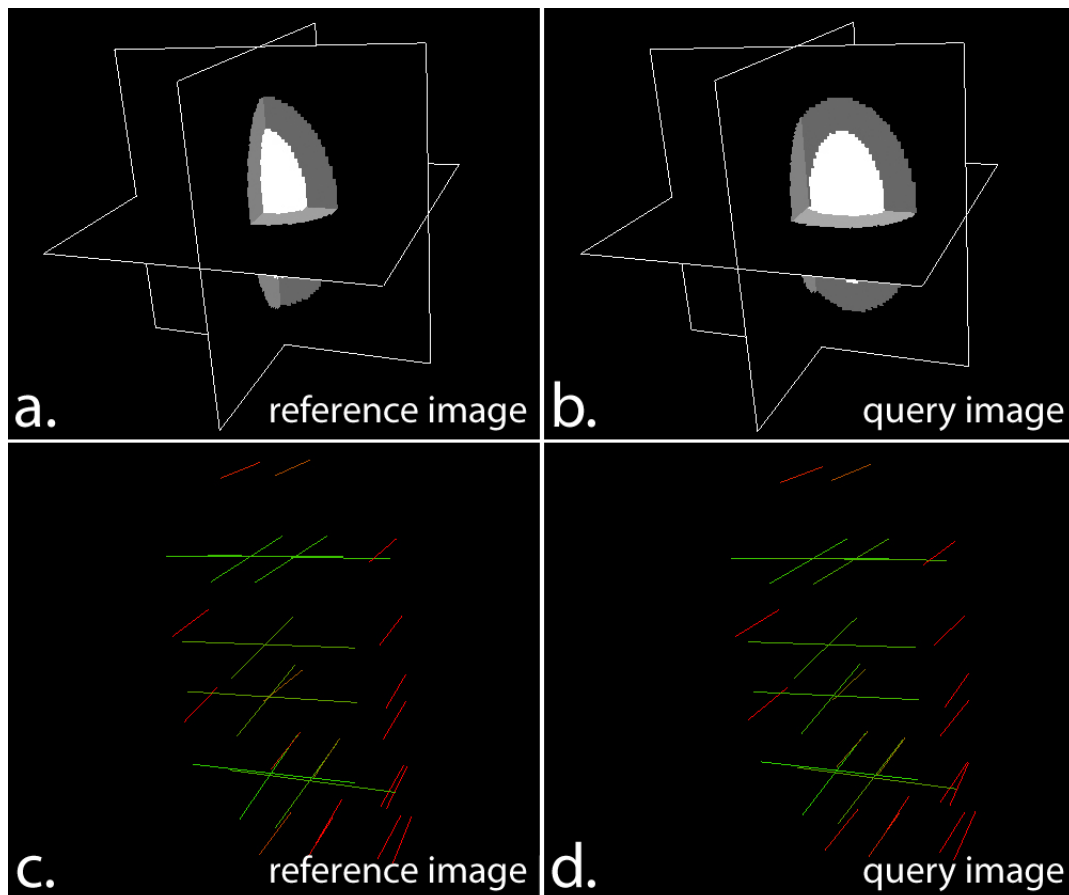


Figure 3.39: a.) Reference image containing two concentric ellipsoids centered in the image. b.) Query image containing two concentric ellipsoids translated along the x-axis from the center of the image. The set of nodes extracted from the c.) reference image and d.) query image.

The translational invariance of the correspondence method is now investigated. The reference image consisted of two image-centered concentric ellipsoids. The query image consisted of two concentric ellipsoids shifted along the x-axis by +10 voxels. The same ellipsoid parameters described in the beginning of this chapter were used. These reference and query images are shown in Figs. 3.18a and 3.18b, respectively. Similar to the previous experiment with rotation, ground-truth was determined by manually shifting the nodes by +10 voxels, comparing them to the reference image nodes, and reordering them in the query node list to have the same index as the corresponding nodes in the reference image. Nodes without a correspondence were removed from the node lists leaving a total of 24 nodes to be examined. The sets of medial nodes extracted from each image are shown in Fig. 3.39.

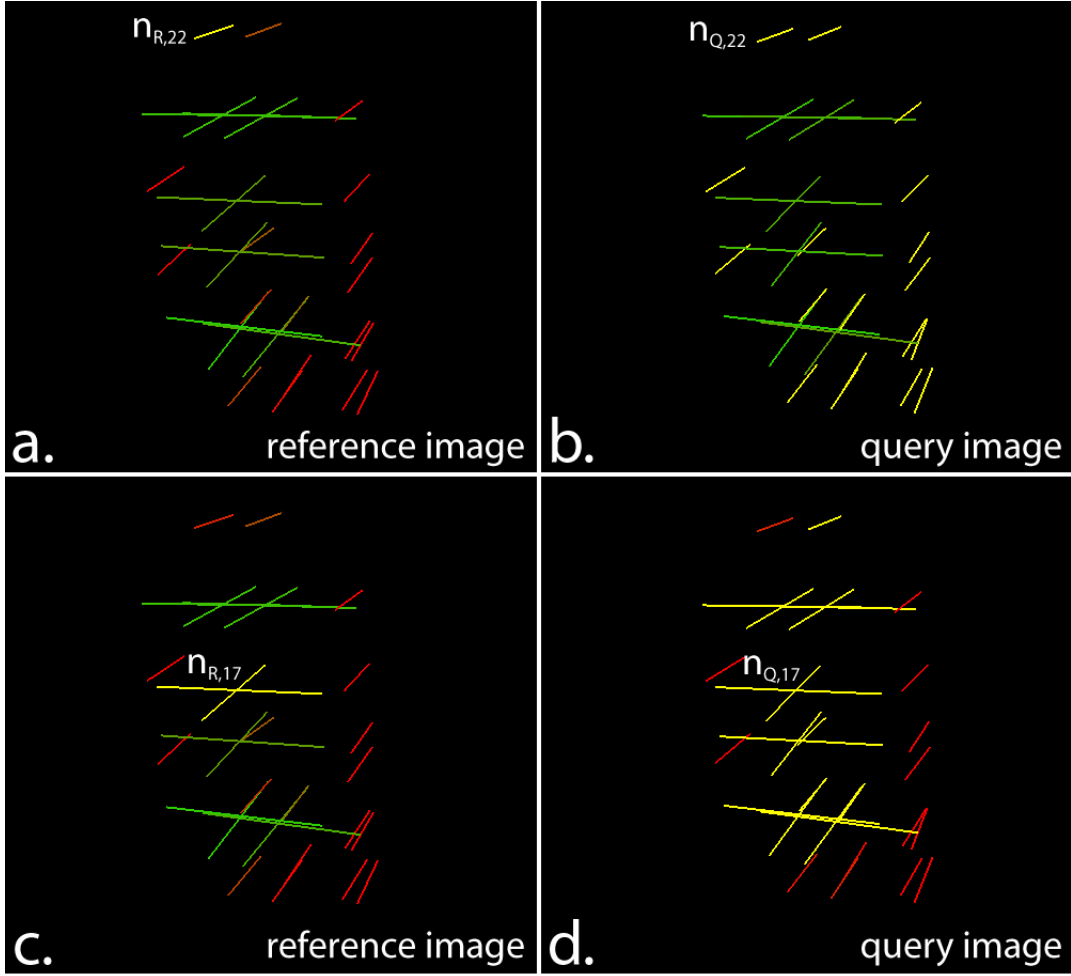


Figure 3.40: Two examples of unary correspondences between nodes extracted from images that contain concentric ellipsoids, which differ by translation. **a).** A slab-like node in the shell of image R is selected (yellow) and the corresponding nodes in image Q are highlighted in **b.** **c).** A cylindrical node within the inner ellipsoid of image R is selected (yellow) and the corresponding nodes in image Q are highlighted in **d.**

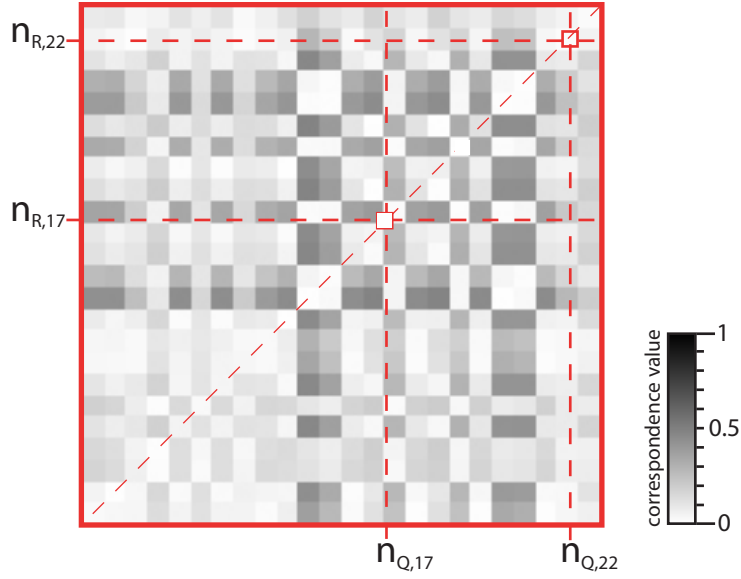


Figure 3.41: Unary correspondence matrix calculated for nodes extracted from images that contain concentric ellipsoids, which differ by translation. White indicates perfect correspondence. Nodes along the diagonal dashed-line are known correspondences. The red squares indicate the correct correspondences for the nodes selected in Figs. 3.40a. and 3.40c.

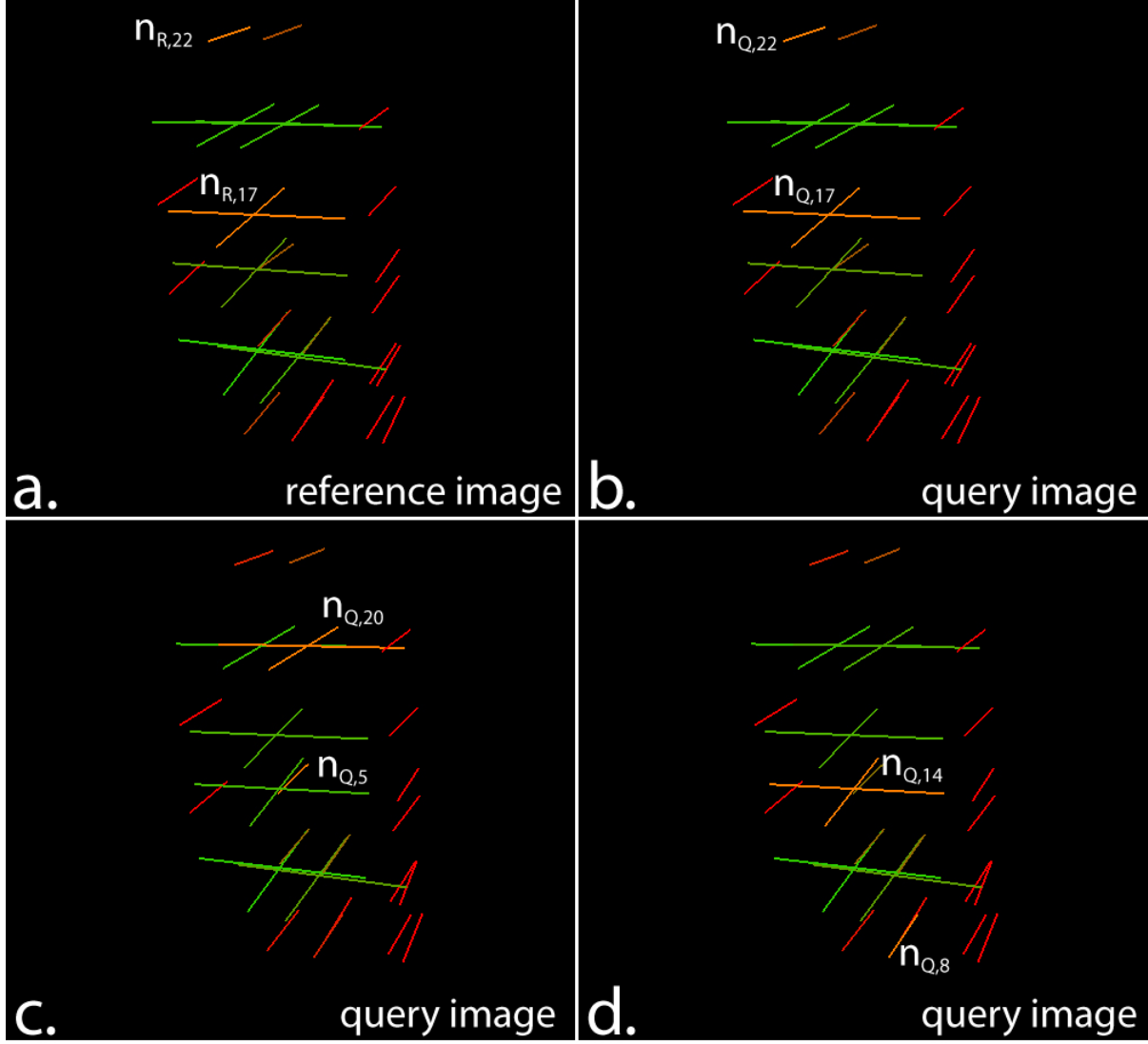


Figure 3.42: Three examples of binary correspondences between doublets extracted from images containing two concentric ellipsoids for demonstrating translational invariance. **a).** A doublet consisting of a cylindrical and slab-like node is selected (orange) in the reference image. **b).** The correctly corresponding doublet is highlighted (orange). **c).** and **d).** Two incorrect correspondences are highlighted (orange).

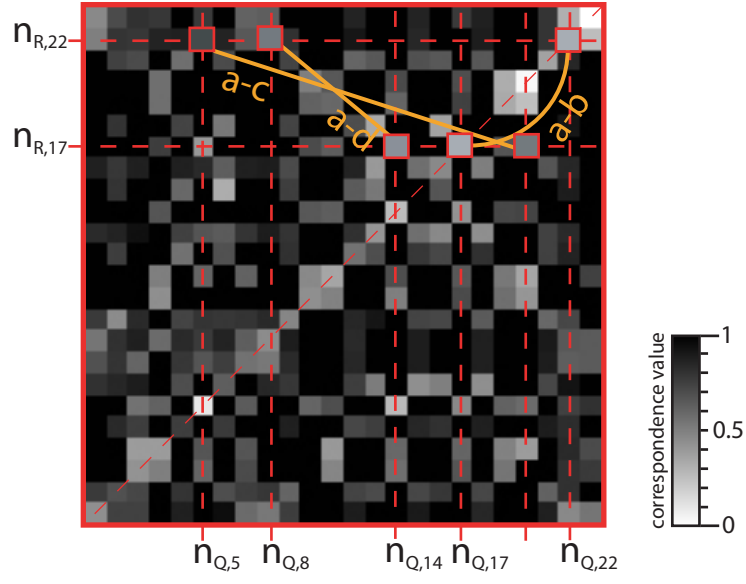


Figure 3.43: Binary correspondence matrix calculated from images containing two concentric ellipsoids for demonstrating translational invariance. Red squares connected by an orange line indicates the binary metric evaluated between a pair of doublets from Fig. 3.42.

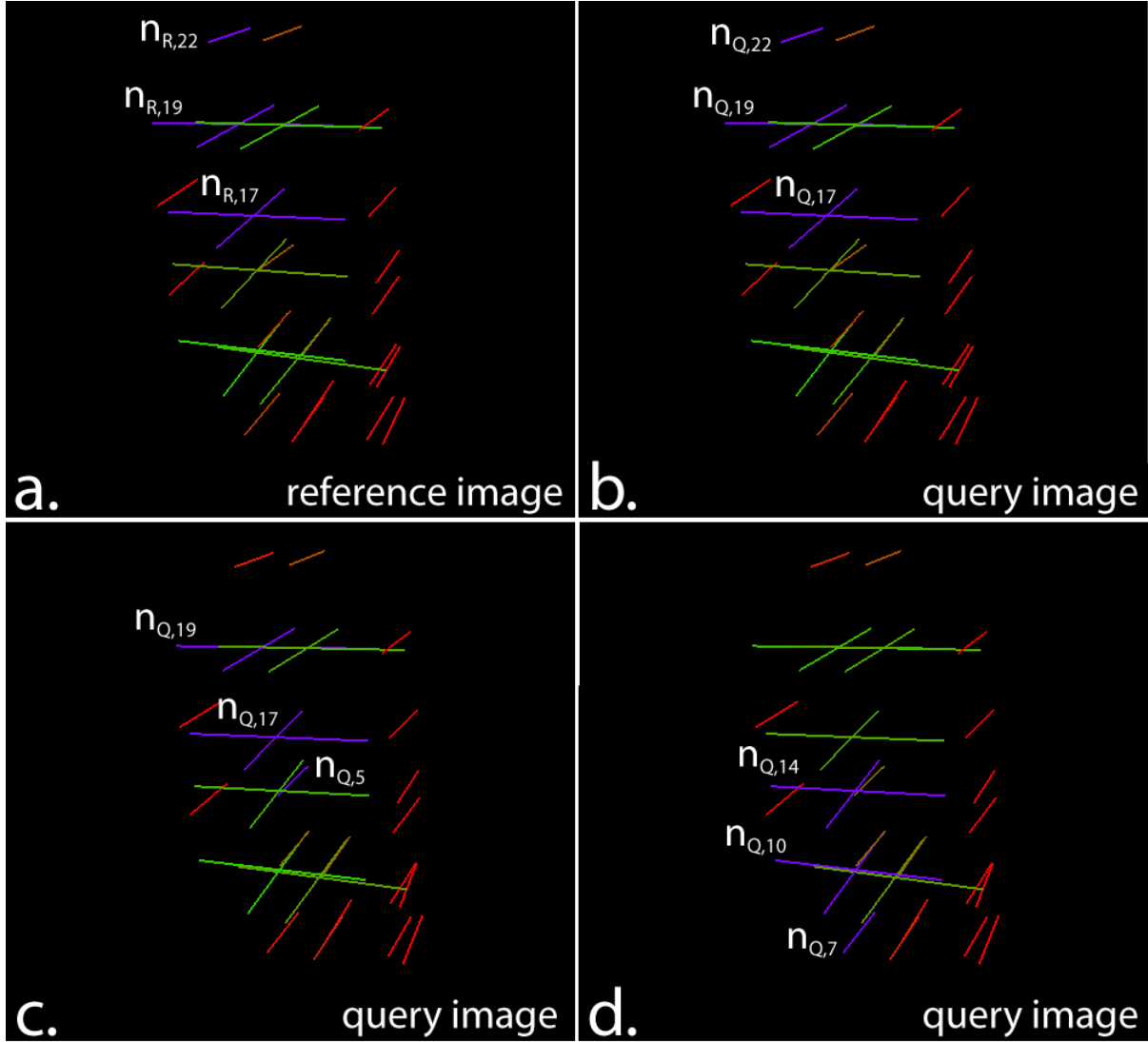


Figure 3.44: Two examples of trinary correspondences between triplets extracted from two images containing concentric ellipsoids, one of which is translated. **a).** A selected triplet (purple) consisting of 2 cylindrical nodes and a slab-like node and **b).** the correctly corresponding triplet. **c).** and **d).** Two incorrectly corresponding triplets are highlighted (purple). (purple).

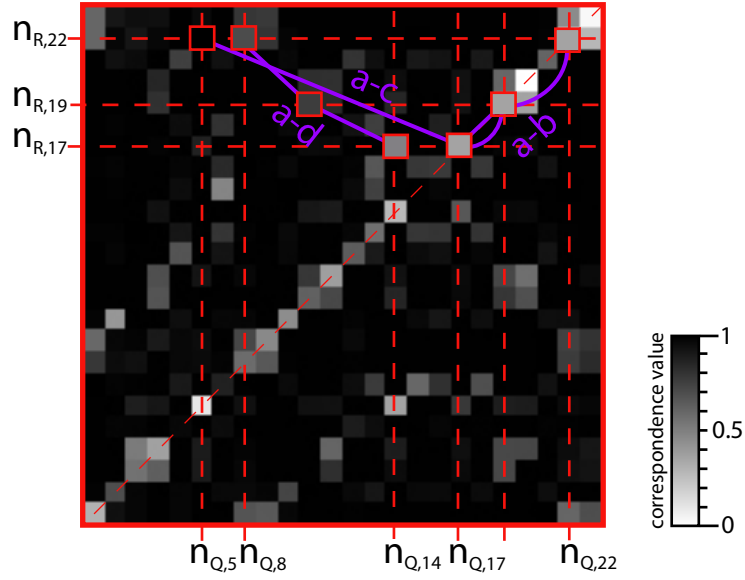


Figure 3.45: Trinary correspondence matrix calculated from two images containing concentric ellipsoids, one of which is translated. Red squares connected by a labeled purple line indicate the trinary metric evaluated for triplets from Fig. 3.44. The first letter refers to the reference image and the second letter refers to the query image in Fig. 3.44.

Unary, binary, and trinary correspondences were determined with $\mu_T = \beta_T = \tau_T = 0.10$. Two examples of unary correspondences are shown in Fig. 3.40. As before, the correct correspondence is highlighted among other nodes of sufficient similarity. The unary correspondence matrix is shown in Fig. 3.41 for which the correct correspondences of the selected nodes in Fig. 3.40 are highlighted with red squares.

Several examples of binary correspondences are shown in Fig. 3.42. A selected doublet from the reference image is shown in Fig. 3.42a with the correct correspondence in Fig. 3.42b. Two incorrect correspondences, with both nodes of the doublets being incorrect, are shown in Figs. 3.42c and d. These doublet correspondences are shown as labeled orange lines in the binary correspondence matrix (Fig. 3.43). Likewise, examples of trinary correspondences are illustrated in Fig. 3.44 and denoted as labeled purple lines in the trinary correspondence matrix (Fig. 3.45).

As can be seen in the binary and trinary correspondence matrices, some false correspondences are removed after the matrix is normalized. For example in the binary correspondence matrix, the pair of doublets indicated by the line labeled $a-c$ is shown in Figs. 3.42a and 3.42c. The doublets were said to correspond because the binary metric was less than the threshold. However, after the correspondence values of the constituent nodes are normalized by the maximum correspondence accumulated by doublets, the nodes no longer correspond. This is best illustrated by the high correspondence value between nodes $n_{R,22}$ and $n_{Q,5}$, i.e., $C_\beta(n_{R,22}, n_{Q,5})$ as shown in Fig. 3.43.

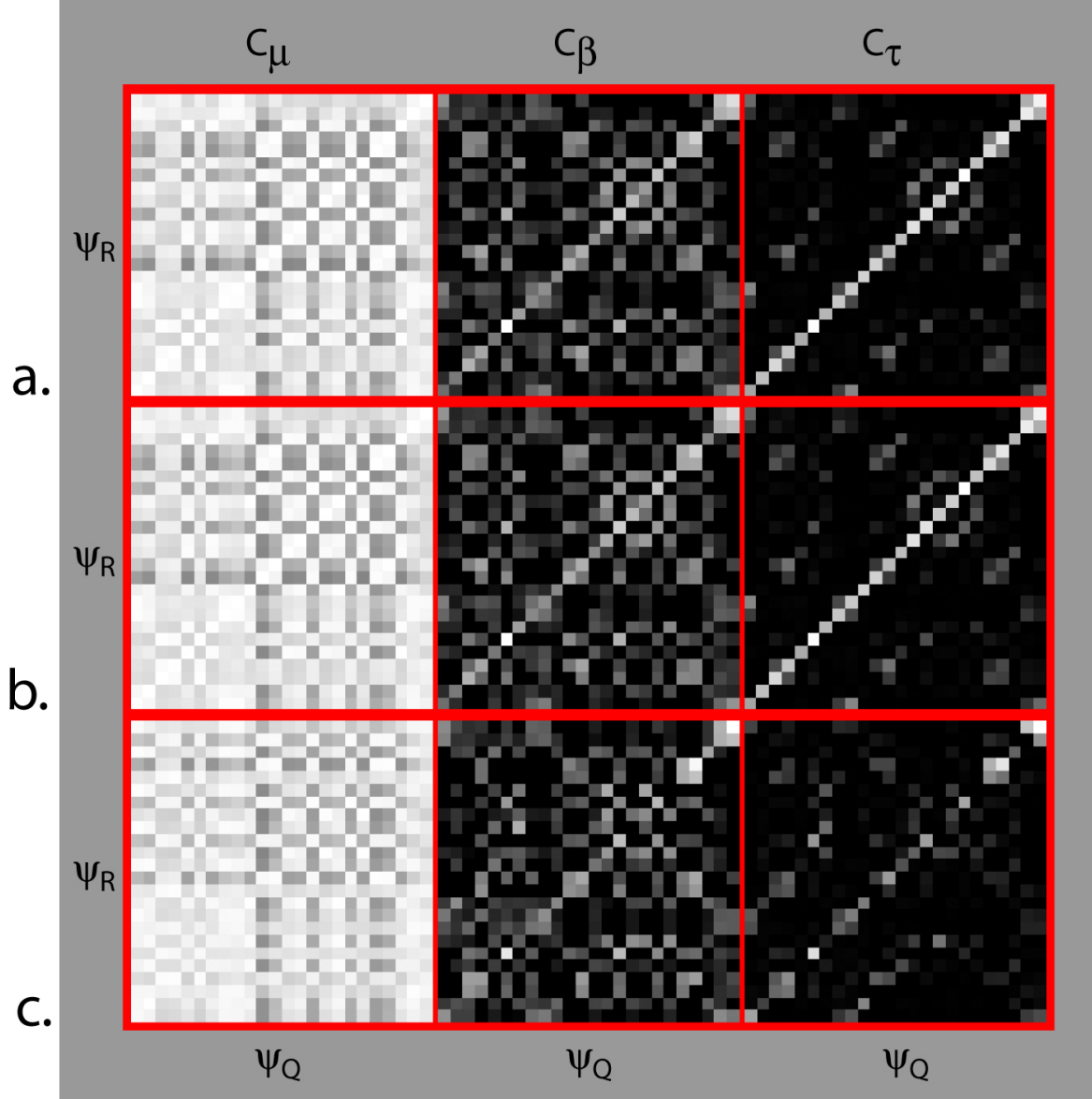


Figure 3.46: Correspondence matrices calculated from the translated concentric ellipsoids data set. From left to right, are the unary, binary, and trinary correspondence matrices. **a).** Calculated correspondence matrices when using the reference nodes as the query nodes. **b).** Correspondence matrices calculated for query nodes produced by translating the reference nodes by 10 voxels along the x-axis. **c).** Matrices calculated for query nodes extracted from translated concentric ellipsoid image.

For comparison, two sets of correspondence matrices were calculated to assess the effect of translation on the correspondence framework. First, the reference nodes were also used as the query nodes, and correspondences were determined to establish the “ideal” correspondence matrices for these nodes. These correspondence matrices are shown in Fig. 3.46a. As expected from the first experiment in this section, there are strong correspondences on the diagonal of the trinary correspondence matrix with sparse cross-correspondences.

Correspondences were then determined after manually shifting the query nodes by +10 voxels along the x-axis. Since the properties of the query nodes are left unchanged, it was expected that any differences in correspondence values can be attributed to the translation. The correspondence matrices calculated from these nodes are shown in Fig. 3.46b, and as expected, are identical to those in Fig. 3.46a since the relative geometric relationship between the nodes is independent of global translation. Shown in Fig. 3.46c are the correspondence matrices calculated from the translated concentric ellipsoids data set. Similar to the results of the rotation experiment, differences between the “ideal” and “experimental” matrices can be attributed to differences in node properties and location as a result of the sampling artifact discussed in Section 2.3.1. Shown in Figs. ??a - ??c are graphs of the absolute difference in scale and dimensionality between corresponding nodes. The magnitude of the location difference between the theoretical and experimental node sets is graphed in Fig. ??d as a function of node index. From these graphs, it appears that fairly large values for scale, λ_2 , and magnitude of location difference account for the small number of correct correspondences.

Rotation by 90° had less of an impact on determining correspondences than translation, which can be observed by comparing the correspondence matrices. The average and standard deviation was calculated for each of the various difference measures from the translated and rotated data (Table 3.3). The largest disparity between the data sets was for scale and λ_2 for which the translated set had a higher average and standard deviation. It is noted that rotation by angles other than 90° and translation by values other than +10 voxels may have produced quite different results.

Table 3.3: Average and standard deviation calculations of difference measures between corresponding nodes for validation in the presence of rotation by 90° and translation.

| Rotation | | |
|-------------------------------|-------------------------|------------------------------------|
| Difference Measure | Average (voxels) | Standard Deviation (voxels) |
| Scale difference | 1.023 | 1.056 |
| λ_1 difference | 0.000344 | 0.000403 |
| λ_2 difference | 0.0194 | 0.0279 |
| Location difference magnitude | 1.332 | 0.356 |

| Translation | | |
|-------------------------------|-------------------------|------------------------------------|
| Difference Measure | Average (voxels) | Standard Deviation (voxels) |
| Scale difference | 1.936 | 2.271 |
| λ_1 difference | 0.000643 | 0.000592 |
| λ_2 difference | 0.0483 | 0.0681 |
| Location difference magnitude | 1.354 | 0.656 |

3.6.2 Measuring Correspondence Sensitivity and Specificity

In this section, correspondence sensitivity and specificity are quantified via ROC curve analysis. A “normal brain” image obtained from the BrainWeb simulated brain database (http://www.bic.mni.mcgill.ca/brainweb/selection_normal.html) was used. Since the brain contains many structures of varying size, it was expected that a large number of differing nodes would be extracted. The following parameters were selected for the simulation:

- Modality (pulse sequence): T1
- Slice thickness (in-plane spacing is $1\text{mm} \times 1\text{mm}$): 1mm
- Noise (relative to brightest tissue): 3%
- Intensity non-uniformity: 20%

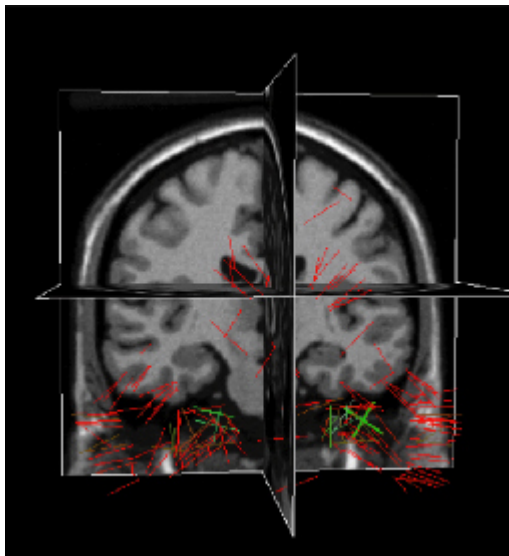


Figure 3.47: Image obtained from the BrainWeb simulated brain database with medial nodes overlayed. Cylindrical are located in the eyeballs and slab-like nodes are located throughout the image such as in the skull and lateral ventricles.

The image was 8-bit and $181 \times 217 \times 181$ in size. Various cross-sections of the image are shown in Figs. 3.19 and 3.47. The same image was used as both the reference and query images to easily establish ground-truth as discussed in Section 3.5. Core atoms were formed with identical parameter settings for each image and clustered into 105 medial nodes. These nodes are overlayed on the image data in Fig. 3.47. Cylindrical nodes (2 intersecting green

lines) make the eyeballs identifiable, since the approximately spherical nature of eyeballs is detected as cylindricalness. Slab-like nodes can be seen throughout the image such as in the skull and lateral ventricles.

To generate an ROC curve for a given correspondence metric, correspondences were determined for 30 different values of C_T (the threshold on calculated correspondence values). The values for C_T were chosen at evenly spaced intervals between 0 and 1. A value of 0.5 was arbitrarily chosen for D_T (the ground-truth threshold). Correspondences were then classified, and the True Positive Fraction and False Positive Fraction were computed to generate the ROC curve for the given correspondence metric.

The calculated ROC curves for the unary, binary, and trinary metrics are shown on the same graph in Fig. 3.48. The resulting ROC curves reflect the expectation of the unary metric performing better than randomly guessing, the binary metric being more specific than the unary metric, and the trinary metric determining the most correct correspondences. By using the trapezoidal rule, A_{ROC} was calculated for each correspondence test. Results are presented below in Table 3.4, demonstrating the expected increase in accuracy for each successive metric.

Subsequent chapters in this dissertation will develop further methods for assessing correspondence performance.

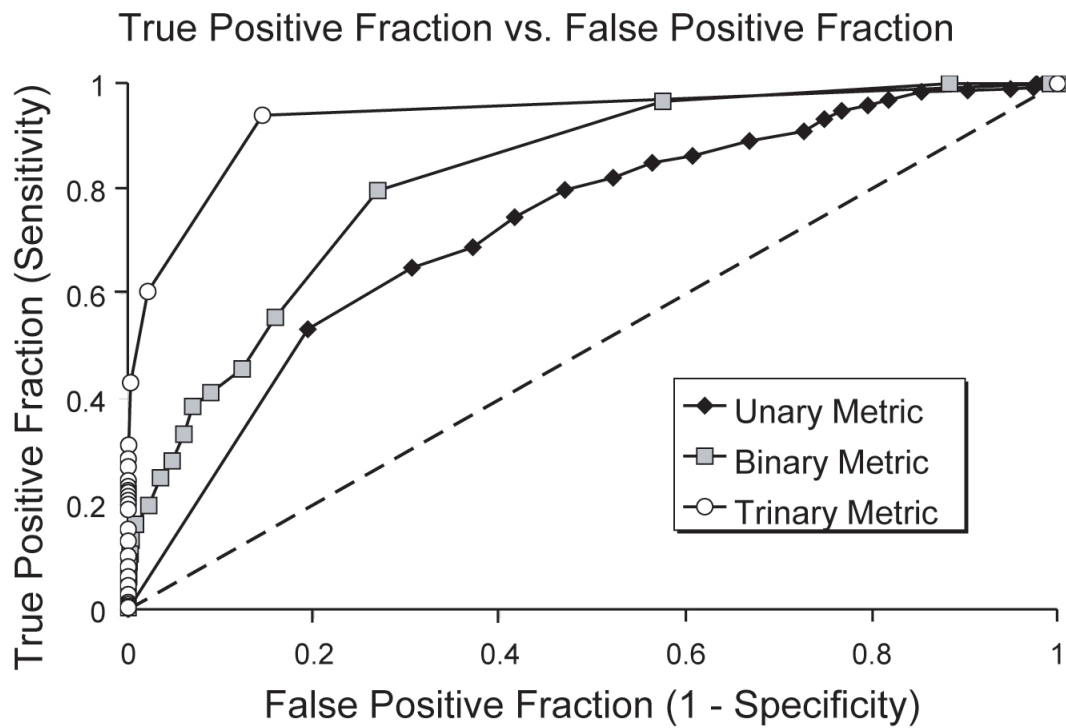


Figure 3.48: Calculated ROC curves for each correspondence metric.

Table 3.4: Calculated A_{ROC} for each correspondence metric.

| | Area Under the ROC Curve |
|----------------|--------------------------|
| Unary Metric | 0.72 |
| Binary Metric | 0.82 |
| Trinary Metric | 0.94 |

4.0 IMPLIED TRANSFORMS TO IMPROVE CORRESPONDENCES

In this chapter, a method to improve upon the correspondence framework is developed. The trinary metric was shown, with ROC curve analysis in Section 3.6.2, to reduce the number of false correspondences that the unary or binary metric might find. Although preliminary results are encouraging, the trinary metric still tends to result in a number of incorrect triplet correspondences. Further reduction in the number of false correspondences may be accomplished by introducing a global analysis of the geometric relationship between nodes via transforms implied by triplet correspondences. It is common in image registration to compute an optimum transform between a local set of features. Theoretically, an optimal transform would maximize the number of true correspondences (true positives) and minimize the number of false correspondences (false positives) between the images. Such methods often assume a *rigid transform* for the entire feature set between images. A rigid transform involves only translation and rotation, preserving the distance between any two points. For many applications, a rigid transformation is sufficient for describing the spatial relationship between feature sets. For example, features extracted from intra-subject brain images may be aligned with a rigid transformation given that the brain’s motion is minimal. However, in some applications *non-rigid transforms* are required to accommodate for structure deformation.

Each triplet correspondence implies a transform that maps triplets in the query image to triplets in the reference image as shown in Fig. 4.1. To better accommodate local structure deformation, *affine transforms* are computed. An affine transform applies translation, rotation, and anisotropic scale, preserving collinearity, but not necessarily angle or distance. The affine transform is chosen because, in a 2D space, any triangle (triplet) can be perfectly transformed into any other triangle (triplet). However in 3D, computing an affine transform

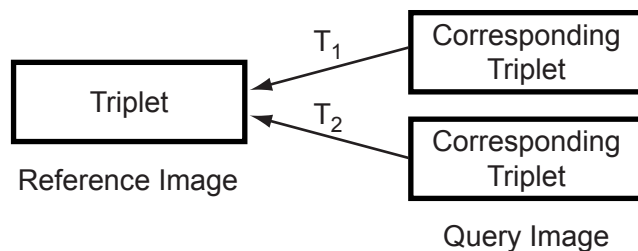


Figure 4.1: Shown are the implied transforms between a triplet in the reference image and potentially corresponding triplets in the query image.

requires 4 points. Rather than extending cliques to include 4 points a method to extrapolate a triplet to a tetrahedron is developed.

Affine transforms are computed between corresponding tetrahedra, mapping them from image Q to image R. These affine transforms are applied to every tetrahedron in image Q, and the transformed tetrahedrons are then compared to the respective corresponding tetrahedra in image R. A *correspondence likelihood* is computed for each correspondence based on how well a given transform fits. The correspondence likelihood is essentially a cumulative vote for correspondence. Tetrahedra accumulate correspondence likelihood for each transform for a final measure of correspondence. The integration of triplet transforms into the correspondence hierarchy is shown in Fig. 4.2. The following sections detail the methods of calculating implied transforms from triplet correspondences, extrapolating triplets to tetrahedra, and calculating the correspondence likelihood as a measure of correspondence.

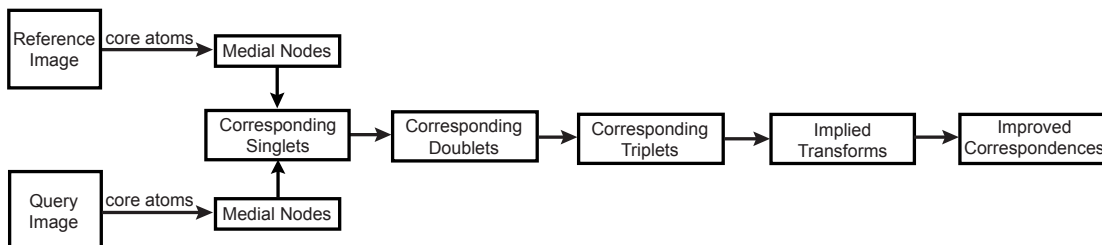


Figure 4.2: For each set of corresponding triplets, a transform is implied that maps a triplet from image Q to its corresponding triplet in image R. The transform, once computed, can be applied to other triplets to improve their accuracy.

4.1 COMPUTING IMPLIED 3D AFFINE TRANSFORMS

The 3D affine transform is formulated by first considering the transform between two 3D points. Only 3 degrees of freedom are required to describe such a transformation, which may be construed as translations along each cardinal axis. Let $\mathbf{r}_i = (x_{R,i}, y_{R,i}, z_{R,i})$ be the coordinates of the i^{th} node in image R and $\mathbf{q}_j = (x_{Q,j}, y_{Q,j}, z_{Q,j})$ be the coordinates of the corresponding j^{th} node in image Q. By using homogeneous coordinates, point \mathbf{q}_j can be mapped to point \mathbf{r}_i in the coordinate system of image R by

$$\mathbf{R}_i = \mathbf{T}\mathbf{Q}_j, \quad (4.1)$$

where $\mathbf{Q}_j = [\mathbf{q}_j, 1]$, $\mathbf{R}_i = [\mathbf{r}_i, 1]$, and \mathbf{T} is the transformation matrix. In matrix representation

$$\begin{bmatrix} x_{R,i} \\ y_{R,i} \\ z_{R,i} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{Q,j} \\ y_{Q,j} \\ z_{Q,j} \\ 1 \end{bmatrix}, \quad (4.2)$$

where t_x , t_y , and t_z are translations along the respective cardinal axes in the coordinate system of image R.

For a 3D affine transform, there are 12 parameters to account for various properties such as translation, rotation, and anisotropic scale. These particular properties have a total of 9 independent modes of variation. A triplet consists of three nodes that comprise the vertices of a triangle called a *triplet triangle*. Aligning two triplet triangles in 3D space requires 8 modes of variation; 2 rotations and 1 translation to make the triangles co-planar and 2 translations, 1 rotation, and 2 scales to make the vertices of the triangles coincident. Scale in the direction orthogonal to the triangle plane is not accounted for making it impossible to compute a 3D affine transform between triplets since it is an under-constrained problem. Utilizing quadruplets of nodes can account for the remaining mode of variation. However, it is unlikely that a sufficient number of similar cliques containing quadruplets of nodes would exist to determine correspondences. The computational requirements would also be huge because of the increased combinatorial complexity, i.e., $\binom{m}{4}$, where m is the number of

nodes. Instead, triplet triangles are extended to tetrahedrons to account for the remaining mode of variation, making the transform solvable.

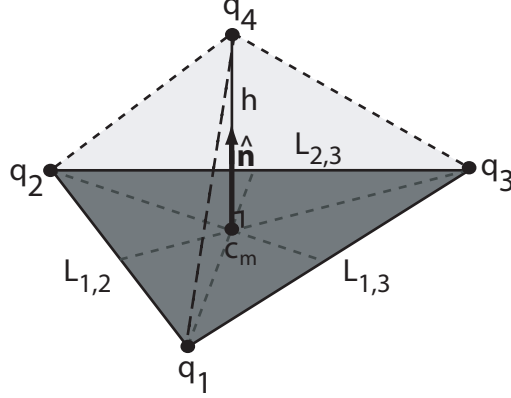


Figure 4.3: A triplet tetrahedron formed from an equilateral triplet triangle consisting of nodes \mathbf{q}_1 , \mathbf{q}_2 , and \mathbf{q}_3 . Black dashed lines indicate the edges of the tetrahedron formed by projecting a vector to \mathbf{q}_4 , orthogonal to the triangle in the direction of $\hat{\mathbf{n}}$ at a distance of h from the center of mass of the triangle.

Three points of the tetrahedron are from the underlying triplet, and the fourth point is calculated by projecting a vector orthogonal to the plane of the triplet triangle. A triplet tetrahedron formed from a medial node triplet is shown in Fig. 4.3. Points \mathbf{q}_1 , \mathbf{q}_2 , and \mathbf{q}_3 are nodes from the triplet. The fourth point, \mathbf{q}_4 , is positioned along a unit vector $\hat{\mathbf{n}}$, orthogonal to the plane made by the triplet triangle. The origin of the orthogonal vector is located at the center of mass of the triplet triangle, which is defined as the average location of those three points, i.e.,

$$\mathbf{c}_m = \frac{\mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3}{3}. \quad (4.3)$$

The distance, h , to the fourth point from the triplet triangle's surface, has been chosen to be a function of the average length of the triplet triangle sides. Roughly equilateral triplet triangles are sought to avoid transformations involving eccentric triangles which would be prone to degenerate solutions of the affine transform. To extend the triplet to a tetrahedron, h is now derived with reference to Fig. 4.4.

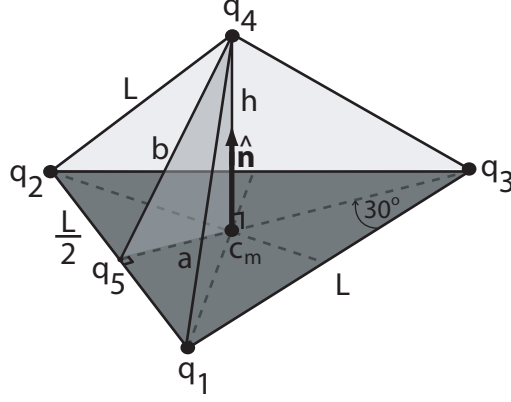


Figure 4.4: A triplet tetrahedron to illustrate the derivation of h .

For the derivation of h , the case where a triplet triangle is exactly equilateral is examined, with the length of each triplet triangle side denoted as L . In the plane of the equilateral triangle, the angle at each vertex is bisected and the intersection of these bisects occurs at the center of mass c_m . The bisect at vertex \mathbf{q}_3 intersects the $(\mathbf{q}_1, \mathbf{q}_2)$ side of the triangle at point \mathbf{q}_5 . This intersection is orthogonal to that side of the triangle, and thus $\|\mathbf{q}_5 - \mathbf{q}_2\| = \|\mathbf{q}_5 - \mathbf{q}_1\| = \frac{L}{2}$. The distance between \mathbf{q}_5 and \mathbf{c}_m is denoted as a , and is calculated by

$$a = \frac{L}{2} \tan(30^\circ) = \frac{L}{2} \frac{\sqrt{3}}{3}. \quad (4.4)$$

The points \mathbf{q}_5 , \mathbf{c}_m , and \mathbf{q}_4 form a right triangle for which h can be calculated. The hypotenuse of this right triangle is of length b , calculated by the Pythagorean Theorem as

$$b = \sqrt{L^2 - \frac{L^2}{4}} = \sqrt{\frac{3}{4}L^2} = \frac{\sqrt{3}}{2}L. \quad (4.5)$$

Once again, utilizing the Pythagorean Theorem,

$$h^2 = b^2 - a^2. \quad (4.6)$$

Solving for h , and substituting Eqs. 4.4 and 4.5 into the above equation yields:

$$\begin{aligned}
h &= \\
&\sqrt{b^2 - a^2} = \\
&\sqrt{\left(\frac{\sqrt{3}}{2}L\right)^2 - \left(\frac{\sqrt{3}}{3}\frac{L}{2}\right)^2} = \sqrt{\frac{3}{4}L^2 - \frac{3}{9}\frac{L^2}{4}} = \\
&\sqrt{\frac{27}{36}L^2 - \frac{3}{36}L^2} = \sqrt{\frac{24}{36}L^2} = \\
&\sqrt{\frac{2}{3}}L.
\end{aligned} \tag{4.7}$$

To extend the equilateral case to the general case of any triplet triangle, h is calculated with the average length weighted by $\sqrt{\frac{2}{3}}$, i.e.,

$$h = \sqrt{\frac{2}{3}}\bar{L}, \tag{4.8}$$

where \bar{L} is the average length of the triangle sides, $\bar{L} = \frac{\mathbf{L}_{1,2} + \mathbf{L}_{2,3} + \mathbf{L}_{1,3}}{3}$. The constant $\sqrt{\frac{2}{3}}$ ensures that a perfect equilateral triangle would yield a perfect equilateral tetrahedron and that, in general, h will produce a reasonably proportioned tetrahedron.

The orthogonal unit vector $\hat{\mathbf{n}}$ is calculated by taking the cross product of the vectors between any two vertices of the triplet triangle. The direction of $\hat{\mathbf{n}}$ is determined by the “right-hand rule.” For example, using the $(\mathbf{q}_1, \mathbf{q}_2)$ and $(\mathbf{q}_1, \mathbf{q}_3)$ sides of the tetrahedron,

$$\hat{\mathbf{n}} = [(\mathbf{q}_2 - \mathbf{q}_1) \times (\mathbf{q}_3 - \mathbf{q}_1)]. \tag{4.9}$$

Finally, the location of the fourth point with respect to the triplet’s surface that converts the triangle into a tetrahedron is calculated by

$$\mathbf{q}_4 = \mathbf{c}_m + h\hat{\mathbf{n}}. \tag{4.10}$$

To expand Eq. 4.1 to four points, let $\mathbf{r}_i = (x_{r,i}, y_{r,i}, z_{r,i})$ be the coordinates of the i^{th} node from a triplet in image R and $\mathbf{q}_j = (x_{q,j}, y_{q,j}, z_{q,j})$ be the coordinates of the corresponding j^{th} node from a triplet in image Q. The system can be represented as

$$[\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4] = \mathbf{T} [\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3, \mathbf{Q}_4], \tag{4.11}$$

where \mathbf{R}_i and \mathbf{Q}_i for $1 \leq i \leq 3$ are the homogeneous coordinates of the vertices of the triplet in images R and Q, respectively, \mathbf{R}_4 and \mathbf{Q}_4 are the points orthogonal to the respective triplet triangles that create *triplet tetrahedrons*, and \mathbf{T} is the 3D affine transform matrix. The matrix form of the equation is

$$\begin{bmatrix} x_{R,1} & x_{R,2} & x_{R,3} & x_{R,4} \\ y_{R,1} & y_{R,2} & y_{R,3} & y_{R,4} \\ z_{R,1} & z_{R,2} & z_{R,3} & z_{R,4} \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{Q,1} & x_{Q,2} & x_{Q,3} & x_{Q,4} \\ y_{Q,1} & y_{Q,2} & y_{Q,3} & y_{Q,4} \\ z_{Q,1} & z_{Q,2} & z_{Q,3} & z_{Q,4} \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad (4.12)$$

where parameters a through l are the 12 parameters of the affine transform. Given a pair of calculated tetrahedrons from images R and Q, the affine transform can be calculated by

$$\mathbf{T} = \mathbf{R}\mathbf{Q}^{-1}. \quad (4.13)$$

Since it is guaranteed that no three points in either tetrahedron are collinear, the system will not be degenerate.

4.2 APPLYING THE AFFINE TRANSFORM AND COMPUTING CORRESPONDENCE LIKELIHOOD

Triplet correspondences imply a transform that maps nodes from image Q to those in image R. Triple tetrahedra are computed for every triplet in both images and an affine transform is computed between every tetrahedron and each of its corresponding tetrahedra. Each transform is then applied to all of the tetrahedra in image Q and a quantitative measure for each transform fit is calculated.

A transformed tetrahedron in image Q is denoted as \mathbf{Q}' and is calculated by

$$\mathbf{Q}' = \mathbf{T}\mathbf{Q}, \quad (4.14)$$

where $\mathbf{Q}' = \mathbf{R}$ if \mathbf{T} was calculated between these particular tetrahedra, and the underlying triplets are very similar.

The transform fit is calculated as the Root Mean Square (RMS) error between the vertices of triplet \mathbf{q}' and \mathbf{r} . It is defined as

$$RMS = \sqrt{\frac{1}{3} \sum_{i=1}^3 (r_i - q'_i)^2}. \quad (4.15)$$

A small value for RMS indicates that the tetrahedra are closely aligned, whereas a large value for RMS is indicative of a poor alignment between the tetrahedra.

Each pair of tetrahedra is assigned a correspondence likelihood as determined by a weighting function based on the RMS . Potential likelihood functions are the normalized decaying exponential or normalized Gaussian. The likelihood function assigns a value between 0 and 1, where 1 signifies a transform with perfect alignment. Tetrahedra accumulate correspondence likelihoods for a final vote of correspondence, denoted as χ . It is expected that false triplet correspondences will be further reduced and the number and strength of correct correspondences will be increased. To summarize, Algorithm 1 will compute a correspondence likelihood for tetrahedra in images R and Q. The algorithm is also summarized as a flow diagram in Fig. 4.5.

Algorithm 1 Algorithm for Computing the Triplet Transform Metric

for all triplets in images R and Q **do**

 compute tetrahedra from triplets in images R and Q

end for

for all tetrahedra in image Q **do**

 compute the affine transform between a corresponding tetrahedron in image R

 apply the affine transform to every tetrahedron in image Q

 calculate the RMS error for each transformed tetrahedron in image Q and its corresponding tetrahedron in image R

 calculate a correspondence likelihood for each applied transformation

end for

compute the triplet transform metric by normalizing the correspondence likelihood measures

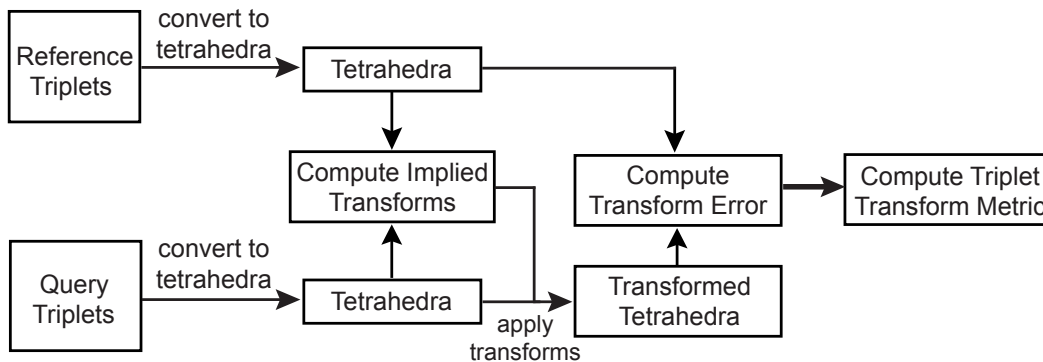


Figure 4.5: Computing the triplet transform metric.

This method is basically a form of RANdom SAMpling Consensus (RANSAC) [109], an algorithm typically used for fitting models in the presence of many outliers. The triplet transform metric tries to identify correspondences in the presence of many incorrect correspondences. It is expected that using implied transforms will increase the number of correctly identified correspondences, as compared to the unary, binary, and trinary metrics as illustrated in Fig. 4.6. In the figure, known correspondences are shown as \times 's on the matrix diagonal. The \times 's that are off the matrix diagonal illustrate incorrect correspondences. Following the application of implied transforms, these incorrect correspondences are removed, as illustrated in the left figure with dark circles. It may be more efficient to focus on triplets that have only one corresponding triplet since these are more likely to be a correct correspondence rather than a triplet with multiple correspondences.

4.3 EMPIRICAL RESULTS

The triplet transform metric is now demonstrated using medial nodes extracted from BrainWeb images. The BrainWeb image described in Section 3.6.2 was used for both the reference and query images. A total of 24 nodes were extracted using the core atom parameters listed in Table 4.1. Some of the nodes are shown with one of the images in Fig 4.7. Like previous

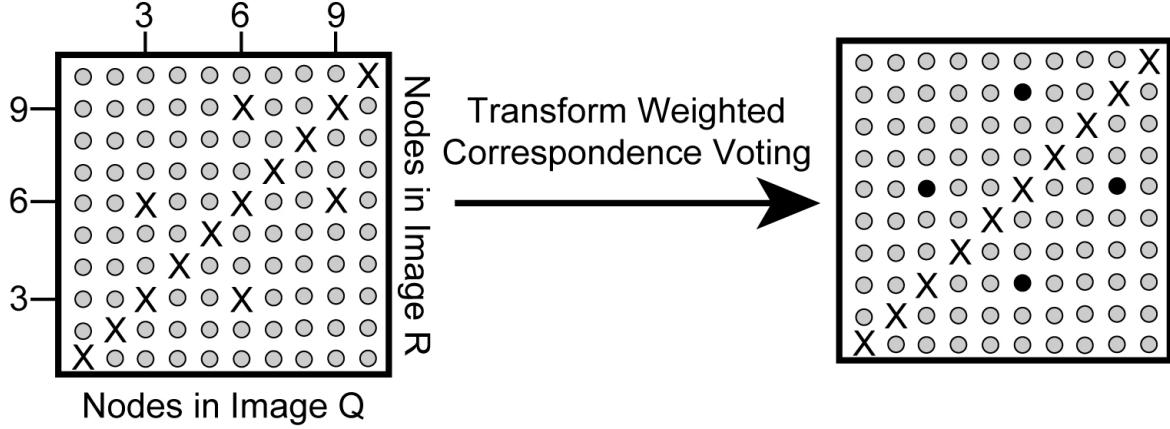


Figure 4.6: Left shows a correspondence matrix prior to using correspondence likelihood, where \times 's denote correspondences. Right shows an increased number of correct correspondences following the application of implied transforms. Dark circles are the previously incorrect correspondences.

experiments with identical images, known correspondences will be on the diagonal of a given correspondence matrix. Correspondences were calculated as before and a Gaussian with a mean of 0, standard deviation of 10, and amplitude of 1 was used for the correspondence likelihood function when calculating the triplet transform metric. All of these correspondence matrices are shown in Fig. 4.8. By comparing the triplet transform and trinary correspondence matrices, it is evident that correspondences are improved by the reduced number of off-diagonal correspondences.

The calculated ROC curves for the correspondence metrics are shown together in the same graph in Fig. 4.9. The resulting ROC curves show that the unary metric performs better than randomly guessing, and the binary and trinary metric perform sequentially better than the unary metric. Additionally, the triplet transform metric performs better than the trinary metric, which is quantified by the area under the ROC curve (Table 4.2). Unfortunately, the triplet transform metric suffers from computational complexity resulting in slow computation time. This experiment was performed on a laptop with a Pentium IV, 2.4 GHz mobile processor and 1 GHz of RAM. The computation times for evaluating the unary, binary, trinary, and triplet transform metrics were 1.1, 2.8, 5.1, 197.6 seconds, respectively.

As will be described in Chapter 6, an iterative optimization process will utilize the correspondence framework to estimate the location of anatomical landmarks in a query image. Since the process is iterative, including the triplet transform metric will make it extremely slow to converge. Thus, the triplet transform metric will not be included in the optimization process, which will be sufficient without it since the trinary metric has been shown to accurately determine correspondences. The triplet transform metric can be used in applications that are non-iterative or if time is not of concern. Such is the case for initializing anatomical landmarks in the query image, which will be discussed in greater detail in Section 6.4.

Table 4.1: Core atom parameter settings for extracting medial nodes to demonstrate the triplet transform metric.

| Parameter | Description | Value |
|----------------------------|---|-------------------------------|
| Blur Repetitions | Number of applications of boundary point kernel | 2 |
| Boundary Point Threshold | Intensity threshold for determining boundary points | 128 |
| Distance Minimum | Minimum search distance for finding opposing boundary point | 15 |
| Distance Maximum | Maximum search distance for finding opposing boundary point | 40 |
| Face-to-Faceness Tolerance | Allowable tolerance for face-to-faceness | 0.10 |
| Polarity | Direction to search for opposing boundary point | 0 (direction of the gradient) |

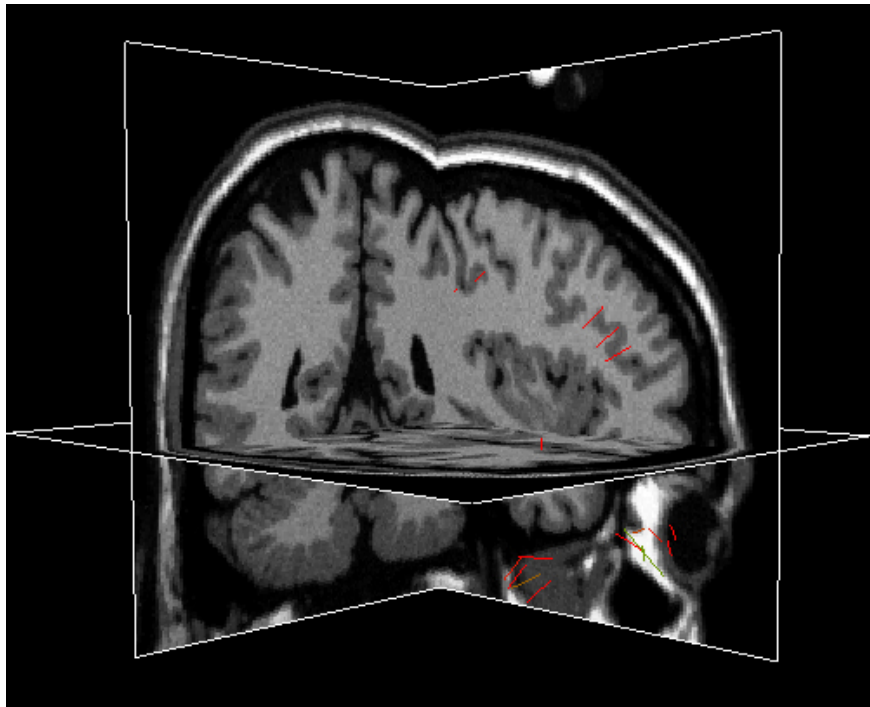


Figure 4.7: BrainWeb image used to demonstrate the triplet transform metric shown with extracted medial nodes.

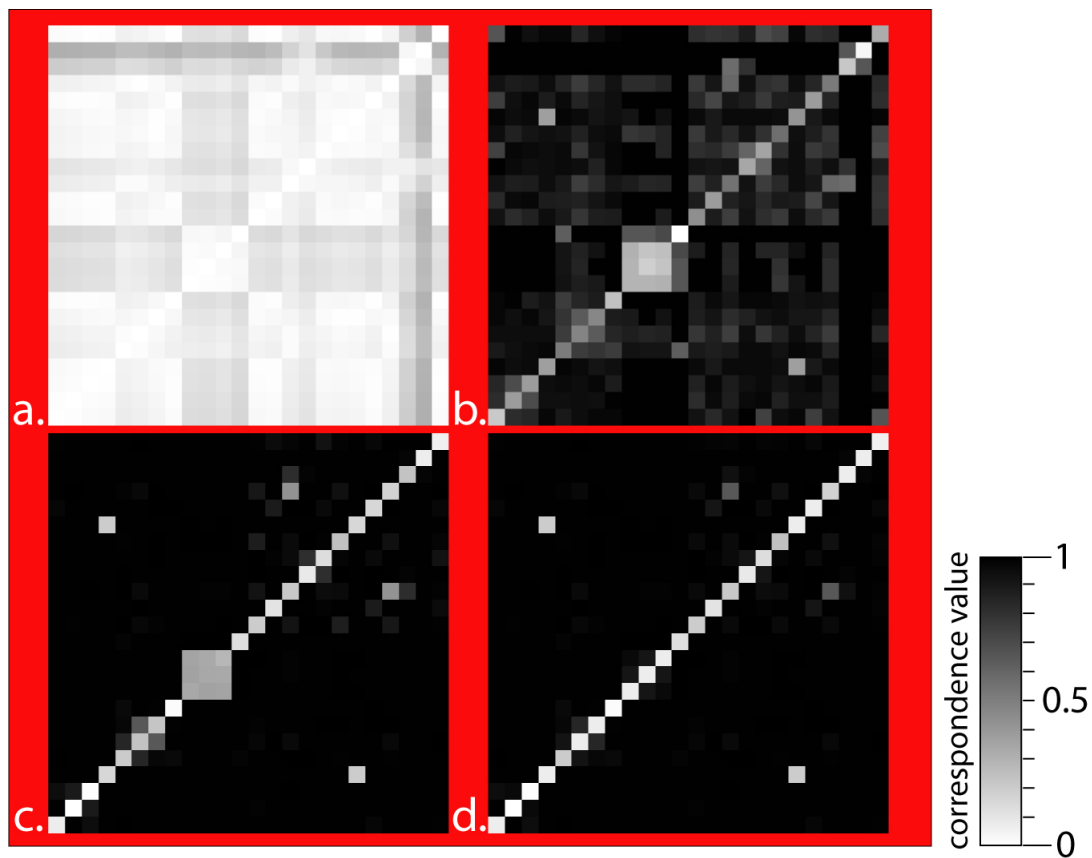


Figure 4.8: Correspondence matrices calculated by evaluating medial nodes extracted from identical BrainWeb images with the **a.)** unary, **b.)** binary, **c.)** trinary, and **d.)** triplet transform metrics.

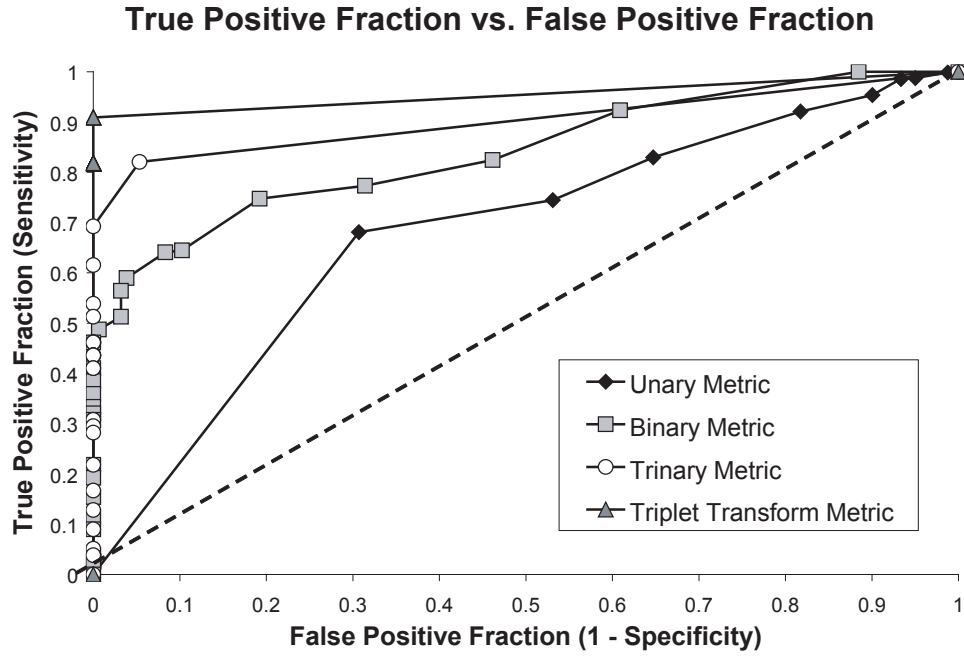


Figure 4.9: Calculated ROC curves for each correspondence metric, including the triplet transform metric. The dashed line indicates random guessing.

Table 4.2: Calculated A_{ROC} for each correspondence metric, including the triplet transform metric.

| | Area Under the ROC Curve |
|--------------------------|--------------------------|
| Unary Metric | 0.66 |
| Binary Metric | 0.84 |
| Trinary Metric | 0.90 |
| Triplet Transform Metric | 0.95 |

5.0 GENERATING VALIDATION DATA FROM A PARAMETERIZED MODEL

A primary goal of this research is the investigation of metric performance as a function of geometric variations between structures in two images. To achieve this goal, experiments will be conducted on carefully constructed computer generated data to 1) extract a reliable set of core atoms 2) permit a means of systematically varying structure geometry and 3) establish correspondence ground-truth. Core atoms are inherently sensitive to image noise and unable to extract medial properties at branches and ends. Core atoms are also dependent on *a priori* knowledge of desired local structure scale and intensity, requiring more than one application to locate multiple structures that differ significantly in these parameters.

The limitations of core atoms will be overcome by creating individual structures that are conducive to core atom formation with a single pass, and by forming core atoms on each structure independently. The core atoms from these individual applications will be combined into a single set of core atoms for further analysis. This population of core atoms will be sufficient for validating the correspondence methods. The computer generated data will be constructed from a parametric model emulating a CT scan of a human brain. The brain was chosen because it consists of a large number of structures of varying size and shape. Since the model is entirely parameterized it will be straightforward to vary structure geometry. The details of determining ground-truth for correspondences from the model are detailed in Chapter 6.

5.1 DETERMINING INTENSITY RANGE OF COMPUTER GENERATED DATA

In practice, the preferred imaging modalities for structural brain scans are CT and MRI since both modalities capture high resolutions images. Generally, MRI images have more anatomic detail than CT images. However, CT images are acquired at a faster rate and are less expensive to produce than MRI images. Another important distinction between MRI and CT images is the calculation of image intensity. The voxel intensity of MRI images depends on several system dependent variables such as repetition time and echo time, whereas, the intensities of CT images are calculated on an absolute scale based on tissue attenuation. For example cerebrospinal fluid always has the approximate intermediate value of 15 HU in CT images, but can have a higher or lower native value in MRI images depending on the system settings used.

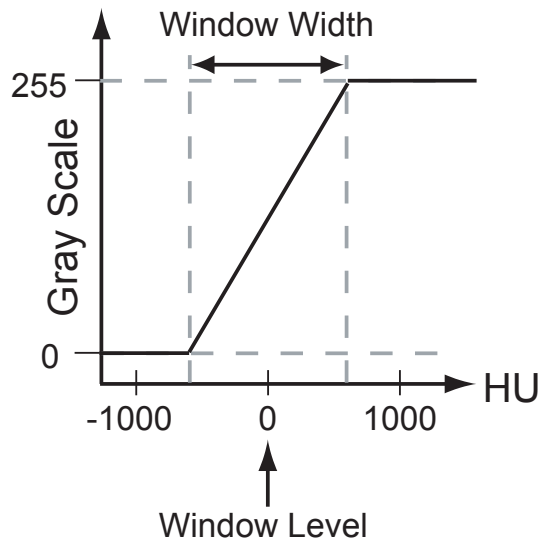


Figure 5.1: Assigning gray values to image pixels by windowing the range of Hounsfield Units. In this figure, the entire HU range is mapped to 8-bit gray values.

During CT image reconstruction, pixels are assigned a numerical value directly related to the X-ray attenuation characteristics of the particular scanned tissues. These values are measured in Hounsfield Units (HU), ranging from -1000 for air to +1000 for dense bone. Water is assigned a HU value of 0. Gray levels on an image display can be assigned to a

range of Hounsfield Units by *windowing* the HU range as shown in Fig. 5.1. The window width determines the range of Hounsfield Units to be displayed as gray values, and the window level is the center of this region. For imaging soft tissue, a window width of 400 or 600 HU and a window level of 0 or 100 HU are generally acceptable.

Rather than arbitrarily assigning intensity values to the brain structures in the parametric model, the HU for each structure is chosen such that it corresponds to what is typically found in clinical use. Gray values are calculated for each structure using a window width and level typical for visualizing soft tissue. The chosen Hounsfield Units were referenced from a paper published by Biswas and Gupta [110]. The authors developed methods for calculating the true tissue color of brain structures from CT images based on several parameters including HU values. In making their calculations, the HU values for brain structures were tabulated. Although, the CT images were obtained from goats, the measured HU should sufficiently reflect those of a human brain scan.

5.2 CREATING BRAIN STRUCTURES FROM SHAPE PRIMITIVES

The geometry of brain structures is built from shape primitives such as spheres, cylinders, and ellipsoids to ensure a rich set of core atoms. Compound shapes such as the cortex and cerebellum are generated using Constructive Solid Geometry (CSG). Shape primitives are combined using the four basic operations of CSG: union, intersection, difference, and merge. These CSG operations are briefly described in Table 5.1.

Table 5.1: The four basic CSG operations for creating complex shapes from primitives. A and B are the shape primitives and C is the resultant shape.

| CSG Operation | Brief Description | Notation |
|---------------|--|------------------|
| Union | Fuses two or more shapes into a single shape | $C = A \cup B$ |
| Intersection | Creates a shape from the overlapping regions of two or more shapes | $C = A \cap B$ |
| Difference | Subtraction of the second shape from the first shape | $C = A - B$ |
| Merge | Variant of union, where the internal surfaces are removed | $C = A \oplus B$ |

The parametric model for the generation of a synthetic brain image includes the following structures: cortices, eyeballs, optic nerves, brain stem, cerebellum, lateral ventricles, and thalami. The geometric representations and HU values for these structures are summarized in Table 5.2.

Table 5.2: Geometric representations and assigned Hounsfield Units of modeled brain structures.

| Structure | Shape | Description | HU |
|--------------------|---------------------|---|-----|
| Cortex | Multiple Ellipsoids | Merge of two ellipsoids to create right and left cortices then various difference operations with ellipsoids to refine the shape the cortex | 40 |
| Eyeballs | 2 Spheres | Perfect spheres defined by location and radius | 100 |
| Optic Nerves | 2 Cylinders | Cylinders cross each other anterior to the brain stem | 50 |
| Brain Stem | 1 Cylinder | Angled 45° from posterior to anterior in sagittal plane | 30 |
| Cerebellum | 2 Ellipsoids | Two merged and flattened (2 long axes and 1 short axis) ellipsoids | 45 |
| Lateral Ventricles | 2 Bent Cylinders | Manually delineated as a bent tube | 10 |
| Thalami | 2 Ellipsoids | Placed laterally to the brain stem | 34 |

5.3 EMPIRICAL RESULTS

In this experiment, a simulated CT brain image was generated using the previously described geometric representations and HU values for individual structures. The specific parameters for the eyeballs, optic nerves, brain stem, cerebellum, and thalami are listed in Table 5.3. The parameters for the lateral ventricles and cerebral cortex are not tabulated, but instead described below.

Table 5.3: Parameters used to model brain structures.

| Structure | Shape Parameters |
|--------------|--|
| Eyeballs | Spheres of radius 10 separated by 25 voxels along the x-axis |
| Optic Nerves | 2 cylinders of radius 2 and length 30 |
| Brain Stem | 1 cylinder of radius 5 and length 50 |
| Cerebellum | 2 ellipsoids with axes of length 30 and 30 along x- and z-axes, and 20 along the y-axis separated by 10 voxels |
| Thalami | 2 ellipsoids with axes of length 10 and 10 along the x- and y-axes, and 17 along z-axis separated by 14 voxels |

The lateral ventricles were modeled as 2 bent tubes with non-uniform curvature. The medial axis of one ventricle was digitally drawn on a grid. The coordinates of points along the medial axis were used to parameterize a tube. A tube with a radius of 3.9 voxels was centered at (42, 35, 65) and rotated 15° about the z-axis. The other ventricle was created by mirroring the same tube with respect to the yz-plane.

The left and right cortices were created by first merging 2 ellipsoids. Each ellipsoid had an axis length of 70 voxels along the x-axis, 60 voxels along the y-axis, and 80 voxels along the z-axis. The ellipsoids were centered 7 voxels from the image origin along the x-axis. To trim the bottom of these merged ellipsoids, an ellipsoid region was centered at (50, 60, 50) of size $70 \times 114 \times 96$ voxels. This region encased the merged ellipsoids, and any voxels not contained within this region were removed. Ellipsoid regions were also used to create grooves in front of and underneath the cortex by removing voxels within these defined regions.

The 8-bit gray level pixel values were calculated using a window range of 200 and window level of 100. Cross-sections of the resulting 8-bit image and a surface rendering are shown in Fig. 5.2 with structures labeled. Several other viewpoints of the surface rendering are shown in Fig. 5.3. The image size is $100 \times 100 \times 100$ voxels and structure locations are positioned relative to the center of the image. Geometric relationships between structures are defined, permitting a means to efficiently adjust structure geometry for a thorough analysis of the correspondence metrics in the presence of geometric variation. If more structures are required, they may be easily incorporated with the software that has been developed. The range of gray values can also be easily adjusted by changing the settings of the window range and level.

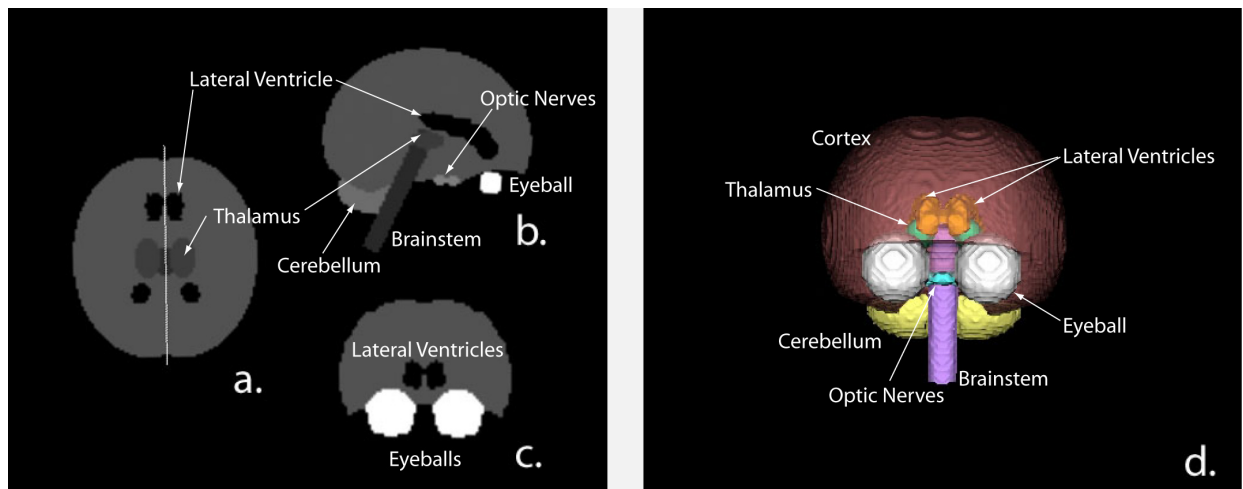


Figure 5.2: Computer generated CT brain image with structures labeled. Left shows a). transverse b). sagittal and c). coronal cross-sections of a computer generated CT brain image. Right shows a 3D rendering of brain structures modeled as shape primitives and compound shapes

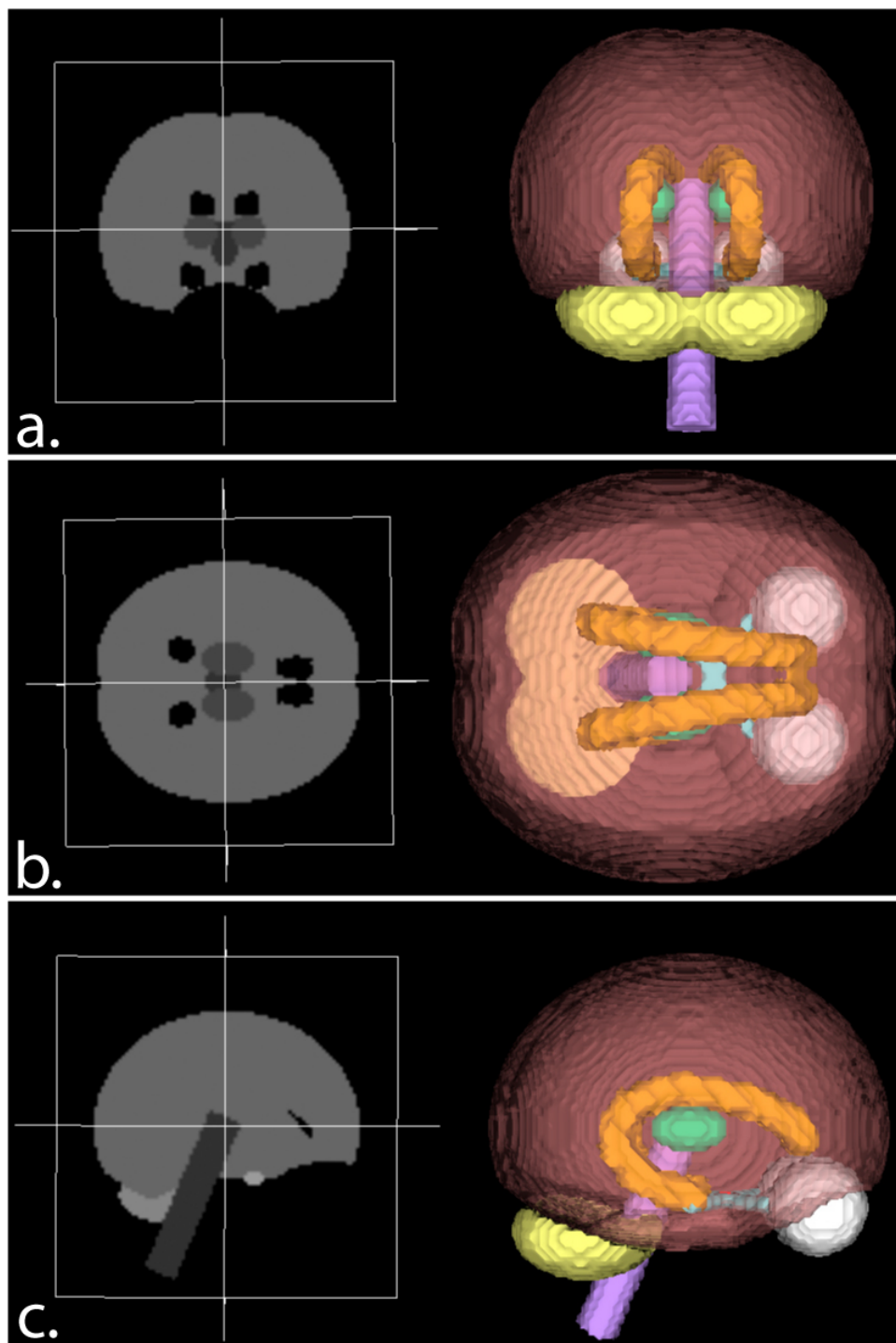


Figure 5.3: a). Coronal b). transverse, and c). sagittal views of a synthetic CT brain image. Left column images are cross-sections at the image origin. Right column of images are renderings of brain structures within a translucent cerebral cortex.

6.0 AUTOMATICALLY ESTIMATING ANATOMICAL LANDMARK LOCATIONS

The parametric brain model was developed to systematically perturb the geometry of structures to create query images that differ from the reference image. Perturbing the query image will introduce variation in the location and number of individual medial nodes due to sampling artifact, which rules out the previous method of validation based on known correspondences between individual nodes. Therefore, correct correspondences can no longer be expected to be along the diagonal of a correspondence matrix. Instead, the correspondence metrics will be validated by their ability to infer the location of anatomical landmarks in the query image given that the same landmarks are at known locations in the reference image. Landmarks differ from nodes in that landmarks are directly related to underlying anatomy and have unique identities, whereas nodes are processed from local image features.

Shown in Fig. 6.1 is a reference image depicting an eyeball with landmarks (solid \times 's) placed at the cornea, optic nerve, and center of the eyeball. Also shown is a query image with a larger, displaced eyeball. Ground-truth landmarks are shown as solid \times 's as determined by the parametric model. The dashed \times 's indicate landmark locations inferred from correspondences between cliques containing landmarks and nodes (dots). In this chapter, landmarks are defined, the method for inferring their locations from feature correspondences is developed, and their role in validating correspondences for images with structure variation is detailed.

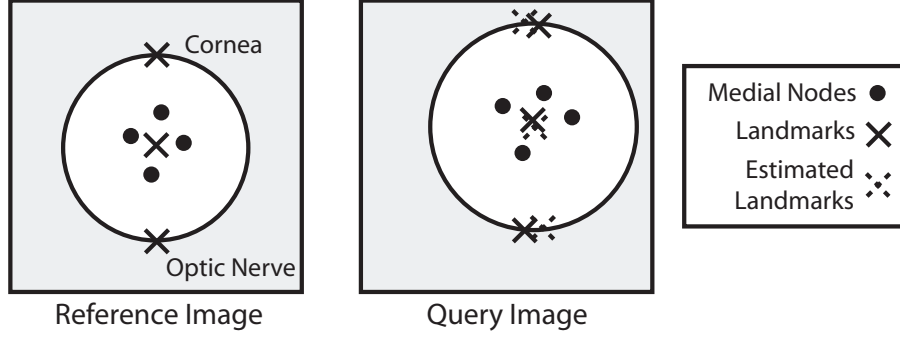


Figure 6.1: Reference image depicting an eyeball and query image depicting an eyeball with geometric variation (shifted and larger). Left shows landmarks placed on the reference image at the cornea, optic nerve, and center of the eyeball (solid \times 's). Right shows ground-truth landmarks as solid \times 's and hypothetical estimated landmarks as dashed \times 's.

6.1 DEFINING LANDMARKS

In the eventual clinical system, an expert will place landmarks of interest in the reference image. Since the data is parametrically defined by a model, anatomical landmarks can be automatically placed at significant locations on a shape's medial manifold, surface, or anywhere within the shape, and the location of these landmarks can deform with the model providing ground-truth. Thus, there are three different classes of landmarks: *medial landmarks*, e.g., center of a spherical structure, *surface landmarks*, e.g., anterior horn of the lateral ventricles, and *interior landmarks*. All classes of landmarks can be placed in an image by the parametric brain model. An example set of landmarks for brain structures are tabulated in Table 6.1.

It may be possible to reduce the number of clique comparisons by differentiating between the three types of landmarks. However, the current goals of this dissertation will focus on developing methods to include all types of landmarks together. Computational efficiency will be addressed in future research. With the introduction of landmarks, there exists three different clique configurations: those consisting of only 1) nodes, 2) only landmarks, or 3) both landmarks and nodes.

Table 6.1: Examples of potential landmarks for select brain structures.

| Structure | Potential Landmark Locations |
|-------------------|--|
| Eyeball | Center, optic nerve junction, and corneal junction |
| Optic Nerves | Intersection of nerves and points along the central axes |
| Brain Stem | Points along the central axis |
| Cerebellum | Center location |
| Lateral Ventricle | Ends and points along the central axis |
| Thalamus | Center and points along central axis |

In addition to location, each landmark has properties including a label, scale, and dimensionality. The i^{th} landmark from the landmark set from image α is defined as

$$L_{\alpha,i} = \{\ell_{\alpha,i}, m_{\alpha,i}, \sigma_{\alpha,i}, \lambda_{\alpha,i}\}, \quad (6.1)$$

where $\ell_{\alpha,i}$ is its location in image space, $m_{\alpha,i}$ is a label, $\sigma_{\alpha,i}$ is medial scale, and $\lambda_{\alpha,i}$ is medial dimensionality (a 2D vector). The set of N_L landmarks from image α is defined as

$$\Gamma_\alpha = \{L_{\alpha,1}, L_{\alpha,2}, \dots, L_{\alpha,N_L}\}. \quad (6.2)$$

The label is an *English descriptor* that uniquely describes the landmark for the purpose of identifying landmarks. Labels are chosen to describe the shape or feature represented by the landmark. Medial properties are assigned to landmarks, allowing correspondence metrics to be calculated between landmark-containing cliques by using the same correspondence metrics defined for medial nodes (Chapter 3). The medial scale and dimensionality of the landmarks are arbitrarily assigned since landmarks do not inherently have any medial properties.

There are several strategies to consider when assigning medial scale and dimensionality to a landmark. It is conceivable to assign landmarks varying dimensionality and scale based on nearby nodes or by anatomical membership (e.g., eyeball landmarks are small spheres and vessel landmarks are large cylinders). Conversely, all landmarks could be assigned the

same dimensionality and scale. In that case, spheres are a nice representation of landmarks because their orientation is arbitrary. For the remainder of this dissertation, all landmarks will be spherical with eigenvalues $\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}$ and eigenvectors in the direction of the cardinal axes (orientation is inconsequential due to spherical symmetry). The scale of all landmarks is assigned the same arbitrary value.

6.2 CALCULATING CORRESPONDENCES BETWEEN LANDMARKS

By assigning medial properties to landmarks, correspondences between landmarks can be evaluated with the same unary, binary, and trinary correspondence metrics already functioning for nodes. Since all landmarks have the same dimensionality and scale, there is no discrimination between landmarks when evaluated with the unary metric. However, geometric relationships between the landmarks are captured by the binary and trinary metrics. The correspondence values for the landmarks are appended to the rows and columns of the correspondence matrices originally devised for node correspondences as shown in Fig. 6.2. A correspondence matrix now contains all correspondences and cross-correspondences between nodes and landmarks. In the diagram, 1) node-to-node correspondences are shown in the white region, 2) landmark-to-landmark correspondences are shown in the light gray region, and 3) node-to-landmark correspondences are shown in the dark gray region. The portion of the matrix containing landmark-to-landmark correspondences is necessarily square since both images must contain the same number of landmarks. The reference landmarks are arbitrarily ordered and query landmarks are ordered in the same way since correspondences between them are known, i.e., corresponding landmarks have the same label. Similar to correspondence matrices computed for identical sets of nodes, corresponding landmarks will be on the matrix diagonal. The analysis of the matrix portion containing only landmark correspondences will be a critical step towards validating the correspondence metrics in the presence of geometric variation as described in the following section. It is important to note that correspondences evaluated between node-to-landmark cliques are not considered since it is not plausible for nodes to correspond to landmarks.

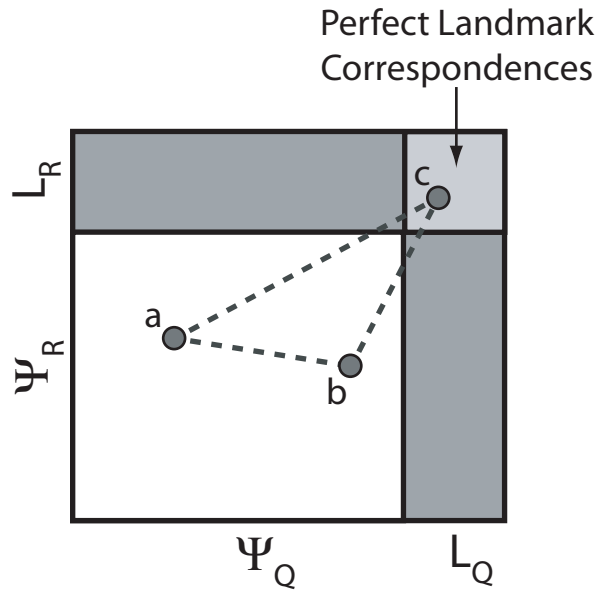


Figure 6.2: A correspondence matrix that includes the correspondences and cross-correspondences between nodes and landmarks. Node-to-node correspondences are shown in the white region. Node-to-landmark correspondences are shown in the dark gray region. Landmark-to-landmark correspondences are shown in the light gray region. A pair of corresponding triplets that consist of 2 nodes (labeled “a” and “b”) and a landmark (labeled “c”) is shown as dark gray circles.

6.3 CORRESPONDENCE OPTIMIZATION TO INFER LANDMARK LOCATION

The system for inferring landmark locations is called MCTILL, an acronym for Medial Correspondences To Infer Landmark Location. Correspondences between cliques consisting of nodes and landmarks are employed by the system to estimate the location of landmarks in the query image by iteratively repositioning the query landmarks in an effort to maximize the landmark correspondences as measured between landmarks known to correspond, and to minimize it between those known not to correspond (cross-correspondences). MCTILL is illustrated in Fig. 6.3. MCTILL operates on two images: a reference image and a query image. In this dissertation, the reference image has an associated collection of landmarks automatically placed by the parametric model called the *landmark model*. These landmarks are to be automatically placed in the query image. The ground-truth location for the landmarks in the query image is determined parametrically to validate the system (Chapter 7). Eventually, the system will operate on medical images, and the landmark model in the reference image will be manually defined by experts.

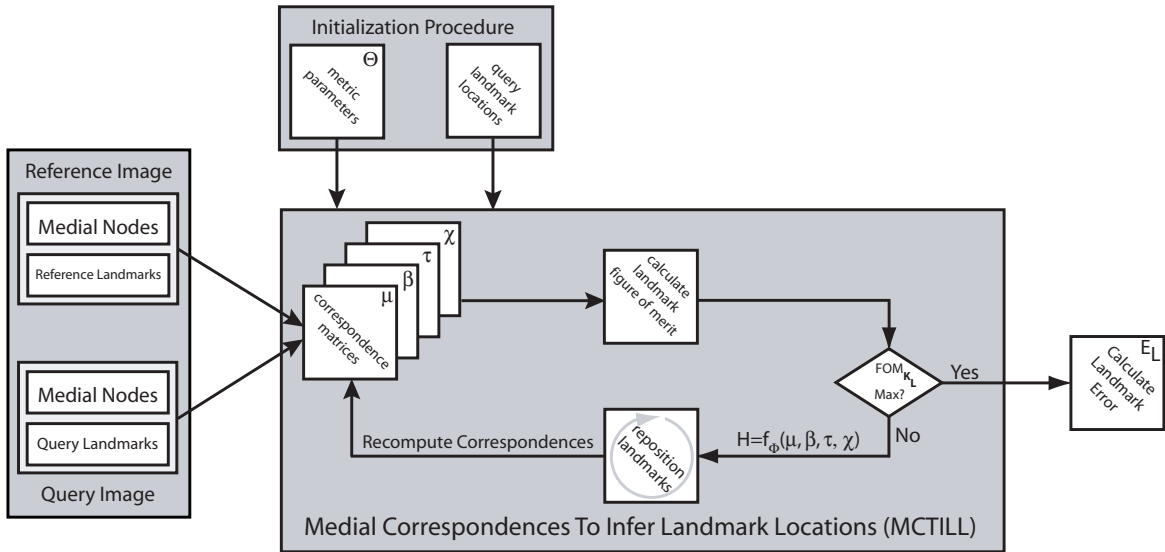


Figure 6.3: System for inferring landmark locations from clique correspondences.

The primary objective of MCTILL is to estimate landmark locations as shown in Fig. 6.4. The landmark model consists of the solid \times 's in the reference image. The geometric relationships between landmarks are shown as gray, dashed lines. A doublet consisting of a landmark and node is shown as a black dashed line. The solid lines between the images indicates a correspondence between this doublet and its corresponding doublet in the query image. Landmarks whose locations have been estimated based on such correspondences are shown as dashed \times 's in the perturbed query image. After landmark locations are estimated with MCTILL, the accuracy in estimating the location of query landmarks can be quantified and used as a measure of metric performance. This accuracy measure will be calculated as the RMS error between the estimated landmarks and the ground-truth landmarks (established by the parametric model as discussed in Chapter 5).

The method for placing landmarks in the query image is briefly outlined below in Table 6.2. *Steps 1 and 2* are initialization procedures for the inherent correspondence parameters and the location of query landmarks. *Steps 3-6* describe how MCTILL will optimize landmark correspondences by repositioning the query landmarks in an effort to maximize the landmark correspondences and minimize the landmark cross-correspondences. *Step 7* evaluates MCTILL's error in estimating landmark location, which is only applicable if a ground-truth is known for the query landmarks, as it is with the parametric model.

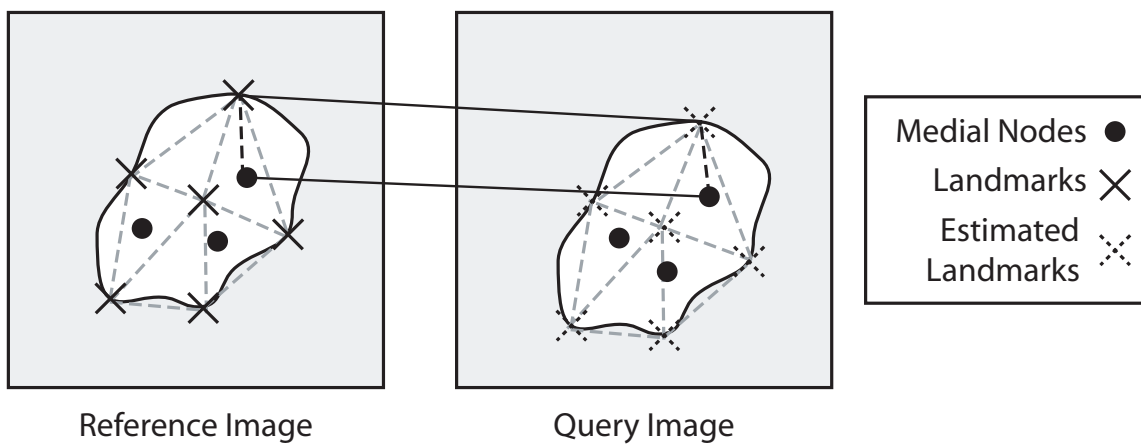


Figure 6.4: Left shows the landmark model as solid \times 's in the reference image. Geometric relationships between the landmarks are shown as gray dashed lines. A doublet consisting of a landmark and node is shown as a black dashed line. The solid lines between the images indicates corresponding doublets. Right shows estimated landmarks as dashed \times 's in the perturbed query image.

Table 6.2: Outline for utilizing the correspondence framework to infer landmark locations in a query image. Steps 3-6 describe MCTILL.

| Step | Description |
|------|--|
| 1 | Heuristically initialize the inherent correspondence parameters. |
| 2 | Initialize the location of the landmarks in the query image. |
| 3 | Compute correspondences between the three types of cliques that consist of either all nodes, all landmarks, or both nodes and landmarks (unary correspondences do not need to be computed more than once). |
| 4 | Compute the landmark correspondence figure of merit from a correspondence matrix. |
| 5 | If the figure of merit satisfies a threshold criteria, go to step 7. If the figure of merit does not satisfy the threshold criteria, go to step 6. |
| 6 | Reposition the query landmarks and repeat steps 3 - 5 until the figure of merit satisfies the threshold criteria. |
| 7 | Calculate MCTILL’s error in estimating the location of landmarks in the query image if ground-truth is known, otherwise, end. |

As part of the initialization procedure, correspondences parameters inherent to each metric will be equally weighted. Let the set of parameters inherent to the unary and binary metrics be defined as

$$\Theta = \{a_\mu, b_\mu, a_\beta, b_\beta\}. \quad (6.3)$$

The parameters for the unary metric, a_μ and b_μ , will be set to $\frac{1}{2}$ since $a_\mu + b_\mu = 1$. The parameters for the binary metric will also be set to $\frac{1}{2}$ since they are constrained the same way as the unary metric. The trinary metric does not have any inherent parameters. Thresholds for all metrics were set to 0.10. Techniques to initialize landmarks in the query image are described below in Section 6.4.

MCTILL is an iterative process that operates by repositioning landmarks with the goal of maximizing landmark correspondences and minimizing landmark cross-correspondences. Critical to this optimization procedure is the portion of a given correspondence matrix that consists of only landmark-to-landmark correspondences since correspondence between landmarks are known. This sub-matrix is referred to as the *Landmark Correspondence Matrix* (LCM) and is illustrated in Fig. 6.5. A correspondence matrix is shown in the upper left corner. The circled quadrant of the correspondence matrix is the LCM, consisting of only landmark-to-landmark correspondences. The ideal case is shown for an LCM, where the 0's on the diagonal indicate perfect landmark correspondences and the 1's off the diagonal indicate no incorrect landmark correspondences. In computing these correspondences only the binary and trinary metrics are meaningful since only the geometric relationships will change as the landmarks are moved.

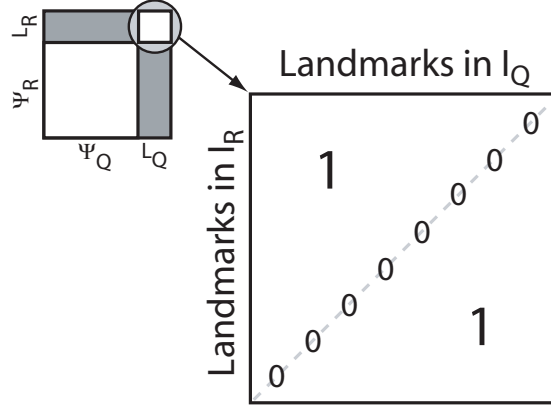


Figure 6.5: A correspondence matrix is shown in the upper left corner. The circled quadrant of the matrix consists of only landmark-to-landmark correspondences. The ideal case is shown, where the 0's on the diagonal indicate perfect landmark correspondences and the 1's off the diagonal indicate incorrect landmark correspondences

An LCM from a calculated correspondence matrix is denoted as \mathbf{K}_μ , \mathbf{K}_β , \mathbf{K}_τ , or \mathbf{K}_χ for the unary, binary, trinary, and transform metrics, respectively, and any LCM is denoted as \mathbf{K}_L . Recall that the landmarks in the LCM are automatically ordered such that proper correspondences fall on the diagonal of the matrix, as shown in Fig. 6.5. Thus, a figure of merit for maximizing landmark correspondences can be defined as

$$FOM_{\mathbf{K}_L} = \rho - \frac{\text{trace}(\mathbf{K}_L)}{N_L}, \quad (6.4)$$

where $\text{trace}(\mathbf{K}_L) = \sum_{i=1}^{N_L} \sum_{j=1}^{N_L} \mathbf{K}_{Li,j}$ for $i = j$, N_L is the number of landmarks, and ρ is a penalty factor to quantify cross-correspondences. The penalty factor is calculated as the average of the off-diagonal elements,

$$\rho = \frac{1}{N_O} \sum_{i=1}^{N_L} \sum_{j=1}^{N_L} \mathbf{K}_{Li,j} \text{ for } i \neq j, \quad (6.5)$$

where N_O are the number of off-diagonal elements in the LCM. The range of possible values for the landmark correspondence figure of merit is $-1 \leq FOM_{\mathbf{K}_L} \leq 1$. A value of -1 indicates completely incorrect correspondences and a value of 1 indicates perfect correspondences. The figure of merit can be calculated for all of the landmark correspondence

matrices, but we are aiming to maximize the figure of merit for \mathbf{K}_τ since the trinary metric is more accurate than the unary and trinary metrics, and faster to compute than the triplet transform metric.

The method for repositioning the query landmarks is a function of the correspondences and a set of parameters, Φ . This generalized function, called the *landmark displacement function* (LDF) is defined as

$$H = f_\Phi(\mu, \beta, \tau, \chi). \quad (6.6)$$

The LDF systematically chooses new landmark locations at each iteration through MCTILL to increase $FOM_{\mathbf{K}_L}$. The estimated landmark locations in the query image is constrained by the geometric relationship between landmarks in the reference image, as well as their relationships to corresponding nodes in the two images. In the terminology of deformable methods, i.e., snakes [24], the *internal energy* enforces geometric constraints on the landmarks and the *external energy* relates the landmarks to nodes, which are, themselves, related to both the reference and query images.

Two optimization techniques were developed and both of them are detailed in Section 6.5. The first technique is a *brute force optimization method* and the second technique is a *gradient ascent-like optimization method*. Each technique has advantages and disadvantages and both were employed for this dissertation research. The brute force optimization method is computationally expensive, operating by performing an exhaustive search, which provides insight into the optimization problem without having to categorize the local behavior of the figure of merit as a function of location. The gradient ascent-like optimization method is faster than the brute-force optimization method and used as the LDF for results presented in the following chapter.

Given the ground-truth for landmark locations provided by the parametric model, the performance of MCTILL can be assessed by computing a location error for each estimated landmark. This error measure is defined as the L2-Norm of the difference between the estimated landmark location and its ground-truth location, i.e.,

$$E_{L,i} = \|\ell_{Q,i} - \ell_{GT,i}\|, \quad (6.7)$$

where $\ell_{Q,i}$ is the estimated location of the i^{th} landmark in image Q and $\ell_{GT,i}$ is the i^{th} landmark's ground-truth location from the parametric model. An overall error measure for locating the set of landmarks in the query image Γ_Q is defined as the average of landmark errors, which is calculated as

$$\overline{E}_L = \frac{1}{N_L} \sum_{i=1}^{N_L} E_{L,i}. \quad (6.8)$$

It is possible to calculate E_L since the ground-truth locations for landmarks in Γ_Q are known from the parametric model. The overall performance of the system in estimating landmarks in the query image can be quantified by these error measures. Moreover, these error measures provide a means to evaluate the correspondence framework in the presence of geometric variation (Section 7).

In the following sections, strategies for initializing landmarks in the query image and optimization techniques for maximizing the landmark correspondence figure of merit $FOM_{\mathbf{K}_L}$ are described.

6.4 INITIALIZING QUERY LANDMARKS

Landmarks must be initialized in the query image prior to MCTILL’s optimization of landmark locations. If the orientation and size of the two images are known, the landmark locations in the query image can be approximated by their locations in the reference image. It becomes problematic with medical images since the object being imaged can move or deform. Thus, pre-processing would be required to align the image objects manually or with a registration method. Rather than relying on such pre-processing, it is proposed that the implied transforms developed in Chapter 4 be used to initialize landmark locations. Correspondences are computed with the triplet transform metric and the 4 node pairs with the highest triplet transform correspondence value in \mathbf{C}_χ are used to compute an affine transform. The affine transform is then applied to the set of reference landmarks, mapping the query landmarks into the coordinate system of the reference image. The transformed landmarks can then be placed within the query image, and thus initialize landmark locations. Results presented at the end of this chapter utilized this initialization method.

6.5 LANDMARK CORRESPONDENCE OPTIMIZATION TECHNIQUES

To maximize $FOM_{\mathbf{K}_L}$, two optimization techniques were developed. The *Brute Force Optimization* (BFO) method uses an LDF that displaces a given landmark to every location within a defined region. This technique is computationally intensive, but is an exhaustive search at a given resolution thereby avoiding being stuck in a local extremum. The *Gradient Ascent-Like Optimization* (GALO) method operates by calculating a $FOM_{\mathbf{K}_L}$ gradient and displacing the landmark in the direction of increasing $FOM_{\mathbf{K}_L}$. The gradient ascent-like method is computationally more efficient than the brute force method. The brute force method is still developed here because it provides crucial insight into the optimization problem.

The generalized algorithm to optimize landmark correspondences is shown as pseudocode in Algorithm 2. The set of landmarks is repeatedly iterated through until one of

two conditions are satisfied: Either landmarks can no longer be displaced to maximize the figure of merit, which indicates a global maximum, or a pre-determined number of iterations has been met. For each landmark in Γ_Q , correspondences are determined and $FOM_{\mathbf{K}_L}$ is calculated. Within a local sampling region, the current landmark is temporarily displaced via the LDF. All correspondences involving landmarks are then recomputed, and a new $FOM_{\mathbf{K}_L}$ is computed.

Once a landmark has been displaced and tested throughout the sampling region, the location yielding the largest $FOM_{\mathbf{K}_L}$ is recorded. After recording these maximum locations for each landmark in Γ_Q , the landmarks are permanently relocated. Landmarks are displaced in this manner to avoid order dependency when traversing the set of landmarks. Therefore, correspondences are determined based on the geometric relationships between the landmarks at their initial location.

After relocating the landmarks to their new locations, several criteria for ending the optimization are evaluated. First, it is determined if a global maximum for $FOM_{\mathbf{K}_L}$ was found by observing a decrease in $FOM_{\mathbf{K}_L}$ for the following several iterations. A second criteria is that if individual landmarks no longer move, presumedly because they have reached their local maximum. The third criteria is that if a pre-determined number of iterations have passed. This is a safety precaution for stopping oscillatory behavior and infinite loops. Shown in line 7 of Algorithm 2 is the LDF of either the BFO or GALO methods. The details of each method are described in the following sections.

Algorithm 2 General Algorithm for Optimizing Landmark Correspondences

```
1: while optimization criteria not satisfied do
2:   for all landmarks in  $\Gamma_Q$  do
3:     initialize the sampling region to the current landmark's location
4:     while  $FOM_{K_L}$  is not a local maximum and current landmark is within sampling
       region do
5:       calculate correspondences with the current landmark at its current location
6:       evaluate  $FOM_{K_L}$  for current landmark at its temporary landmark location
7:       temporarily displace the current landmark (LDF for either BFO or GALO meth-
         ods)
8:     end while
9:     record location of maximum  $FOM_{K_L}$  for the current landmark
10:  end for
11:  permanently relocate all landmarks to their new locations
12:  evaluate criteria to end optimization procedure
13:  increment iteration count
14: end while
```

6.5.1 Brute Force Optimization

The Brute Force Optimization (BFO) method makes use of an LDF that displaces a given landmark to every grid location within a sampling region. The sampling region 1) encapsulates voxels within an isometric search space of pre-defined size, 2) has the same number of dimensions as the image, and 3) is sampled with respect to the image's cardinal axes. The BFO method is described below, detailing its particular LDF in line 7 of the landmark optimization algorithm (Algorithm 2).

The sampling region is centered at the current landmark's initial location to allow for potential displacement to a landmark's previous location after the set of landmarks has been permanently repositioned. For a 3D image, let $\ell_{Q,i} = (x_i, y_i, z_i)$ be the location of the i^{th} landmark in the query image. The sampling region is centered at (x_i, y_i, z_i) and is of size $x_{ext} \times y_{ext} \times z_{ext}$, where $x_{ext} = y_{ext} = z_{ext}$ for an isometric search space. Thus, the origin of the sampling region is at $(x_i - \frac{x_{ext}}{2}, y_i - \frac{y_{ext}}{2}, z_i - \frac{z_{ext}}{2})$. The landmark is initially positioned at the origin and displaced along the cardinal axes at sampling intervals of $\Delta x = 1$, $\Delta y = 1$, and $\Delta z = 1$ for the x-, y-, and z-coordinates, respectively. Since every location is sampled within this region, a local maximum within the sampled space is guaranteed to be identified. At each sampled location, correspondences are calculated and $FOM_{\mathbf{K}_L}$ is evaluated. When the entire region has been traversed the location yielding the largest value for $FOM_{\mathbf{K}_L}$ is recorded. This new location is defined as

$$\ell'_{Q,i} = (x_i + n_x \Delta x, y_i + n_y \Delta y, z_i + n_z \Delta z), \quad (6.9)$$

where n_x , n_y , and n_z are between 0 and x_{ext} , y_{ext} , and z_{ext} , respectively, and correspond to the n^{th} sampled location along the respective axes. After local maxima have been determined for the set of landmarks, each landmark is permanently displaced to its maximum location. This iterative process is repeated for the set of landmarks until one of the three end criteria discussed in the previous section is satisfied.

Even though the BFO technique is computationally intensive, critical information regarding the figure of merit as a function of landmark displacement can be obtained since identifying local extrema is guaranteed. Specifically, the figure of merit as a function of

landmark displacement can be investigated, and it can be shown that the global maximum for $FOM_{\mathbf{K}_L}$ occurs at or near a landmark's ground-truth location. In this case, the LDF keeps all but one query landmark fixed at their ground-truth locations and the optimization procedure displaces the unfixed landmark throughout the entire image, and $FOM_{\mathbf{K}_L}$ is evaluated for each iteration. The figure of merit should be maximized when the unfixed landmark reaches its ground-truth location.

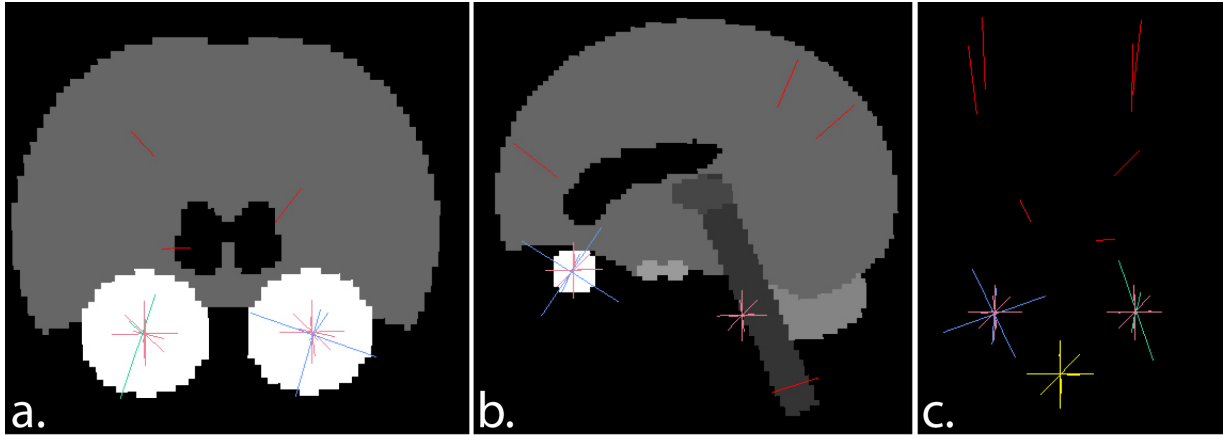


Figure 6.6: **a.** Coronal and **b.** sagittal cross-sections of a synthetic brain image shown with the sets of medial nodes and landmarks (pink) overlaid. **c.** The sets of medial nodes and landmarks used to demonstrate the brute force optimization technique. The landmark being displaced throughout the image is shown as yellow.

To demonstrate this phenomenon an experiment was conducted with a 3D image generated from the parametric brain model. This image was of size $100 \times 100 \times 100$ voxels and used as both the reference and query images (Fig. 6.6). Medial nodes were extracted from the reference image at the center of the eyeballs and throughout the cerebellum. Query nodes were obtained by manually and randomly perturbing the location and dimensionality of the medial nodes from the reference image to introduce variation between the reference and query landmarks. Landmarks were assigned perfect spherical dimensionality (recall $\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}$ for a perfect sphere) and a scale smaller than that of spherical nodes calculated from the image so as to not confuse a landmark for an spherical node.

A total of 10 medial nodes were extracted from the reference image: a spherical node in the right eyeball, a cylindrical node in the left eyeball, and 7 slab-like nodes throughout the cerebellum. Three landmarks were used for both images: 1 at the center of each eyeball

and 1 along the surface of the brainstem. The location of these landmarks, in data space coordinates, are tabulated in Table 6.3. The nodes and landmarks are shown without the image in Fig. 6.6c. The “brainstem” landmark was chosen to be the unfixed landmark and is highlighted yellow.

Table 6.3: The set of landmarks for demonstrating the brute force optimization technique.

| | Anatomical Location | Data Space Coordinates (x,y,z) |
|------------------|----------------------------|---------------------------------------|
| Fixed Landmark 1 | Center of Right Eyeball | (62.5, 35, 25) |
| Fixed Landmark 2 | Center of Left Eyeball | (37.5, 35, 25) |
| Brain Stem | Surface of Brainstem | (50, 25, 50) |

It was expected that as the “brainstem” landmark was moved closer to its ground-truth, the figure of merit would increase, peaking at the ground-truth location. While the landmark was far from its ground-truth, not much change was expected in the figure of merit since such large geometric distortions will no longer affect correspondences. Thus, for this reason, and for the sake of computation, rather than sampling all 100^3 voxels, 3 sampling schemes were utilized in a coarse-to-fine sampling resolution. The unfixed landmark was displaced for 3 sampling intervals within 3 different regions. First, a 25-voxel interval was used for the entire domain of the image. Next, a 5-voxel interval was used for a $25 \times 25 \times 25$ voxel region centered at the landmark’s ground-truth location. Finally, a 1-voxel interval was used for a $5 \times 5 \times 5$ voxel region centered at the landmark’s ground-truth. The coarse sampling regions verified that the figure of merit would not be maximum far away from the ground-truth. The finer sampling region allowed the figure of merit to be observed when the landmark was near its ground-truth. Ideally, displacing the landmark to maximize the figure of merit would lead directly to the ground-truth. Thus, if the landmark was initialized fairly close to the ground-truth, convergence to the ground-truth would be highly likely.

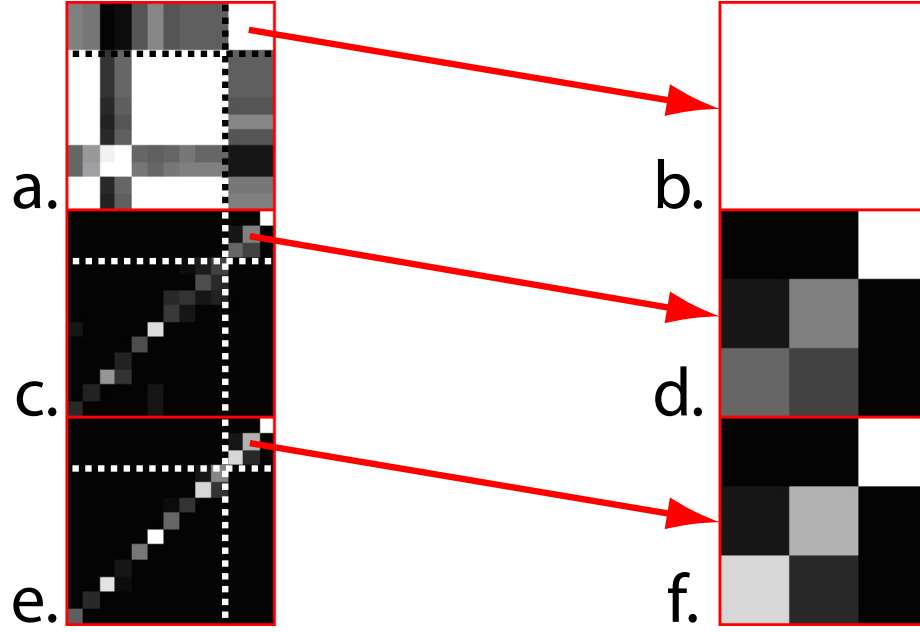


Figure 6.7: Correspondence matrices for each of the metrics used within the correspondence framework when the unfixed landmark is at ground-truth. Left: Dashed lines subdivide the correspondence matrices into node-to-node (lower left subdivision), node-to-landmark (upper left and lower right subdivisions), and landmark-to-landmark (upper right subdivision) correspondences. Right: Landmark correspondence matrices extracted from their respective correspondence matrices. **a).** Unary, **c).** binary, and **e).** trinary correspondence matrices. **b).** Unary, **d).** binary, and **f).** trinary landmark correspondence matrices.

Correspondences were determined and the figure of merit was computed at each displacement iteration. The calculated correspondence matrices with the unfixed landmark at its ground-truth are shown with the landmark correspondence matrices in Fig. 6.7. In the figure, dashed lines subdivide the correspondence matrices into node-to-node (lower left subdivision), node-to-landmark (upper left and lower right subdivisions), and landmark-to-landmark (upper right subdivision) correspondences. By observing the landmark correspondence matrices on the right side of the figure it can be seen that, as expected, the unary LCM is all white, and the binary and trinary LCM have a dominant diagonal. The figure of merit (Eq. 6.4), with and without penalty ρ , was calculated at the landmark's ground-truth for each metric (Table 6.4).

Table 6.4: Figure of merit for each metric when the unfixed landmark is at its ground-truth.

| | Unary Metric | Binary Metric | Trinary Metric |
|-----------------|--------------|---------------|----------------|
| With Penalty | 1.0 | 0.508 | 0.690 |
| Without Penalty | 1.0 | 0.652 | 0.774 |

Table 6.5: Figure of merit for each metric when the unfixed landmark is at the maximum.

| | Unary Metric | Binary Metric | Trinary Metric |
|-----------------|--------------|---------------|----------------|
| With Penalty | 1.0 | 0.633 | 0.754 |
| Without Penalty | 1.0 | 0.800 | 0.848 |

The maximum value for $FOM_{\mathbf{K}_\tau}$ was determined to occur at location (50, 25, 51), one voxel along the z-axis from the ground-truth. The figure of merit calculated for each correspondence matrix is shown in Table 6.5. In Figs. 6.8 and 6.9 are graphs of the figure of merit (with and without penalty) versus the distance of the landmark’s location to the ground-truth. As shown in either graph, the maximum occurs at a landmark error of 1 voxel. Also shown is an approximate 2.5 voxel accuracy for determining the correct location of the landmark in the presence of the manual perturbation in the location and dimensionality of the query nodes. Excluding the penalty factor in the figure of merit calculation merely shifts the data vertically. Also of note is the asymptotic behavior of $FOM_{\mathbf{K}_\tau}$ as the landmark error increases. When the unfixed landmark is moved too far from its ground-truth, correspondences consisting of the unfixed landmark remain unchanged resulting in a baseline correspondence measure consisting of the other landmark correspondences.

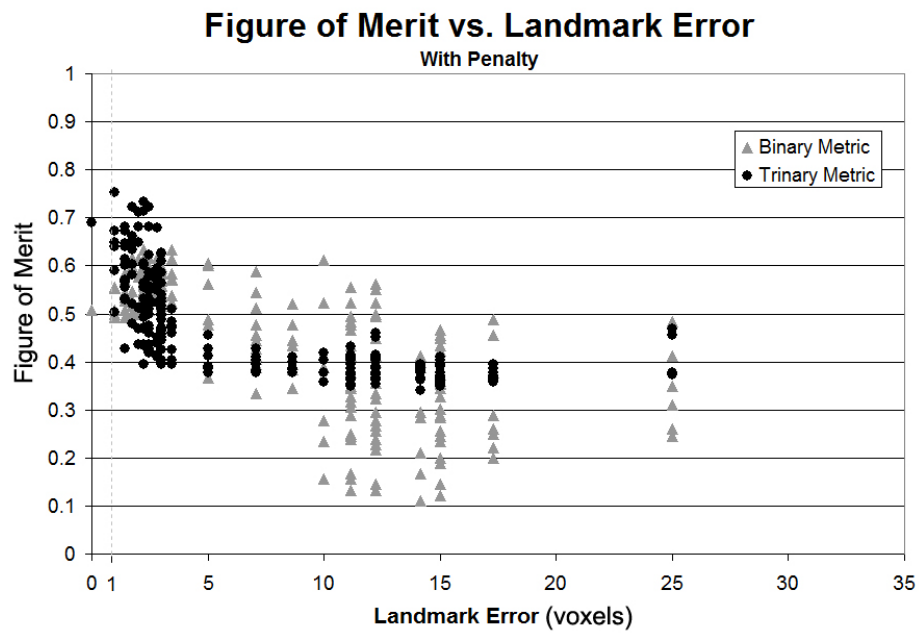


Figure 6.8: Figure of merit calculated with cross-correspondence penalty as a function of landmark error.

The correspondence metric value for the unfixed landmark as a function of landmark error is shown in Fig. 6.10. From the graph, it can be seen that there is good correspondence at the landmark's ground-truth among other locations. However, due to the cumulative nature of the correspondence matrix and variation introduced into the query landmarks, the figure of merit was not maximized exactly at the ground-truth.

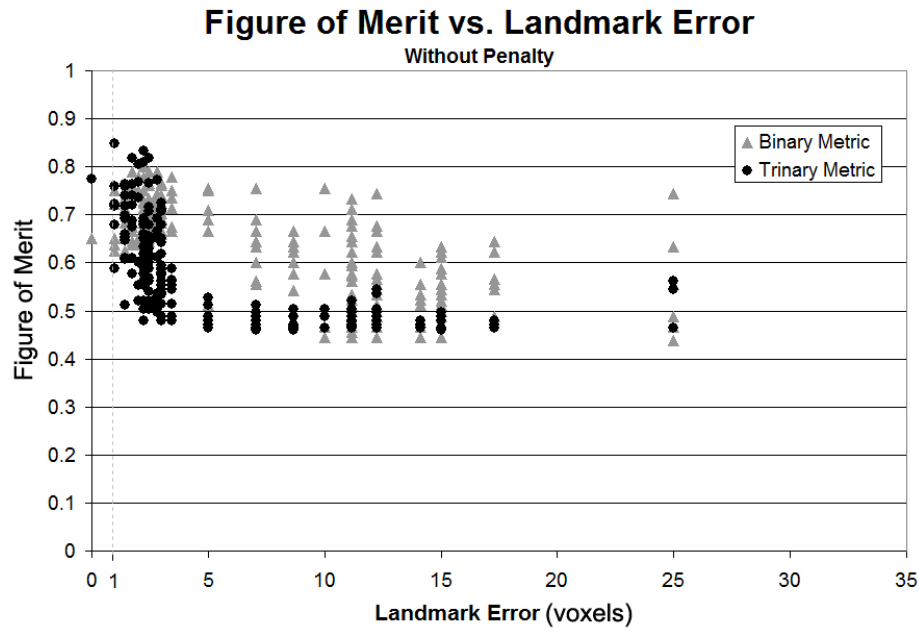


Figure 6.9: Figure of merit calculated without cross-correspondence penalty as a function of landmark error.

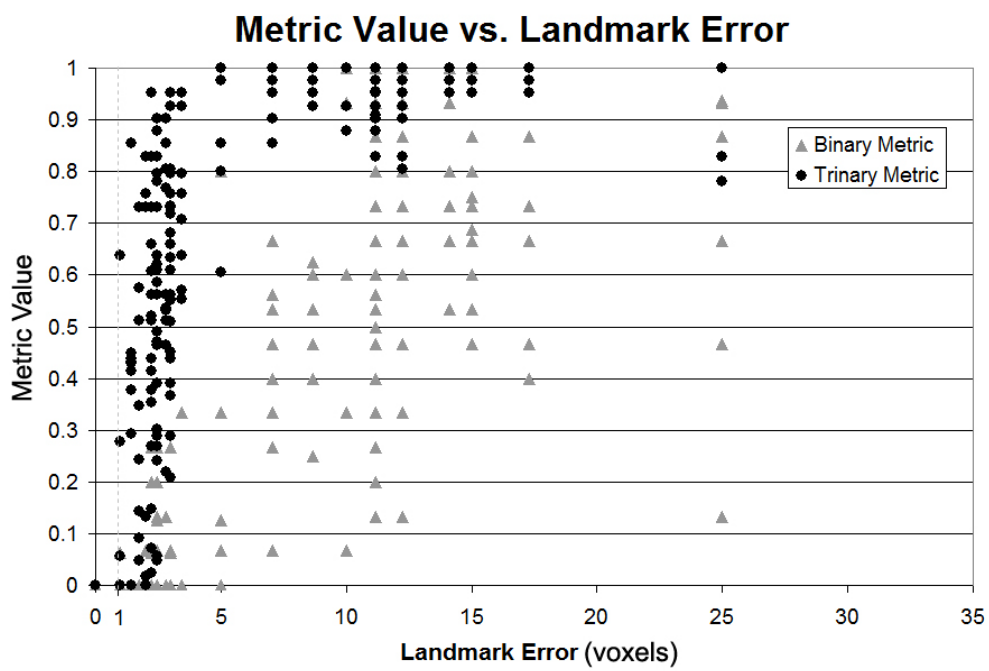


Figure 6.10: The unfixed landmark's correspondence metric value as a function of landmark error.

The figure of merit evaluated at each location in the 5^3 sampling region is visualized as a series of 2D images, as shown in Figs. 6.11 and 6.12. Each 2D image shows the value $FOM_{\mathbf{K}_\tau}$ scaled to color at each location in the 5×5 x-y plane while varying the z-coordinate. Additionally, a contour was created from these 2D images to better assess the smoothness of the figure of merit as a function of location. It is clear from observing the contours, that there exists a smooth path for increasing values of $FOM_{\mathbf{K}_\tau}$ to the maximum value. This was a critical new insight into the optimization problem, which encouraged the development of a more clever and efficient means of displacing the landmark. The new optimization technique is called *gradient ascent-like optimization* (GALO) and is detailed in the following section.

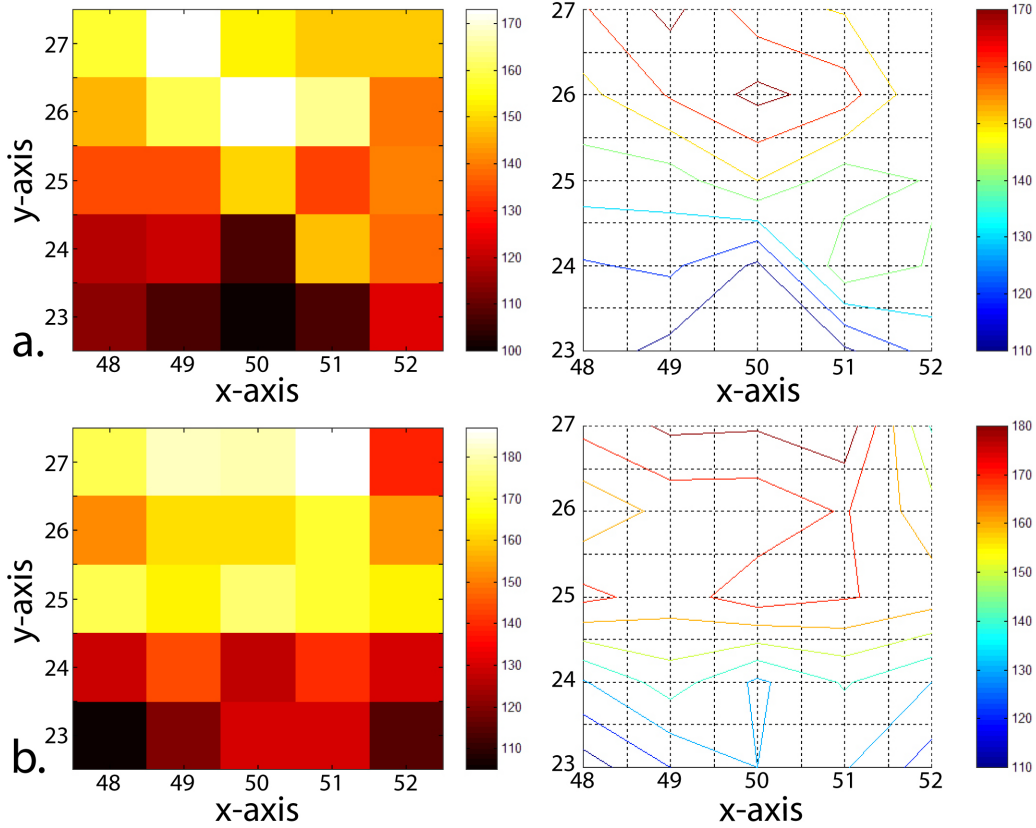


Figure 6.11: The figure of merit is displayed for the landmark in a 5×5 region in the x-y plane for **a.)** $z = 49$ and **b.)** $z = 50$. The ground-truth location of the landmark at (50, 25, 50). Note that the color scale is different between images.

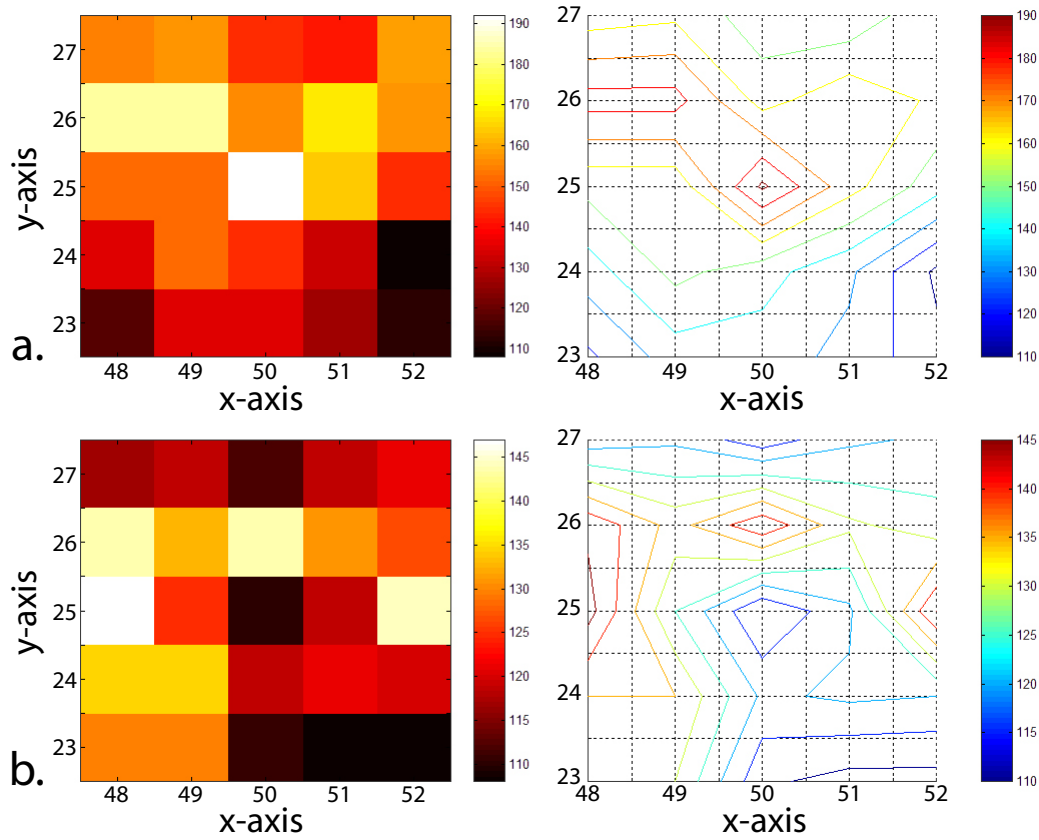


Figure 6.12: The figure of merit is displayed for the landmark in a 5×5 region in the x-y plane for **a.)** $z = 51$ and **b.)** $z = 52$. The maximum figure of merit is at (50, 25, 51). Note that the color scale is different between images.

6.5.2 Gradient Ascent-Like Optimization

The Gradient Ascent-Like Optimization (GALO) method is a greedy algorithm that utilizes the concept of gradient ascent, which is commonly used to maximize continuous, differentiable equations by moving in the direction of the gradient towards the global maximum. In general, greedy algorithms take the current local optimum with the hope of eventually finding the global optimum [111, 112, 113]. Greedy algorithms tend to be computationally efficient and give reasonable approximations of the global optimum. However because they do not usually consider all of the data, finding the global optimum is not guaranteed. Additionally, if a greedy algorithm makes erroneous commitments too early, they will be prevented from finding the global optimum. To combat these pitfalls, special care was taken to prevent them from occurring within the GALO method. The GALO method uses a sampling region in the same manner as the BFO method. If a plateau, i.e., where the maximum occurs for more than one consecutive location, is detected, the location furthest from the initial landmark location is utilized to advance the optimization. Also, if a local maximum is detected, the optimization process is advanced for another iteration to ensure that it was indeed at a maximum, i.e., the new maximum figure of merit will be less than the previous maximum figure of merit.

As shown in the previous section, the figure of merit as a function of landmark location is locally smooth. Utilizing this observation, it is reasonable to assume that there exists a gradient vector, which gives an approximate direction to the maximum figure of merit within the search space. This gradient vector is determined by first evaluating the figure of merit for each location along the cardinal axes (within the sampling region). The location that yields the maximum figure of merit along each cardinal axis is recorded under the hypothesis that it is a projection of the maximum location. These projections are then composed into a gradient vector, originating from the initial landmark location in the direction of the maximum. To determine the location that yields the maximum figure of merit along this gradient vector, a variant of the *bisection method* was devised. The bisection method is a common method for solving for the roots of a function by dividing the search interval in half at each iteration. A sign change of the function evaluated at the interval's midpoint determines the new interval.

Rather than a sign change, the GALO method looks for an increase in figure of merit at the midpoint to determine the new interval. The bisection method was implemented in a similar way as described in Numerical Recipes in C [\[114\]](#). Estimating landmark location with the GALO method is explored in the following chapter.

7.0 ASSESSING METRIC PERFORMANCE IN THE PRESENCE OF GEOMETRIC VARIATION:

In this chapter, the performance of the correspondence method is assessed in the presence of geometric variation between structures. Geometric variation is introduced by perturbing the location and scale of structures between images via the parametric model defined in Chapter 5. Correspondences are validated by their accuracy in inferring the location of anatomical landmarks. Since the ground-truth for landmark location is established by the parametric model, correspondence performance is quantified by the error in estimating landmark location as defined in Section 6.3.

Several experimental trials were conducted to investigate the capabilities of the correspondence and landmark estimation methods. For each trial, images were created from the parametric model (Chapter 5) with structures in the query image being geometrically perturbed from those structures in the reference image. Landmarks were initialized in the query image with the triplet transform method described in Section 6.4. For these trials, a 3D image of size $100 \times 100 \times 100$ voxels was generated from the parametric brain model using the parameters listed in Table 5.3 (Section 5.3).

Medial nodes were extracted at the center of the eyeballs and throughout the cerebellum. A total of 10 medial nodes were extracted from the reference image: a spherical node in the right eyeball, a cylindrical node in the left eyeball, and 7 slab-like nodes throughout the cerebellum. These nodes are shown with the image and landmarks in Fig. 6.6 (Section 6.5.1). Landmarks were assigned perfect spherical dimensionality and a scale of 3. The three landmarks tabulated in Table 6.3 (Section 6.5.1) were placed in the reference image.

7.1 DISPLACEMENT ALONG THE CARDINAL AXES

For this trial, the right eyeball was displaced in the query image. Displacement of the right eyeball in both positive and negative directions along the cardinal axes occurred at 5-voxel intervals with a maximum distance of 10-voxels along one of the cardinal axes. Shown in Figs. 7.1-7.3 are graphs of individual landmark error versus displacement of the right eyeball along each cardinal axis. A displacement of 0 indicates that the right eyeball is at the same location as in the reference image. A negative displacement indicates that the right eyeball was moved in the negative direction toward the image center, and a positive displacement indicates that the right eyeball was moved in the positive direction towards the image edge. The “right eyeball” landmark was most effected by each displacement since it is near the node associated with the right eyeball. As expected the landmark error was lowest near 0 displacement, and increased as the right eyeball was moved further away. The largest landmark error was approximately 2.6 voxels. The landmark error for the other two landmarks varied between 0 and 1.5 voxels. The average of errors for the set of landmarks \overline{E}_L was calculated for each displacement along each axis. These error measures are tabulated in Table 7.1 with their associated standard errors of the mean.

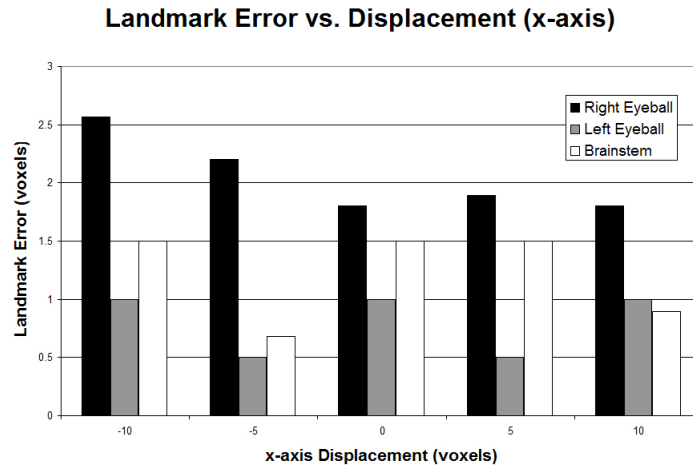


Figure 7.1: Landmark error vs. displacement of the right eyeball along the x-axis. A displacement of 0 indicates that the right eyeball is at the same location as in the reference image.

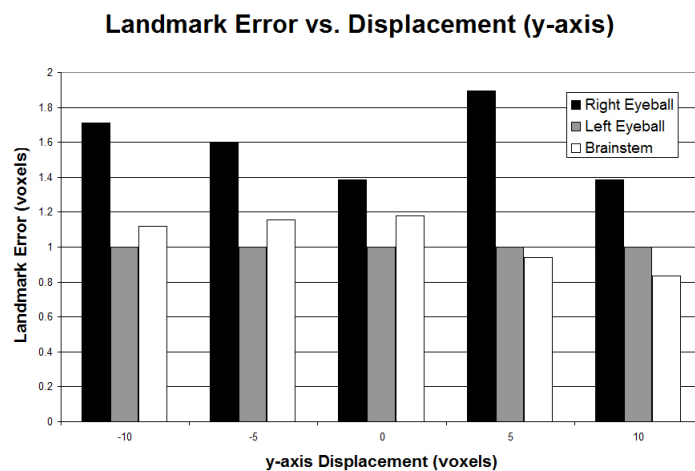


Figure 7.2: Landmark error vs. displacement of the right eyeball along the y-axis. A displacement of 0 indicates that the right eyeball is at the same location as in the reference image.

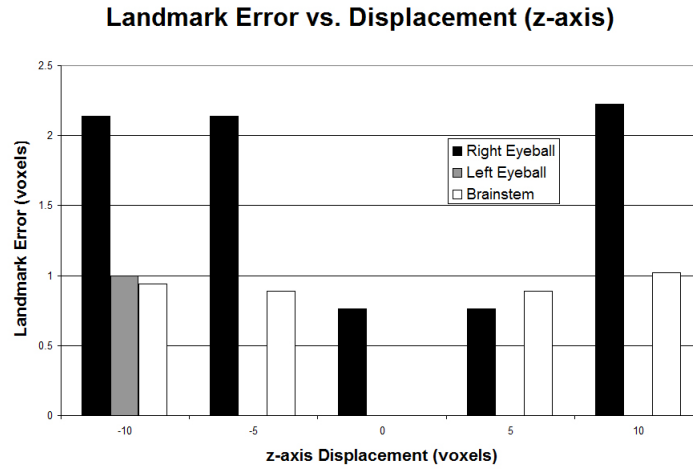


Figure 7.3: Landmark error vs. displacement of the right eyeball along the z-axis. A displacement of 0 indicates that the right eyeball is at the same location as in the reference image.

Table 7.1: Average landmark error and standard error for each displacement of the right eyeball along the cardinal axes.

| Displacement (voxels) | x-axis | y-axis | z-axis |
|-----------------------|---------------------------|---------------------------|---------------------------|
| | \overline{E}_L (voxels) | \overline{E}_L (voxels) | \overline{E}_L (voxels) |
| -10 | 1.69 ± 0.46 | 1.28 ± 0.22 | 1.36 ± 0.39 |
| -5 | 1.12 ± 0.53 | 1.25 ± 0.18 | 1.01 ± 0.62 |
| 0 | 1.43 ± 0.23 | 1.19 ± 0.11 | 0.25 ± 0.25 |
| 5 | 1.30 ± 0.41 | 1.28 ± 0.31 | 0.55 ± 0.28 |
| 10 | 1.23 ± 0.29 | 1.07 ± 0.16 | 1.08 ± 0.64 |

7.2 MEDIAL AND LATERAL DISPLACEMENT

In the second experimental trial, the effects of displacing multiple structures on estimating landmark location were investigated. In particular, nodes associated with each eyeball were displaced medially and laterally. For medial/lateral displacement, the nodes were moved from their initial locations toward the image center/edge at 2.5 voxel intervals for a maximum displacement of 10 voxels. Landmark error was computed for each landmark and graphed against medial (Fig. 7.4) and lateral (Fig. 7.5) displacement. From each figure it is observed that landmark error for the “right eyeball” and “left eyeball” landmarks increases as the eyeballs are displaced away from their initial locations. Similar to the previous experiment, the “brainstem” landmark is not significantly effected by eyeball displacement. The average of landmark error and standard error for each landmark is tabulated in Table 7.2. For both modes of displacement the average landmark error was approximately 2.5 voxels for each “eyeball” and approximately 1 voxel for the “brainstem” landmark. Note that the standard error calculations support the observation that the “eyeball” landmarks were more effected by displacement than the “brainstem” landmark.

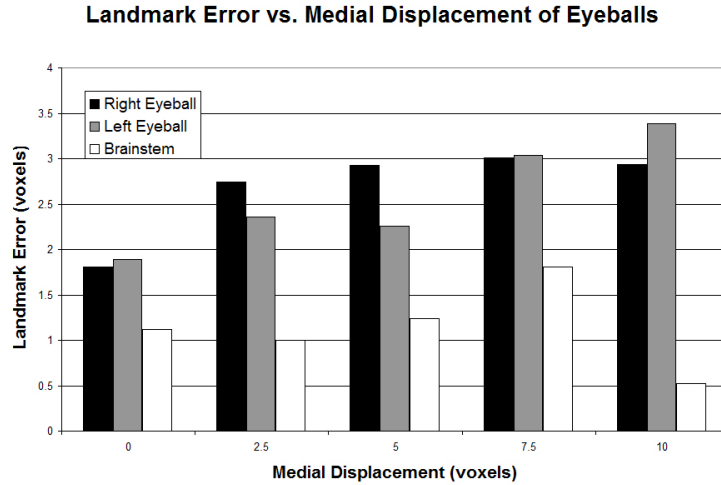


Figure 7.4: Landmark error vs. medial displacement of the eyeballs.

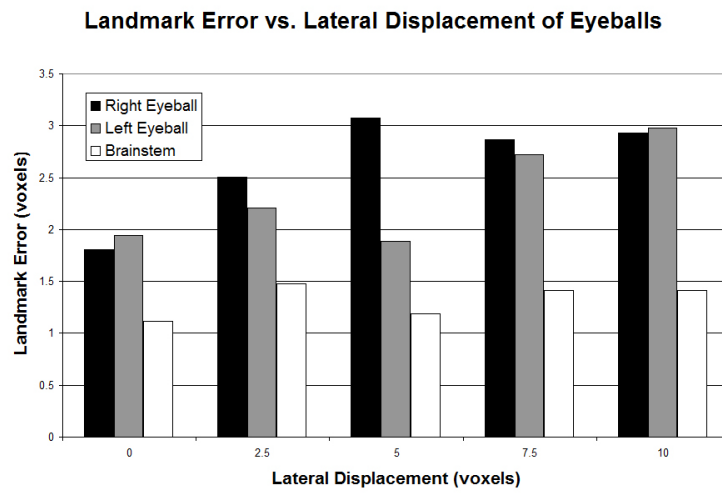


Figure 7.5: Landmark error vs. lateral displacement of the eyeballs

Table 7.2: The average of each landmark’s location error as a function of medial or lateral displacement.

| Displacement | R. Eyeball Landmark E_L (voxels) | L. Eyeball Landmark E_L (voxels) | Brainstem Landmark E_L (voxels) |
|---------------------|---------------------------------------|---------------------------------------|--------------------------------------|
| Medial | 2.68 ± 0.22 | 2.58 ± 0.27 | 0.98 ± 0.12 |
| Lateral | 2.63 ± 0.23 | 2.35 ± 0.22 | 1.32 ± 0.07 |

7.3 SCALE VARIATION

In this third experiment, landmark location estimation was investigated in the presence of variable eyeball diameter. Rather than extracting medial nodes from images generated from the parametric model, node scale was directly varied to eliminate any sampling artifacts that may occur during node extraction. The nodes extracted from both eyeballs were increased and decreased in scale at 5 voxel increments for a maximum scale change of 10 voxels from the original scale. Recall that the unary metric evaluates singlet correspondences by similarity in scale and dimensionality. In this experiment, the parameters for scale and dimensionality were equal in weighting (the flexibility of these parameters for tolerating scale variation is explored in the following experiment). Shown in Fig. 7.6 is landmark error as a function of change in scale. The approximate error for each landmark was 1 voxel with no apparent dependency on scale. This observation was quantitatively confirmed by computing the average error and standard error for each landmark over all scales. The average error for the “right eyeball,” “left eyeball,” and “brainstem” landmark was 1.12 ± 0.032 voxels, 0.98 ± 0.055 voxels, and 1.01 ± 0.056 voxels, respectively.

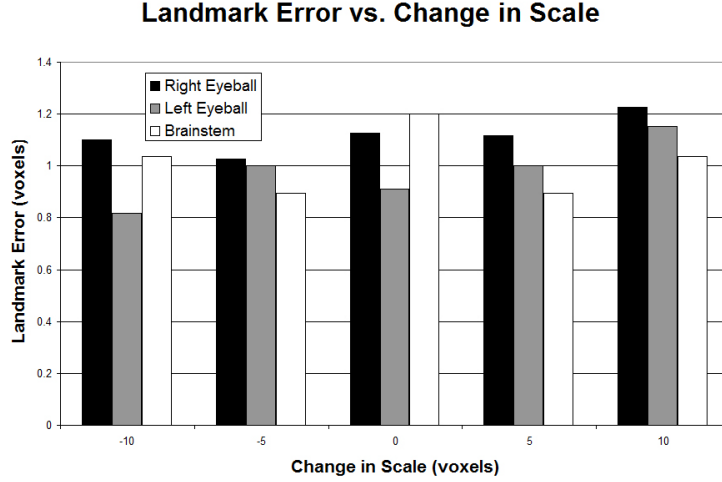


Figure 7.6: Landmark error as a function of change in node scale for the nodes extracted from the eyeball.

7.4 SCALE VARIATION AND THE UNARY METRIC

The effects of the unary metric parameters on correspondences and landmark location estimation in the presence of scale variation were examined in this fourth experiment. The right and left eyeball nodes were scaled to 28.00 and 28.24 voxels, respectively. Correspondences were determined while varying the scale term parameter for the unary metric b_μ from 0.3 to 0.6 in 0.1 increments. For each value of b_μ , landmark locations were estimated and landmark errors were calculated for each landmark (Table 7.3). As b_μ decreased, there is a general decrease in error for each landmark. A persistent decrease in landmark error was not observed due the nature of the optimization procedure to potentially improve upon a landmark's location at the expense of another landmark. A better measure is the average of landmark errors for each b_μ . These measures are also reported in Table 7.3, indicating an overall decrease in landmark error, i.e., increased accuracy in estimating landmark location.

After estimating landmark location for each b_μ , correspondences were recomputed with the query landmarks placed at their estimated locations. The resulting trinary correspon-

dence matrices are shown in Fig. 7.7. When $b_\mu = 0.5$, the nodes extracted from the right (orange square) and left (blue square) eyeballs have very low correspondence values. However, when b_μ was decreased to 0.4 and 0.3, the eyeball nodes have significantly higher correspondence values. Also note that when $b_\mu = 0.3$ and $b_\mu = 0.4$, there is very little observable differences between the matrices. In particular, the landmark correspondence matrices are very similar, which is a likely explanation for the right and left eyeball landmark estimates being identical for each of these values of b_μ .

In summary, by adjusting the weight on the scale term of the unary metric, correspondences can be determined in the presence of scale variation. This first prototype of MCTILL did not justify a complete tuning of metric parameters or large-scale validation. Rather, it was a test, not just for the system itself but for methods of validation and parameter tuning that would be applied to a future system that utilizes improved features.

Table 7.3: Each landmark location error for multiple values of b_μ .

| b_μ | Right Eyeball Landmark E_L (voxels) | Left Eyeball Landmark E_L (voxels) | Brainstem E_L (voxels) | \overline{E}_L (voxels) |
|---------|--|---|-----------------------------|---------------------------|
| 0.6 | 1.10 | 0.82 | 1.94 | 1.28 ± 0.34 |
| 0.5 | 1.23 | 1.15 | 1.04 | 1.14 ± 0.056 |
| 0.4 | 0 | 1.00 | 1.50 | 0.83 ± 0.44 |
| 0.3 | 0 | 1.00 | 0.89 | 0.63 ± 0.32 |

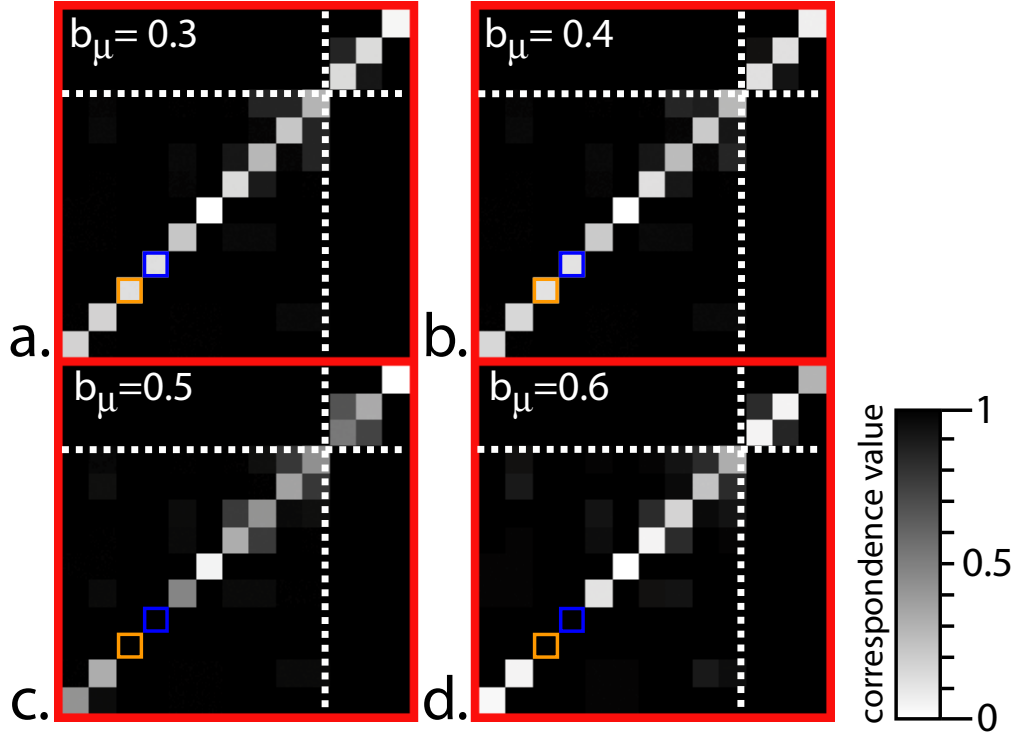


Figure 7.7: Trinary correspondence matrices calculated with the query landmarks placed at their estimated locations while varying the scale term of the unary metric b_μ . The orange square indicates the nodes extracted from the right eyeball, and the blue square indicates the nodes extracted from the left eyeball. The upper right partition bound by the dashed white lines is the landmark correspondence matrix.

7.5 LOCATION AND SCALE VARIATION

In this fifth experiment, both location and scale variation was introduced. The eyeball nodes were displaced laterally by 10 voxels, and the scale of each eyeball node was increased by 10 voxels. Correspondences were computed with $b_\mu = 0.3$ and $b_\mu = 0.4$ since these values yielded sub-pixel accuracy for the same scale perturbation in the previous experiment.

With $b_\mu = 0.4$, the landmark errors for the “right eyeball,” “left eyeball,” and “brainstem” landmarks were 3.76, 3.11, and 1.90 voxels, respectively with an average landmark error of 2.92 ± 0.54 voxels. For $b_\mu = 0.3$, landmark errors for the “right eyeball,” “left eyeball,” and “brainstem” landmarks were 3.37, 3.03, and 1.48 voxels, respectively with an average landmark error of 2.63 ± 0.58 voxels. With either parameter value, an approximate 3 voxel error for estimating the location of the “eyeball” landmarks is probably acceptable considering that the eyeballs are 30 voxels in diameter.

7.6 COMPARISON TO MANUAL LANDMARK IDENTIFICATION

This final experiment is an attempt to determine acceptable error measures for automated landmark placement by comparison to the accuracy in manually identifying landmarks. In practice, it is common to utilize manual landmark placement as ground-truth if there is sufficient intra- and inter-user reliability since a parametric ground-truth is usually not available. In this experiment, an expert user (the author) manually identified the “right eyeball,” “left eyeball,” and “brainstem landmarks” in several sets of perturbed data. Only the author participated in this experiment because he designed the software interface, and knew the exact structure parameters and landmark locations as parametrically defined. Thus, errors can be directly attributed to the user’s ability to place landmarks and not any of the aforementioned technicalities.

7.6.1 Software Interface and Landmark Placement Protocol

The software interface displays the reference and query images as 3 orthogonal cross-sections adjacent to each other. The reference landmarks are overlayed on the reference image to provide a frame of reference to the user. A landmark is displayed as an orange point with a translucent blue sphere centered at that point. The user interacts with the query image to place a landmark with a single mouse-click. To isolate a desired location in the query image, the user is able to rotate and translate the image volume, slice through the volume with the 2D cross-sections, and zoom the camera in and out.

The protocol for placing a landmark relied on first locating a desired landmark in the reference image. After appropriately orienting the volume, a cross-section was translated until the orange point just became visible. The same cross-section was then sought in the query image by identifying similar structure geometry in the image slice. This was repeated for other cross-sections to further isolate the area of interest. The landmark of interest was then placed with a click of a mouse button. Landmarks were placed by observing the landmark location in the reference image and using prior information about the location of the landmark. For instance, the eyeball landmarks are located exactly in the center of the eyeballs. The brainstem landmark is on the anterior surface of the brainstem and the brainstem is easily located by slicing through the sagittal plane.

7.6.2 Empirical Results

In the first study, two trials were performed following the same structure displacement for the experiment performed in Section 7.1, i.e., the right eyeball was displaced at 5 voxel intervals from the original location along each direction of the cardinal axes at a maximum displacement of 10 voxels. In *trial 1* the reference and query images were identical, providing a baseline comparison for how well landmarks could be manually placed in the absence of geometric perturbations. In *trial 2*, the query image was the perturbed version of the reference image as in the original experiment. The time to place landmarks was not recorded, but it generally took less than 30 seconds to place all 3 landmarks.

For displacement along a given axis, the average of landmark errors was computed for each manual trial. Shown in Figs. 7.8-7.10 are the results for the manual trials compared to that of the automated trials. The error bars represent the standard error of the mean.

Placing the “brainstem” landmark proved to be more difficult than the others since there are no structure details to provide image cues. The automated method was nearly as accurate in estimating the location of the “eyeball” landmarks, and more accurate in estimating the location of the “brainstem” landmark.

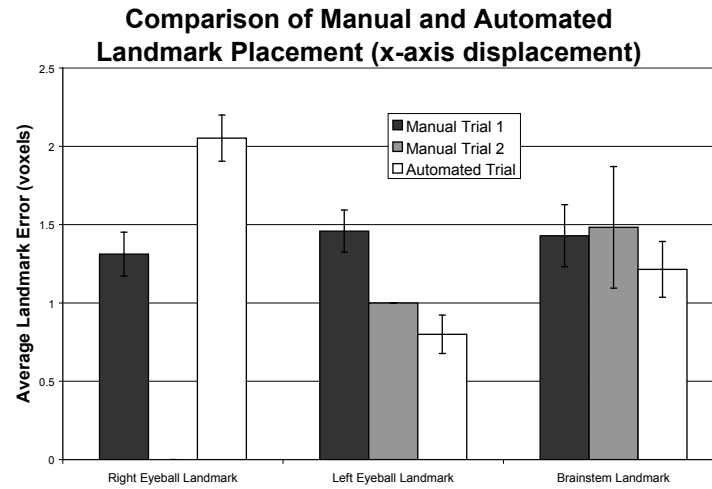


Figure 7.8: Average landmark error for each trial as the right eyeball was displaced along the x-axis. Note that there was no error when placing the “right eyeball” landmark during the second manual trial.

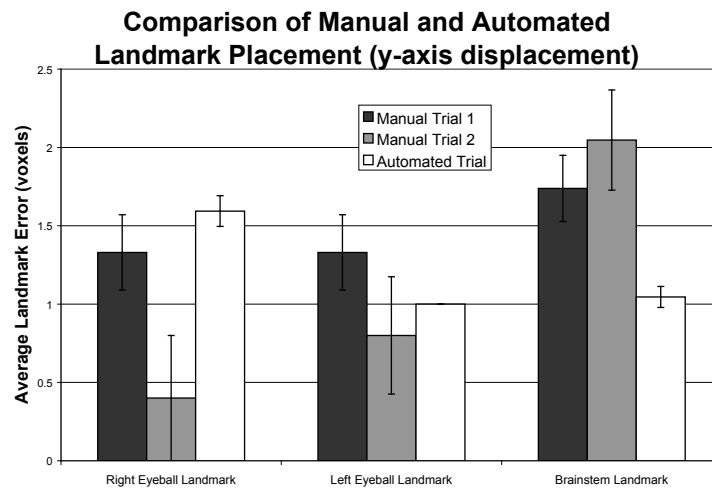


Figure 7.9: Average landmark error for each trial as the right eyeball was displaced along the y-axis

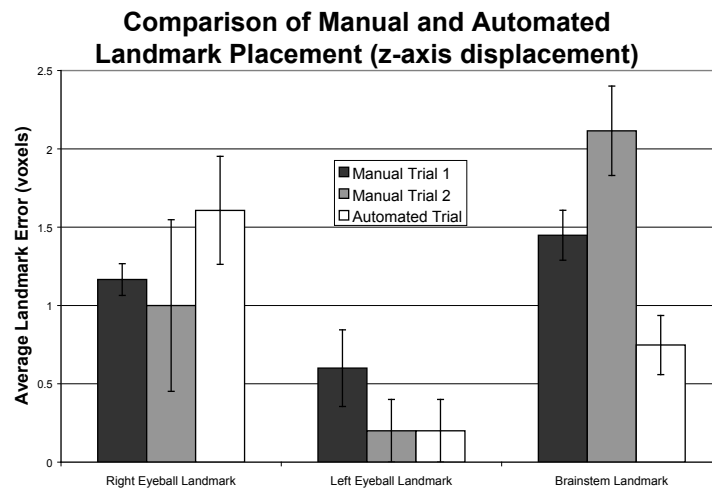


Figure 7.10: Average landmark error for each trial as the right eyeball was displaced along the z-axis

In this next study, manual landmark placement for medial and lateral eyeball displacement was investigated and compared to the results obtained with the automated method in Section 7.2. Shown in Figs. 7.11 and 7.12 are the average landmark errors for each of the trials. Manual landmark placement proved to yield very accurate results. The automated method had significantly larger error for most landmarks, but still a maximum error of less than 3 voxels.

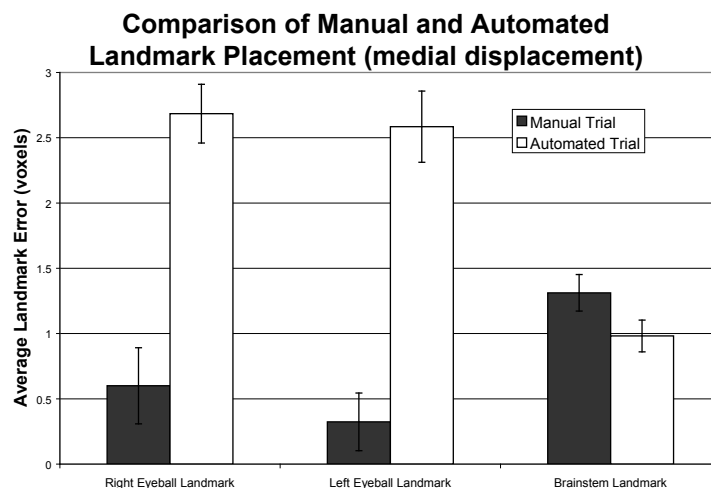


Figure 7.11: Comparison of manual versus automated method of estimating landmark location in the presence of medial displacement of the eyeballs. The average of landmark errors is shown with error bars depicting standard error of the mean.

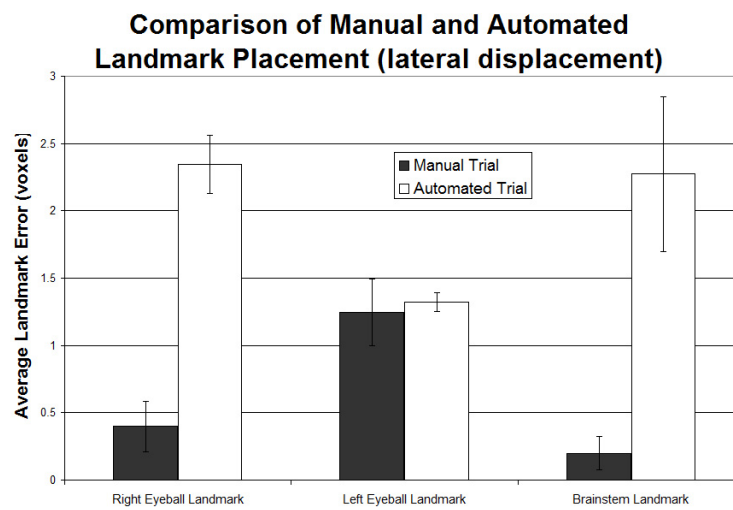


Figure 7.12: Comparison of manual versus automated method of estimating landmark location in the presence of lateral displacement of the eyeballs. The average of landmark errors is shown with error bars depicting standard error of the mean.

In this study, the diameter of both eyeballs was increased by 10 voxels and they were laterally displaced by 10 voxels. Landmarks were manually placed 5 times and compared to the automated results presented in Section 7.5. Shown in Fig. 7.13 is the average landmark error for the 5 manual trials compared to the landmark error for the automated method computed with $b_\mu = 0.3$ and $b_\mu = 0.4$. The manual method yielded an approximate 1 voxel accuracy for each landmark. The automated method produced an approximate 3.5 voxel accuracy for the “eyeball” landmarks and 2 voxel accuracy for the “brainstem” landmark.

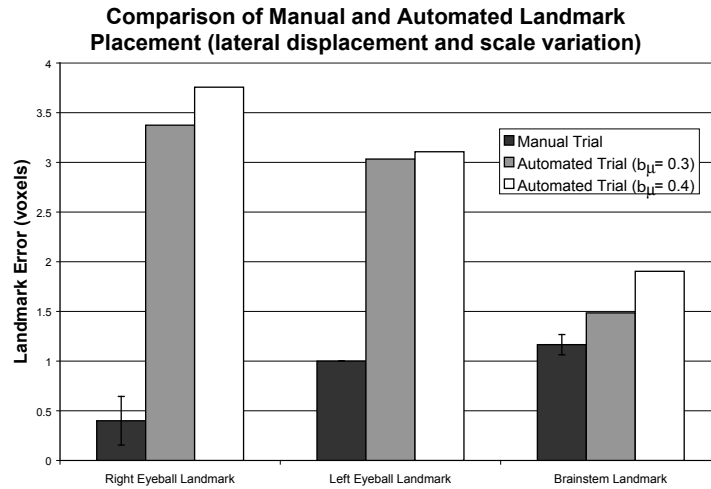


Figure 7.13: Comparison of methods for estimating landmark location in the presence of lateral displacement and scale variation of the eyeballs. The average of landmark error is shown for 5 manual trials. Two automated trials are shown with $b_\mu = 0.3$ and $b_\mu = 0.4$.

8.0 PROGRAMMING TECHNIQUES

Programming tasks were achieved with the C++ programming language and various open-source toolkits. Image analysis related methods were developed with the Insight Toolkit (ITK) to utilize ITK's image architecture and vast array of imaging base classes [115, 116]. The ITK project started in 1999 by the National Library of Medicine (NLM) for the development of an open-source, cross-platform segmentation and registration toolkit. The Insight Software Consortium originally consisted of 6 prime contractors and several subcontractors, which included the University of Pittsburgh. As a contributing developer and user of ITK since its conception, the coding standards established by ITK were followed for the possible contribution of this research to ITK for use by the image analysis community.

Graphical User Interfaces (GUI) were programmed with the Fast Light Toolkit (FLTK), which provides modern GUI functionality and supports 3D graphics [117]. The FLTK-based interfaces were produced with a freely available, GUI-based, user interface builder called FLUID. Data was visualized with the Visualization Toolkit (VTK) [118], which supports a wide variety of visualization algorithms and advanced modeling techniques.

9.0 CONCLUSIONS

Developed in this dissertation are novel methods for determining inter-image correspondences between cliques of features and inferring the location of anatomical landmarks from such correspondences. Correspondences are determined from a hierarchy of metrics that quantify inter-image clique similarity by properties inherent to the features and the geometric relationship between the comprising features and landmarks. The correspondence approach differs from others in that the medial features describe shape properties in addition to intensity, and the metrics rely on more than a distance measure. Both the correspondence and landmark estimation methods are automated and invariant to global rotation and translation.

Results for the methods developed in this dissertation are encouraging. The correspondence hierarchy was first validated on identical images with ROC curve analysis, then later validated for images containing structural variation. This validation process was based on the accuracy in inferring the location of anatomical landmarks for various structure perturbations. Consequently, it was shown that anatomical landmarks could be accurately placed in an image, with results comparable to manual placement of landmarks.

The remainder of this chapter summarizes the results and revisits the claims made in Chapter [1.2](#), concluding with potential future directions for the research.

9.1 CLAIMS REVISITED

The claims made in Chapter 1.2 are revisited and reviewed in this section.

Claim 1 By using a hierarchy of similarity metrics, inter-image correspondences between cliques of features can be determined.

Towards the goal of automatically determining feature correspondences, the geometric relationships between features was utilized as constraints and a hierarchy of similarity metrics were developed. The correspondence framework was first demonstrated with identical images containing two concentric ellipsoids in Chapter 3.6. In two separate experiments, global rotational and translational invariance was demonstrated with images containing rotated and translated concentric ellipsoids (Chapter 3.6.1.2). The accuracy of the framework was assessed with ROC curve analysis on two simulated MRI brain images, and results showed that accuracy increased with each successive metric in the correspondence hierarchy.

In medical image analysis, feature-based methods for determining correspondences typically operate by mapping the features into a common coordinate system, and then minimizing a global distance metric for the set of features. The correspondence framework developed here also uses distance as a component in its metrics, but is restricted to only local intra-clique distance. By utilizing clique-based metrics, the correspondence framework is less sensitive to outliers and features absent from one of the images. Common approaches to the correspondence problem utilize intensity-based features such as boundary points, ridges, corners etc., making them sensitive to image noise. The features used here also describe local shape properties providing a more robust means for determining correspondences.

Claim 2 By incorporating global geometric constraints into the correspondence hierarchy, correspondence accuracy can be improved.

A method was developed in chapter 4 for improving upon correspondences by utilizing global geometric information. The method operates by first computing an affine trans-

form that is implied by a triplet correspondence. The transform is then applied to every query image triplet and a measure of fit is calculated with the corresponding reference image triplets. These measures of fit are accumulated for each triplet with a high fitness measure being consistent with the geometry of other cliques. Thus, this method employs a loose constraint of a global affine transformation by consensus. By empirical verification in Chapter 4.3, it was shown with ROC curve analysis on two simulated MRI brain images, with the query image being a transformed version of the reference image, that the addition of this method improves the accuracy of the correspondence hierarchy.

Claim 3 By defining a parametric model, it is possible to establish ground-truth for validating correspondences in the presence of variations in structure geometry.

A parameterized model emulating a CT scan of a human brain was defined in Chapter 5. Brain structures were modeled with parametrically defined shape primitives, and combinations of such shapes with computational shape geometry. Structures encompassing the model included the cortices, eyeballs, optic nerves, brain stem, cerebellum, lateral ventricles, and thalami. The model was developed to permit systematic perturbation of structure geometry to assess correspondence performance under such conditions. Also provided by the model was an automated means for defining anatomical landmarks and establishing their ground-truth in a distorted query image, which played a crucial role in correspondence validation.

Claim 4 Landmark locations can be inferred from feature correspondences and automatically placed in a query image to validate the correspondence method, as well as provide anatomical information to the user.

A technique for estimating the location of anatomical landmarks in an image was developed in Chapter 6. Analogous to atlas-based registration, a reference image containing predefined landmarks is used to place those landmarks in a new query image. The method relies on determining correspondences between cliques extracted from the reference and query images, with each clique containing features and landmarks. Thus, the cliques describe the geometric relationships between constituent features and landmarks.

By adding landmarks to the cliques along with the features, landmark location can be inferred by means of an optimization procedure that maximizes a landmark correspondence figure of merit. Two optimization procedures were developed. In Chapter 6.5.1, the brute force method was developed and tested on two images from the brain model. The brute force method provided critical insight into the optimization problem, namely, validation of smoothness and unimodality, which allowed for the development of a faster gradient ascent-like method in Chapter 6.5.2. With this new optimization technique, as was shown in Chapter 7, correspondences were validated in the presence of geometric variation by their accuracy in inferring landmark location.

Claim 5 Correspondences can be validated for differing images by their accuracy in inferring the location of anatomical landmarks in a query image.

Since it is difficult to establish ground-truth for correspondences in the presence of geometric variation, a theory based on the parametric model was developed for validating correspondences under such conditions. The parametric model not only allowed for systematic perturbation of the data, but provided an automated means to parametrically define ground-truth for landmarks. Thus, an error measure could be calculated between the estimated landmark locations and their ground-truth as a function of geometric variations between structures. In Chapter 7, results were presented for a number of structural perturbations that indicated correspondences can infer landmark locations accurately, and nearly as well as an expert user manually identifying landmark locations.

9.2 FUTURE RESEARCH DIRECTIONS

Central to this dissertation was the development of a novel framework for determining feature-based correspondences. These medial nodes based on core atoms were utilized because they are typically sparse and provide a measure of local shape properties. Medial nodes have inherent limitations, but permitted the development of the correspondence framework in anticipation of more robust medial features being developed by others in my laboratory. In order for the correspondence framework, and consequently the landmark estimation method, to be viable for analyzing medical images, more robust medial features are needed. In this light, a framework called *Shells and Spheres* for extracting a more complete description of the medial manifold has been recently developed by Cois *et al.* [119].

Briefly, Shells and Spheres is based on a sphere map, which consists of a sphere at each voxel in the image. The goal is to optimize the sphere map such that each sphere is grown to touch but not cross a structure boundary. Sphere growth is governed by computing *Variable Scale Statistics* for voxels within spherical regions and voxels among adjacent spherical regions. Once the sphere map is optimized, measures of boundariness and medialness can be computed. Developing new features from these boundariness and medialness measures and integrating these features into the landmark placement procedure will benefit it significantly, hopefully leading to a practical system for placing landmarks in medical images.

To demonstrate the potential benefits of this improved technique, landmark locations were estimated on a contrast-enhanced CT chest image (courtesy of Robert Gorman, M.D. and Michael Sacks, Ph.D.). In this experiment, medial nodes were extracted from a segmentation of the aorta produced from the Shells and Spheres medialness measure. After the sphere map was optimized, the segmentation was produced by labeling voxels within spheres and neighboring spheres centered at points of significant medialness. From a binary image of the segmentation, 146 medial nodes were extracted. Since the aorta is locally cylindrical, slab nodes were filtered out from this set of nodes leaving 58 (mostly cylindrical) nodes. In the sagittal plane roughly bisecting the descending aorta, three landmarks were placed at arbitrary locations on the surface of the aortic arch. The CT image is shown with a surface model of the segmentation in Fig. 9.1. Also shown are the set of medial nodes extracted

from the segmentation image and landmarks placed on the aortic arch. Correspondences were then determined and landmark locations were estimated. The unary, binary, and ternary correspondence matrices are shown in Fig. 9.2. The errors in estimating landmark location were 3.03, 2.53, and 2.30 voxels for an average error and standard deviation of 2.62 ± 0.38 voxels. Due to the large number of nodes and combinatorial complexity of forming cliques (1,653 possible doublet combinations and 30,856 triplet combinations), several hours were required to estimate landmark locations. The computational requirements of the correspondence framework and landmark placement method are addressed later in this section.

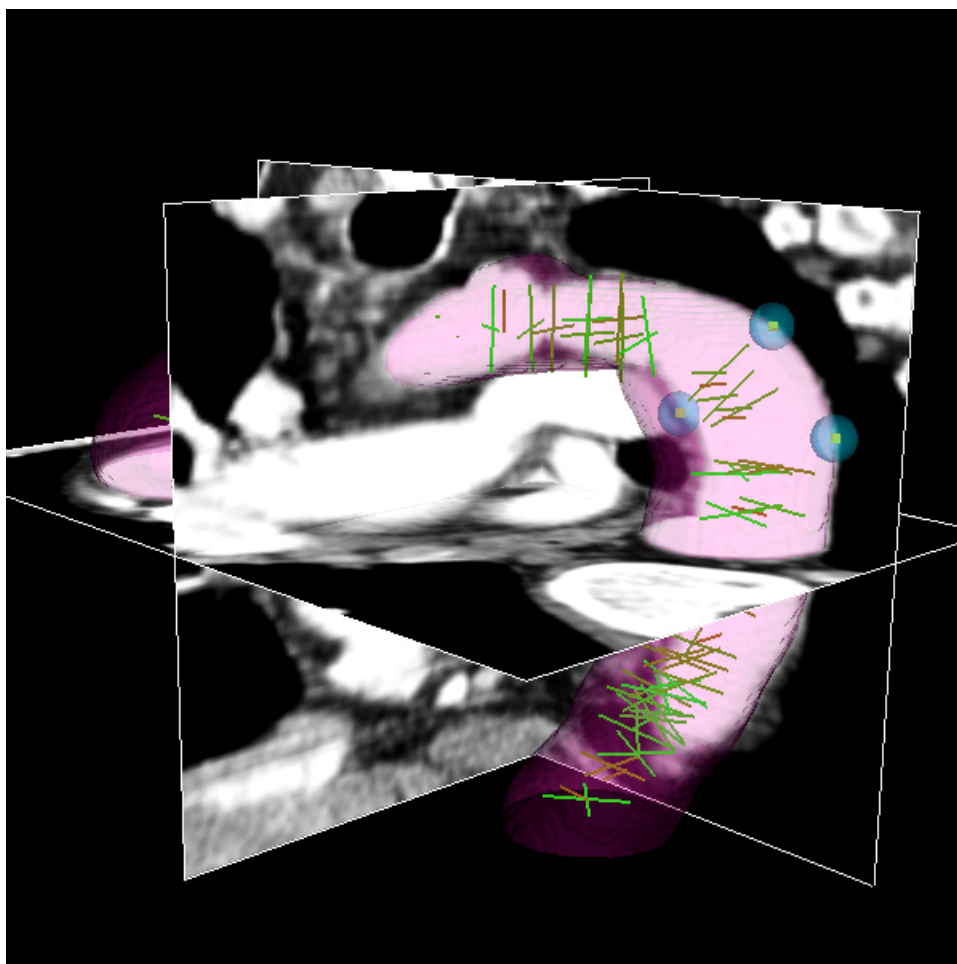


Figure 9.1: Image of the contrast-enhanced CT chest scan with a surface model of the segmentation overlaid as translucent pink. Also shown are the set of extracted medial nodes and aortic arch landmarks.

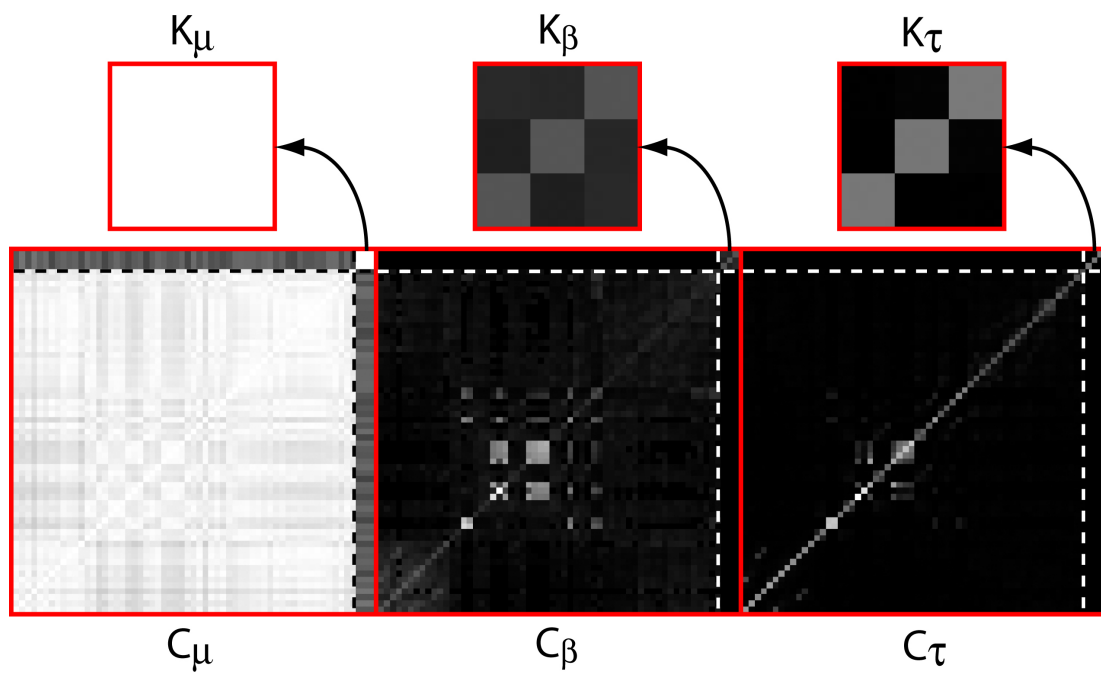


Figure 9.2: Correspondence matrices computed for nodes extracted from the aorta segmentation and landmarks placed on the aorta aortic arch. Each landmark correspondence matrix is shown magnified above the respective correspondence matrix.

Validation of the correspondence framework first relied on using identical images. By incorporating anatomical landmarks into the correspondence framework, correspondences in the presence of structure variation could be validated by establishing ground-truth with a parametric model. In future medical applications, landmark ground-truth will need to be established by an expert user. Not only does estimating the location of anatomical landmarks provide a means for correspondence validation, but it also labels anatomical regions. Such information can be utilized by healthcare professionals to assist them in locating anatomical structures or features. Anatomical landmarks can also be used to assist existing segmentation and registration methods. For example, landmarks can be used to initialize an active contour or region-growing based segmentation algorithm. In segmentation-based registration, landmarks can be used to initialize the model in the query image. Once correspondences between medial nodes and landmarks have been established either may prove directly useful in a registration method.

Towards improving correspondences and consequently automated landmark placement, the triplet transform metric should be included in the correspondence hierarchy. Due to the processing time required for computing the triplet transform metric, it was left out of the iterative landmark optimization procedure. An initial attempt was made to improve computation speed by computing transforms for only those triplets with a single triplet correspondence. This proved to slightly reduce computation time. Further improvement may be accomplished by computing only enough transforms to accomplish a pre-determined amount of accuracy or error. When analyzing a large number of features, triplet correspondences could also benefit from computational improvements by removing features with low correspondences prior to forming triplets.

In the presence of scale variation, it was shown that correspondence specificity could be fine-tuned by adjusting the parameters of the unary metric. For applying the framework to various image modalities and target structures, optimizing the metric parameters could certainly be investigated further.

The automated method for estimating landmark location proved to be reasonably accurate and comparable to manual trials. Major advantages of the automated method are objectivity, consistency, and the lack of user interaction with the query image. Computation

times are acceptable for most applications. For those applications that require fast landmark estimation, speed improvements for the computation of triplets will help address this issue. Moreover, the landmark optimization procedure could possibly be made faster.

To improve the accuracy of landmark estimates, a correlation-based approach could be utilized after the optimization procedure to further isolate landmark placement based on image intensity. A small window could be moved around each query landmark estimate until it correlates with a small window around the corresponding reference landmark. Success of this method would depend on local structure detail, which is the primary reason why it was not developed in this dissertation, since the synthetic images contained purely homogeneous structures.

Landmarks were defined as being small spheres. Future research could explore different representations of landmarks based on prior information, e.g., landmarks associated with blood vessels are cylindrical, or image information, e.g., landmarks take on properties similar to neighboring features.

To further improve the correspondence and landmark placement methods, it would be beneficial to observe an expert while they examine medical images. Learning their thought processes while locating structures could shed some light on utilizing image cues that are commonly used for locating specific structures as *a priori* information.

Developing these improvements and incorporating them into the correspondence and landmark placement framework would facilitate their application to medical images. Thus, accurate, reliable, and non-subjective quantitative measures and qualitative information could be delivered to the clinician to assist in making diagnoses, planning treatment, or identifying relationships between structure properties and pathologies.

APPENDIX A

SIGNATURE INVARIANCE TO ROTATION IN THE DEGENERATE SUBSPACE

In this section, the claim that a signature is independent of the global orientation of a node pair is investigated. A medial node's orientation is defined by its eigenvectors as defined relative to the cardinal coordinate system. However, the eigenvalues themselves are independent of that orientation. When two or more eigenvalues are equal, a degenerate subspace exists that may be expressed by matrix \mathbf{A} whose columns are the corresponding eigenvectors $\mathbf{A} = [\hat{\mathbf{a}}_i \ \hat{\mathbf{a}}_j \ \cdots \ \hat{\mathbf{a}}_N]$.

Rotating a node pair within the subspace defined by the degenerate eigenvectors of node 1 will redefine the eigenvectors as $\mathbf{E} = \mathbf{T}_R \mathbf{A} = [\hat{\mathbf{e}}_i \ \hat{\mathbf{e}}_j \ \cdots \ \hat{\mathbf{e}}_N]$, where \mathbf{T}_R is a pure rotation matrix. The cumulative NCS between node 1 and node 2 will remain unchanged. The case of two eigenvalues being equal is demonstrated. For the case of a 2D degenerate subspace, rotation within that subspace is represented by

$$\begin{aligned}\hat{\mathbf{e}}_i &= \hat{\mathbf{a}}_i \cos(\phi) - \hat{\mathbf{a}}_j \sin(\phi) \\ \hat{\mathbf{e}}_j &= \hat{\mathbf{a}}_i \sin(\phi) + \hat{\mathbf{a}}_j \cos(\phi),\end{aligned}\tag{A.1}$$

where ϕ is a parameter of \mathbf{T}_R , and $\cos^2(\phi) + \sin^2(\phi) = 1$. The NCS for the two eigenvalues will add to the signature at the same location on the λ -axis, namely the $\lambda_i = \lambda_j$, because the unit step functions will fuse into a single unit step function, whose height is $(\hat{d}_i)^2 + (\hat{d}_j)^2$ independent of rotation within the degenerate subspace.

By the Pythagorean Theorem, the quantity $(\hat{d}_i)^2 + (\hat{d}_j)^2$ represents the squared length of the hypotenuse, namely, component of \mathbf{v} projected onto the plane defined by the eigenvectors. This quantity is a constant, independent of which coordinate system was used to calculate it. This can be proven by calculating $(\hat{d}_i)^2 + (\hat{d}_j)^2$ in terms of \mathbf{E} and \mathbf{A} , and showing them to be equal:

$$\begin{aligned}
(\hat{\mathbf{v}} \cdot \hat{\mathbf{e}}_i)^2 + (\hat{\mathbf{v}} \cdot \hat{\mathbf{e}}_j)^2 &= \\
(\alpha \hat{\mathbf{v}} \cdot \hat{\mathbf{a}}_i + \phi \hat{\mathbf{v}} \cdot \hat{\mathbf{a}}_j)^2 + (-\phi \hat{\mathbf{v}} \cdot \hat{\mathbf{a}}_i + \alpha \hat{\mathbf{v}} \cdot \hat{\mathbf{a}}_j)^2 &= \\
(\alpha^2 + \phi^2) \left[(\hat{\mathbf{v}} \cdot \hat{\mathbf{a}}_i)^2 + (\hat{\mathbf{v}} \cdot \hat{\mathbf{a}}_j)^2 \right] + (2\alpha\phi - 2\alpha\phi) \left[(\hat{\mathbf{v}} \cdot \hat{\mathbf{a}}_i)^2 + (\hat{\mathbf{v}} \cdot \hat{\mathbf{a}}_j)^2 \right] &= \\
(\hat{\mathbf{v}} \cdot \hat{\mathbf{a}}_i)^2 + (\hat{\mathbf{v}} \cdot \hat{\mathbf{a}}_j)^2 &= \\
(\hat{d}_i)^2 + (\hat{d}_j)^2, & \quad (\text{A.2})
\end{aligned}$$

where the last two equations are equal as defined in Eq. 3.5.

APPENDIX B

COMPUTING CORRESPONDENCES EFFICIENTLY

Inherent to the correspondence framework, each step through the correspondence hierarchy reduces the number of nodes to be considered for correspondence. However, the combinatorial complexity of forming cliques dramatically increases with higher order cliques, and thus increases the number comparisons to be made. The maximum number of unique cliques that can be formed from a set of m nodes is calculated as the total number of combinations that k nodes can be taken from the set, i.e.,

$$\binom{m}{k} = \frac{m!}{k!(m-k)!}, \quad (\text{B.1})$$

where $k = 1$ for singlets, $k = 2$ for doublets, and $k = 3$ for triplets. In the face of this potential computational nightmare, the development of efficient data structures and algorithms was necessary.

Suppose each image is $V \times V \times V$ in size, and thus consists of V^3 voxels. The number of medial nodes extracted from each image is quite sparse compared to the V^3 voxels in the image, that is $M_R \ll V^3$ and $M_Q \ll V^3$. Each set of medial nodes is stored in a linked-list. Recall from that the set of nodes extracted from images R and Q are denoted as Ψ_R and Ψ_Q , respectively. When calculating singlet correspondences, the unary metric is calculated by simply indexing the medial node lists. A total of $M_R \times M_Q$ unary metric calculations are made. Once these correspondences are calculated they are stored in the unary correspondence matrix.

Doublet and triplet correspondences as evaluated by the corresponding binary or trinary metric, are also stored in a binary or trinary correspondence matrix. However, an additional special data structure is required to organize these correspondences between cliques consisting of more than one node. The data structure serves as a mechanism for efficiently iterating through nodes for computing clique correspondences.

The correspondence data structure comprises several embedded linked-lists that store node indices from Ψ_R and Ψ_Q . The first layer of the data structure consists of a linked-list referred to as the “base-list.” The base-list is the same length as the medial node list from the reference image. Embedded within each element of the base-list is a linked-list called the “secondary-list,” in which the index of a clique’s base node from the reference image is stored. Each element within a given secondary-list, contains the same base node index, i.e., the node in the corresponding element of the base list (see Fig. B1). Described below is the relevance of this when iterating the doublet data structure for determining triplet correspondences. A doublet’s other constituent nodes are stored in a “correspondence-list,” which is embedded at a location specific to that doublet in the secondary-list. The correspondence-list also contains a clique data structure, which stores the index of each corresponding doublet in the query image.

To illustrate the use of the correspondence data structure, the algorithm for determining doublet correspondences is described. In this algorithm, doublet distance is used as a constraint prior to computing the binary metric (Eq. 3.11). For each image, doublet distance is computed between all pairs of nodes. Doublet distances are then compared between images. If pairs of doublets have similar doublet distances, the unary metric is compared for individual nodes within the doublets. If the unary metrics are similar, then the binary metric is calculated. If it satisfies the binary metric threshold, then both cliques are stored in the doublet data structure. The index of a “base node” from the reference image doublet is stored in the secondary list. The index of the second node in the reference image doublet is stored in its particular correspondence-list. The indices of the corresponding doublets in the query image are stored as a clique index structure in the correspondence-list.

A simplified correspondence data structure containing doublet correspondences is illustrated in Fig. B1. Note that for the purposes of clarity, node indices are indicated as $n_{\alpha,i}$.

The base-list is of length 7, which is equivalent to the number of nodes from the reference image. Shown in yellow is a pair of corresponding doublets. The base node of the reference image doublet $n_{R,4}$ is stored in the secondary-list, which is located at *index* 4 of the base-list. The second node's index of the reference image doublet, $n_{R,3}$ is stored in the correspondence-list. The corresponding clique from the query image, $C_{Q,7}$ is also stored in the correspondence-list.

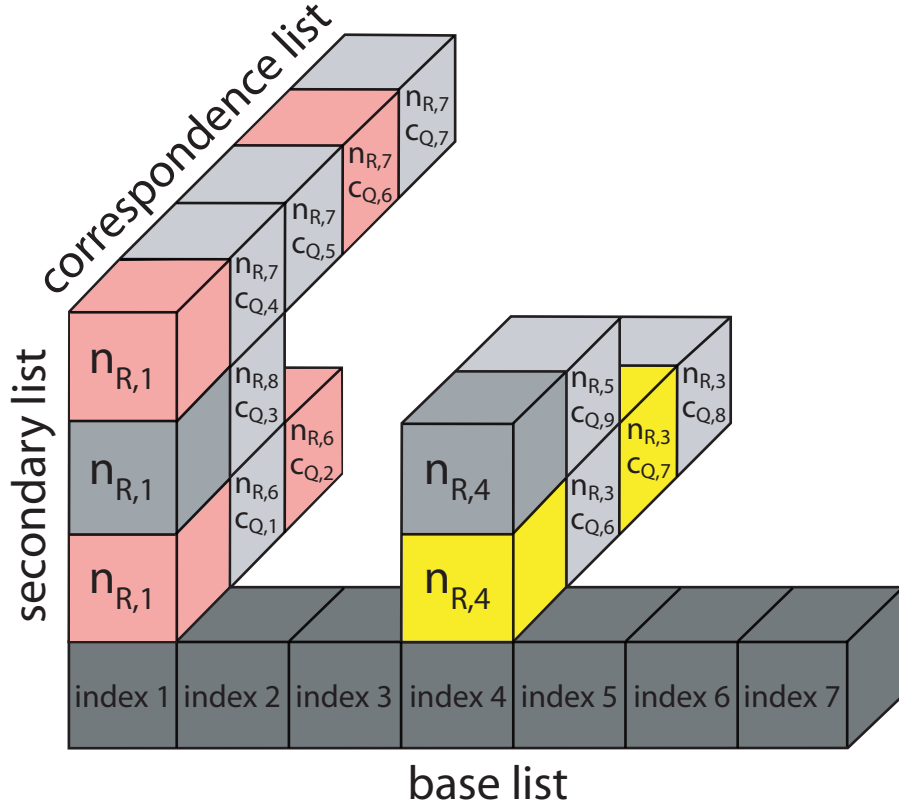


Figure B1: Data structure for efficiently storing doublet correspondences. Shown in yellow are corresponding doublets, and shown in pink are two pairs of corresponding doublets to be considered for triplet formation.

Determining triplet correspondences is straightforward with the doublet data structure. Recall that a triplet consists of two doublets that share a base node. Such doublets are easily indexed from within the same secondary list. A triplet is shown as pink in Fig. B1. The secondary-list at *index* 1 of the base-list consists of node $n_{R,1}$. Doublets with base node $n_{R,1}$ are accessed by iterating through the respective correspondence lists. Two doublets are

shown in pink with base node $n_{R,1}$ in the secondary list. Doublets $(n_{R,1}, n_{R,6})$ and $(n_{R,1}, n_{R,7})$ therefore comprise the triplet $(n_{R,1}, n_{R,6}, n_{R,7})$. To test for triplet correspondence, these pair of doublets would be compared to the triplet that comprises doublets $C_{Q,2}$ and $C_{Q,6}$ in the query image. Nodes are ordered in the data structure to prevent entering the same doublet more than once. With this data structure, cliques of size up to 3 are easily represented, and the computational load for covering clique combinations is kept to a minimum. The data structure could be expanded to represent bigger cliques, but computation would skyrocket. In fact, in chapter 4, a fourth node was considered via an implied transform to avoid using 4-membered cliques.

BIBLIOGRAPHY

- [1] K. T. Dussik. On the possibility of using ultrasound waves as a diagnostic aid. *Neurology and Psychiatry*, 174(153-168), 1942.
- [2] K.T. Dussik, F. Dussik, and L. Wyt. Auf dem wege zur hyperphonographie des gehirnes. *Wien med Wschr*, 97:425–429, 1947.
- [3] P. C. Lauterbur. Image formation by induced local interactions- examples employing nuclear magnetic resonance. *Nature*, 242(5394):190–191, 1973.
- [4] M. Stytz, G. Frieder, and O. Frieder. Threedimensional medical imaging: Algorithms and computer systems. *ACM Computing Surveys*, 23(4):421–499, 1991.
- [5] R. Bracewell. *Two-Dimensional Imaging*. Prentice Hall, New Jersey, 1995.
- [6] O. T. von Ramm and S. W. Smith. Real time volumetric ultrasound imaging system. *SPIE Medical Imaging IV: Image Formation*, 1231:15–22, 1990.
- [7] O.T. von Ramm, H.A. Pavy, S.W. Smith, and J. Kisslo. Real-time, three-dimensional echocardiography: the first human images. *Circulation*, 84, suppl. II(4):685, 1991.
- [8] D. Mitchell. *MRI Principles*. W.B. Saunders Company, Philadelphia, 1999.
- [9] C.N. McEwan and K. Fukuta. Recent advances in medical imaging: Surgery planning and simulation. *World J. of Surgery*, 13(4):343–348, 1989.
- [10] A. Liu, F. Tendick, K. Cleary, and C. Kaufmann. A survey of surgical simulation: applications, technology, and education. *Presence: Teleoperators and Virtual Environments*, 12(6):599 – 614, 2003.
- [11] P. M. Black, T. Moriarty, E. A. Alexander, P. Stieg, E. J. Woodard, and L. Gleason. Development and implementation of intraoperative magnetic resonance imaging and its neurosurgical applications. *Neurosurgery*, 41:831–45, 1997.
- [12] R. M. Satava. Emerging technologies for surgery in the 21st century. *Arch Surg*, 134(11):1197–1202, 1999.

- [13] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1993.
- [14] R. O. Duda and P. E. Hart. *Pattern Classification*. John Wiley and Sons, second edition, 2000.
- [15] D. L. Pham, C. Xu, and J. L. Prince. Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2:315–337, 2000.
- [16] J. S. Duncan and N. Ayache. Medical image analysis: Progress over two decades and the challenges ahead. *Pattern Analysis and Machine Intelligence*, 22(1):85–106, 2000.
- [17] S. U. Lee, S. Y. Chung, and R. H. Park. A comparative performance study of several global thresholding techniques for segmentation. *Comput. Vision, Graphics, Image Proc.*, 52(2):171–190, 1990.
- [18] A. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1986.
- [19] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.
- [20] K. S. Fu and J. K. Mui. A survey of image segmentation. *Pattern Recognition*, 13(1):3–16, 1981.
- [21] G. Matheron. *Random Sets and Integral Geometry*. John Wiley and Sons, 1975.
- [22] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, New York, 1982.
- [23] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.
- [24] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.
- [25] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1(2):91–108, 1996.
- [26] A. Dervieux and F. Thomasset. A finite element method for the simulation of rayleigh taylor instability. *Lecture Notes in Mathematics*, 771:145–158, 1979.
- [27] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Internat. J. Comput. Vision*, 22:61–79, 1997.
- [28] R. Malladi, J. Sethian, and B.C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(2):158–171, 1995.

- [29] T. F. Cootes, A. Hill, and J. Haslam. Use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6):355–365, 1994.
- [30] T. F. Cootes, C. Beeston, G. J. Edwards, and C. J. Taylor. A unified framework for atlas matching using active appearance models. *IPMI '00*, LNCS 1613:322–333, 1999.
- [31] A. Hill, T. F. Cootes, and C. J. Taylor. Active shape models and the shape approximation problem. *Image and Vision Computing*, 14:601–607, 1996.
- [32] J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.
- [33] C. R. Maurer and J. M. Fitzpatrick. A review of medical image registration. In R. J. Maciunas, editor, *Interactive Image-Guided Neurosurgery*, pages 17–44. American Association of Neurological Surgeons, Park Ridge, IL, 1993.
- [34] C. Chen, C. A. Pelizzari, G. T. Y. Chen, M. D. Cooper, and D. N. Levin. Image analysis of pet data with the aid of ct and mr images. *Information processing in medical imaging*, pages 601–611, 1987.
- [35] D. N. Levin, C. A. Pelizzari, G. T. Y. Chen, C. T. Chen, and M. D. M. D. Cooper. Retrospective geometric correlation of mr, ct, and pet images. *Radiology*, 169(3):817–823, 1988.
- [36] N. Ayache, A. Guezic, J. Thirion, A. Gourdon, and J. Knoploch. Evaluating 3d registration of ct-scan images using crest lines. In D. C. Wilson and J. N. Wilson, editors, *Mathematical methods in medical imaging*, volume 2035, pages 60–71, Bellingham, WA, 1993. SPIE Press.
- [37] J. C. Gee, M. Reivich, and R. Bajcsy. Elastically deforming 3d atlas to match anatomical brain images. *Journal of computer assisted tomography*, 17(2):225–236, 1993.
- [38] R. Szeliski and S. Lavalley. Matching 3-d anatomical surfaces with non-rigid deformations using octree-splines. *International journal of computer vision*, 18(2):171–186, 1996.
- [39] R. Bajcsy, R. Lieberman, and M. Reivich. A computerized system for the elastic matching of deformed radiographic images to idealized atlas images. *Journal of computer assisted tomography*, 7(4):618–625, 1983.
- [40] J. Thirion. Non-rigid matching using demons. In *Computer vision and pattern recognition*, pages 245–251, Los Alamitos, CA, 1996. IEEE computer society press.
- [41] N. M. Alpert, J. F. Bradshaw, D. Kennedy, and J. A. Correia. The principal axis transformation method for image registration. *Journal of nuclear medicine*, 31:1717–1722, 1990.

- [42] P. K. Banerjee and A. W. Toga. Image alignment by integrated rotational and translational transformation matrix. *Physics in medicine and biology*, 39:1969–1988, 1994.
- [43] G. J. Ettinger, W. E. L. Grimson, T. Lozano-Perez, W. M. Wells III, S. J. White, and R. Kikinis. Automatic registration for multiple sclerosis change detection. In *Workshop on biomedical image analysis*, pages 297–306, Los Alamitos, CA, 1994. IEEE computer society press.
- [44] P. J. Slomka, G. A. Hurwitz, J. Stephenson, and T. Craddock. Automated alignment and sizing of myocardial stress and rest scans to three-dimensional normal templates using an image registration algorithm. *Journal of nuclear medicine*, 36:1115–1122, 1995.
- [45] C. Studholme, D. L. G. Hill, and D. J. Hawkes. Multi-resolution voxel similarity measures for mr-pet registration. In Y. Bizais and C. Barillot, editors, *Information Processing in Medical Imaging*, pages 287–298, Dordrecht, 1995. Kluwer Academic Publishers.
- [46] G. S. Cox and G. de Jager. Automatic registration of temporal image pairs for digital subtraction angiography. In M. H. Loew, editor, *Medical imaging: image processing*, volume 2167, pages 189–199, Bellingham, WA, 1994. SPIE Press.
- [47] B. A. Ardekani, M. Braun, B. F. Hutton, I. Kanno, and H. Ida. A fully automatic multimodality image registration algorithm. *Journal of computer assisted tomography*, 19(4):615–623, 1995.
- [48] T. Buzug and J. Weese. Improving dsa images with an automatic algorithm based on template matching and an entropy measure. In H. U. Lemke, M. W. Vannier, K. Inamura, and A. G. Farman, editors, *Computer assisted radiology*, volume 1124 of *Excerpta medica - international congress series*, pages 145–150, Amsterdam, 1996. Elsevier.
- [49] D. L. G. Hill and D. J. Hawkes. Medical image registration using voxel similarity measures. In *Applications of computer vision in medical image processing*, pages 34–37, Berlin, 1994. Springer-Verlag.
- [50] C. K. Hoh, M. Dahlbom, G. Harris, Y. Choi, R. A. Hawkins, M. E. Phelps, and J. Maddahi. Automated iterative three-dimensional registration of positron emission tomography images. *Journal of nuclear medicine*, 34:2009–2018, 1993.
- [51] G. E. Christensen, R. D. Rabbitt, M. I. Miller, S. C. Joshi, U. Grenander, T. A. Coogan, and D. C. van Essen. Topological properties of smooth anatomic maps. In Y. Bizais, C. Barillot, and R. di Paola, editors, *Information processing in medical imaging*, pages 101–112, Netherlands, 1995. Kluwer.

- [52] L. Junck, J. G. Moen, G. D. Hutchins, M. B. Brown, and D. E. Kuhl. Correlation methods for the centering, rotation, and alignment of functional brain images. *Journal of nuclear medicine*, 31:1220–1276, 1990.
- [53] J. B. A. Maintz, P. A. van den Elsen, and M. A. Viergever. Using geometrical features to match ct and mr brain images. In L. Beolchi and M. Kuhn, editors, *Medical imaging, analysis of multimodality 2D/3D images*, volume 19 of *Studies in Health, Technology and Informatics*, pages 43–52, Amsterdam, 1994. IOS Press.
- [54] P. Viola and W. M. Wells III. Alignment by maximization of mutual information. In *International conference on computer vision*, pages 16–23, Los Alamitos, CA, 1995. IEEE computer society press.
- [55] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multi-modality image registration by maximization of mutual information. In *Mathematical methods in biomedical image analysis*, pages 14–22, Los Alamitos, CA, 1996. IEEE computer society press.
- [56] A. Venot, J. L. Golmard, J. F. Lebruchec, L. Pronzato, E. Walter, G. Frij, and J. C. Roucayrol. Digital methods for change detection in medical images. In F. Deconinck, editor, *Information processing in medical imaging*, Dordrecht, 1983. Nijhoff.
- [57] P. Hua and I I. Fram. Feature-based image registration for digital subtraction angiography. In M. H. Loew, editor, *Medical imaging: image processing*, volume 1898, pages 24–31, WA, 1993. SPIE Press.
- [58] A. C. Evans, S. Marrett, L. Collins, and T. M. Peters. Anatomical-functional correlative analysis of the human brain using three dimensional imaging systems. In S. J. Dwyer III and R. J. Jost, editors, *Medical imaging: image processing*, volume 1092, pages 264–274, Bellingham, WA, 1989. SPIE Press.
- [59] D. L. G. Hill, D. J. Hawkes, J. E. Crossman, M. J. Gleeson, T. C. S. Cox, E. C. M. L. Brace, A. J. Strong, and P. Graves. Registration of mr and ct images for skull base surgery using pointlike anatomical features. *British journal of radiology*, 64(767):1030–1035, 1991.
- [60] S. Fang, R. Raghavan, and J. T. Richtsmeier. Volume morphing methods for landmark based 3d image deformation. In M. H. Loew and K. M. Hanson, editors, *Medical Imaging: Image processing*, volume 2710, pages 404–415, Bellingham, WA, 1996. SPIE Press.
- [61] T. Peters, B. Davey, P. Munger, R. Comeau, and A. Olivier. Three-dimensional multimodal imageguidance for neurosurgery. *IEEE Transactions on medical imaging*, 15(2):121–128, 1996.

- [62] Z. He, J. C. Maublant, J. C. Cauvin, and A. Veyre. Reorientation of the left ventricular long-axis on myocardial transaxial tomograms by a linear fitting method. *Journal of nuclear medicine*, 32:1794–1800, 1991.
- [63] F. Fontana, A. Crovetto, M. Bergognoni, and A. M. Casali. Multiresolution registration for volume reconstruction in microscopical applications. In M. H. Loew, editor, *Medical imaging: image processing*, volume 1898, pages 55–60, WA, 1993. SPIE Press.
- [64] T. Ault and M. W. Siegel. Frameless patient registration using ultrasonic imaging. *Medical robotics and computer assisted surgery*, pages 74–81, 1994.
- [65] C. S. Chua and R. Jarvis. 3d free-form surface registration and object recognition. *International journal of computer vision*, 17:77–99, 1996.
- [66] J. Thirion. New feature points based on geometric invariants for 3d image registration. *International journal of computer vision*, 18(2):121–137, 1996.
- [67] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992.
- [68] K. Rohr. On 3d differential operators for detecting point landmarks. *Image and Vision Computing*, 15(3):219–233, 1997.
- [69] W. Beil, K. Rohr, and H. S. Stiehl. Investigation of approaches for the localization of anatomical landmarks in 3d medical images. In *Computer Assisted Radiology*, pages 265–270. Elsevier Science, 1997.
- [70] J.-P. Thirion. Extremal points: Definition and application to 3d image registration. In *Proceedings of CVPR*, pages 587–592. IEEE Computer Society Press, 1994.
- [71] R. Pielot, M. Scholz, K. Obermayer, E. D. Gundelfinger, and A. Hess. 3d edge detection to define landmarks for point-based warping in brain imaging. In *International Conference on Image Processing*, volume 2, pages 343–346. IEEE Computer Society Press, 2001.
- [72] A. Rosenfeld and E. Johnson. Angle detection on digital curves. *IEEE Transactions on Computing*, C-22:875–878, 1973.
- [73] H. Freeman and L. S. Davis. A corner-finding algorithm for chain-coded curves. *IEEE Transactions on Computing*, C-26:297–303, 1977.
- [74] P. Zhu and P. M. Chirlian. On critical point detection of digital shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(8):737–748, 1995.
- [75] A. Hill, C. J. Taylor, and A. D. Brett. A framework for automatic landmark identification using a new method of nonrigid correspondence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(3):241–251, 2000.

- [76] W. E. L. Grimson. *Object Recognition by Computer*. MIT Press, Cambridge, MA, 1990.
- [77] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. M.I.T. Press, Cambridge, MA, 1967.
- [78] S. M. Pizer, C. A. Burbeck, J. M. Coggins, D. S. Fritsch, and B. S. Morse. Object shape before boundary shape: scale-space medial axes. *Journal of Mathematical Imaging and Vision*, 4:303–313, 1994.
- [79] S. M. Pizer, D. H. Eberly, B. S. Morse, and D. S. Fritsch. Zoom invariant vision of figural shape: the mathematics of cores. *Computer Vision and Image Understanding*, 69(1):55–71, 1998.
- [80] G. Stetten. *Automated Identification and Measurement of Cardiac Anatomy Via Analysis of Medial Primitives*. Doctoral dissertation, University of North Carolina, Chapel Hill, 1999.
- [81] G. Stetten and S. M. Pizer. Medial-node models to identify and measure objects in real-time 3-d echocardiography. *IEEE Transactions on Medical Imaging*, 18(10):1025–1034, 1999.
- [82] R. Tamburo, A. Cois, D. Shelton, and G. Stetten. Medial node correspondances towards automated registration. In Gee Maintz and Vannier, editor, *WBIR*, Philadelphia, PA, 2003. Springer/LNCS.
- [83] G. Stetten and S. Pizer. Extracting shape properties via populations of medial primitives. *University of North Carolina, Department of Computer Science TR98-008*, 1998.
- [84] S. M. Pizer, D. S. Fritsch, P. A. Yushkevich, V. E. Johnson, and E. L. Chaney. Segmentation, registration, and measurement of shape variation via image object shape. *IEEE Transactions on Medical Imaging*, 18(10):851–865, 1999.
- [85] P. T. Fletcher, S. M. Pizer, A. G. Gash, and S. Joshi. Deformable m-rep segmentation of object complexes. In *IEEE International Symposium on Biomedical Imaging*, pages 26–29, Washington D.C., 2002.
- [86] G. Stetten and S. Pizer. Medial-guided fuzzy segmentation. In SL Delp, AM Digioia, and B. Jaramaz, editors, *Medical image Computing and Computer-Assisted Intervention (MICCAI)*, volume 1935 of *Lecture Notes in Computer Science*, pages 226–235, Pittsburgh, 2000. Springer.
- [87] B. S. Morse, S. M. Pizer, and A. Liu. Multiscale medial analysis of medical images. information processing in medical imaging. *Image and Vision Computing*, 12:327–338, 1993.

- [88] C. A. Burbeck and S. M. Pizer. Object representation by cores: Identifying and representing primitive spatial regions. *Vision Research*, 35(13):1917–1930, 1995.
- [89] H. Tek, P. Stoll, and B. Kimia. Shocks from images: propagation of orientation elements. In IEEE, editor, *Conference on Computer Vision and Pattern Recognition*, Puerto Rico, 1997.
- [90] K. Siddiqi and B. Kimia. Toward a shock grammar for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1996.
- [91] B. Kimia, A. Tannenbaum, and S. Zucker. Shapes, shocks, and deformations, i: The components of shape and the reaction-diffusion space. *International journal of Computer Vision*, 15:189–224, 1995.
- [92] P. Giblin and B. Kimia. On the local form of symmetry sets, and medial axes, and shocks in 3d. In *Conference on Computer Vision and Pattern Recognition*, 2000.
- [93] L. Calabi and W. Hartnett. Shape recognition, prairie fires, convex deficiencies and skeletons. *Amer. Math. Monthly*, 75(335-342), 1968.
- [94] F. Meyer. Skeleton and perceptual graphs. *Signal Processing*, 16:335–363, 1989.
- [95] I. Pitas and A. Vanetsanopoulos. Morphological shape representation. *Pattern Recognition*, 25(6):555–565, 1992.
- [96] S. Pizer, P. Fletcher, A. Thall, M. Styner, G. Gerig, and S. Joshi. Object models in multiscale intrinsic coordinates via m-reps. *Image and Vision Computing*, 21:5–15, 2002.
- [97] F. M. De Monasterio. Center and surround mechanisms of opponent-color x and y ganglion cells of retina of macaques. *Journal of Neurophysiology*, 41:1418–1434, 1978.
- [98] R.A. Young. The gaussian derivative model for machine vision: visual cortex simulation. *Technical Report GMR-5323*, General Motors Research Laboratories, 1986.
- [99] M. H. Zweig and G. Campbell. Receiver-operator characteristic (roc) plots: a fundamental evaluation tool in clinical medicine. *Clinical Chemistry*, 39:561–577, 1993.
- [100] C. E. Metz. Basic principles of roc analysis. *Seminars in Nuclear Medicine*, 8:283–298, 1978.
- [101] P. F. Griner, R. J. Mayewski, A. I. Mushlin, and P. Greenland. Selection and interpretation of diagnostic tests and procedures. *Annals of Internal Medicine*, 94:555–600, 1981.
- [102] W. P. Tanner and J. A. Swets. A decision-making theory of visual detection. psychological review. *Psychological Review*, 61:401–409, 1954.

- [103] L. B. Lusted. *Introduction to Medical Decision Making*. Springfield, Springfield, Illinois, 1968.
- [104] L. B. Lusted. Signal detectability and medical decision-making. *Science*, 171:1217–1219, 1971.
- [105] R. K.-S. Kwan, A. C. Evans, and G. B. Pike. An extensible mri simulator for post-processing evaluation. *Lecture Notes in Computer Science*, 1131:135–140, 1996.
- [106] C. A. Cocosco, V. Kollokian, R. K.-S. Kwan, and A. C. Evans. Brainweb: Online interface to a 3d mri simulated brain database. In *Proceedings of 3rd International Conference on Functional Mapping of the Human Brain*, volume 5, 1997.
- [107] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans. Design and construction of a realistic digital brain phantom. *IEEE Transactions on Medical Imaging*, 17(3):463–468, 1998.
- [108] R. K.-S. Kwan, A. C. Evans, and G. B. Pike. Mri simulation-based evaluation of image-processing and classification methods. *Transactions on Medical Imaging*, 18(11):1085–1097, 1999.
- [109] M. A. Fischler and R. C. Bolles. Random sampling consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [110] T K Biswas and A K Gupta. Retrieval of true color of the internal organ of ct images and attempt to tissue characterization by refractive index: Initial experience. *Ind J Radiol Imag*, 12(2):169–178, 2002.
- [111] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1:127–136, 1971.
- [112] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart, and Winston, 1976.
- [113] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, 1982.
- [114] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, Cambridge, second edition, 1993.
- [115] L. Ibanez, W. Schroeder, L. Nyg, and J. Cates. The itk software guide, 2003.
- [116] The Insight Software Consortium. *Insight Into Images: Principles and Practice for Segmentation, Registration, and Image Analysis*. A K Peters, 2004.
- [117] The Fast Light Toolkit. <http://www.fltk.org/>, 1998.

- [118] W. Schroeder, K. Martin, and B. Lorenson. *Visualization Toolkit: An Object-Oriented Approach To 3D Graphics*. Prentice Hall, Englewood Cliffs, NJ, 3 edition, 2003.
- [119] C. A. Cois. *Shells and Spheres: A Novel Framework for Variable Scale Statistical Image Analysis*. Master's thesis, University of Pittsburgh, 2006 (in preparation).