Fully Automated Medical Image Analysis Facilitating Subsequent User Analysis

by

Shawn Andrews
B.Sc., University of Waterloo, 2008

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the

School of Computing Science
Faculty of Applied Sciences

© Shawn Andrews 2014
SIMON FRASER UNIVERSITY
Summer 2014

All rights reserved.
However, in accordance with the Copyright Act of Canada, this work may be reproduced without authorization under the conditions for "Fair Dealing." Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.
APPROVAL

Name: Shawn Andrews
Degree: Doctor of Philosophy
Title of Thesis: Fully Automated Medical Image Analysis Facilitating Subsequent User Analysis

Examinig Committee:
Dr. Ze-Nian Li
Professor, Computing Science, Simon Fraser University
Chair

Dr. Ghassan Hamarneh (Senior Supervisor)
Associate Professor, Computing Science, Simon Fraser University

Dr. Greg Mori (Supervisor)
Associate Professor, Computing Science, Simon Fraser University

Dr. Faisal Beg (Internal Examiner)
Associate Professor, Engineering Science, Simon Fraser University

Dr. William Wells (External Examiner)
Professor, Department of Radiology, Harvard Medical School

Date Approved: July 3rd, 2014
Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the non-exclusive, royalty-free right to include a digital copy of this thesis, project or extended essay[s] and associated supplemental files ("Work") (title[s] below) in Summit, the Institutional Research Repository at SFU. SFU may also make copies of the Work for purposes of a scholarly or research nature; for users of the SFU Library; or in response to a request from another library, or educational institution, on SFU's own behalf or for one of its users. Distribution may be in any form.

The author has further agreed that SFU may keep more than one copy of the Work for purposes of back-up and security; and that SFU may, without changing the content, translate, if technically possible, the Work to any medium or format for the purpose of preserving the Work and facilitating the exercise of SFU’s rights under this licence.

It is understood that copying, publication, or public performance of the Work for commercial purposes shall not be allowed without the author’s written permission.

While granting the above uses to SFU, the author retains copyright ownership and moral rights in the Work, and may deal with the copyright in the Work in any way consistent with the terms of this licence, including the right to change the Work for subsequent purposes, including editing and publishing the Work in whole or in part, and licensing the content to other parties as the author may desire.

The author represents and warrants that he/she has the right to grant the rights contained in this licence and that the Work does not, to the best of the author's knowledge, infringe upon anyone's copyright. The author has obtained written copyright permission, where required, for the use of any third-party copyrighted material contained in the Work. The author represents and warrants that the Work is his/her own original work and that he/she has not previously assigned or relinquished the rights conferred in this licence.

Simon Fraser University Library
Burnaby, British Columbia, Canada

revised Fall 2013
Abstract

In a clinical setting, accuracy is paramount for medical image analysis tasks such as segmentation and registration. Since it is often required that results be manually verified by a human expert, computational techniques designed to aid clinicians in these image analysis tasks are usually interactive, requiring user input. However, these techniques cannot take advantage of the time between when an image is acquired and when a clinician is available to provide input.

In this thesis, we will present novel techniques for automatically processing medical images, with the goal of facilitating later analysis by a human expert. These techniques fall into two classes. The first class involves leveraging prior anatomical information to automatically generate results that are robust and independent of initialization. The second class involves precomputing data that is used to greatly increase the speed and responsiveness of subsequent interactive techniques, saving clinicians valuable time. Each of the techniques presented focus on encoding meaningful uncertainty information, which can guide human experts to potential errors or pathologies.

Keywords: uncertainty; segmentation; registration; statistical shape models; precomputation; user interaction
Contents

Approval ii
Partial Copyright License iii
Abstract iv
Contents v
List of Tables viii
List of Figures xi
List of Terms xxi

1 Introduction 1
1.1 Thesis Context 1
1.2 Thesis Contributions 3
1.3 Auto-Bibliography 7
1.4 Other Contributions 8

2 Review 10
2.1 Uncertainty in Image Analysis 10
    2.1.1 Uses of Uncertainty 11
    2.1.2 Inherent Uncertainty 13
    2.1.3 Automated Image Analysis Errors 14
2.2 Fully Automated and Interactive Segmentation 19
    2.2.1 Point Distribution Models 23
5 Precomputation

5.1 Fast Random Walker Image Segmentation ........................................ 101
  5.1.1 Precomputation for Random Walker Image Segmentation ................. 103
  5.1.2 Adaptive Precomputation for Segmentation ................................... 112
5.2 Fast Random Walker Image Registration .......................................... 121
  5.2.1 Precomputation for Registration .................................................. 123
  5.2.2 Adaptive Precomputation for Registration ...................................... 126
  5.2.3 Ensuring Anatomical Feasibility .................................................. 132
5.3 Chapter Summary .............................................................................. 141

6 Conclusion and Future Work .............................................................. 144

6.1 Thesis Summary ............................................................................... 144
6.2 Future Work .................................................................................... 145

Bibliography .......................................................................................... 147
List of Tables

2.1 Sample works utilizing uncertainty for various purposes. .................. 13
2.2 Fuzzy versions of popular segmentation representations. ................. 15
2.3 Categorization of segmentation methods based on their internal and external energy terms. While energy terms based on models of the anatomy (e.g. boundary smoothness) or the image acquisition process (e.g. noise distribution) may sometimes provide better results, statistically motivated energy terms can give a more accurate measure of uncertainty when treated as a distribution over segmentations (see (2.5)). It is often difficult to model distributions over image data or shape representations directly, therefore sometimes distributions are modeled over certain features. We also categorize methods based on the complexity of the distribution they use to model shape variability; while energy terms derived from Gaussians are simple and relatively easy to optimize, more complex distributions may model the variability in shapes more accurately. ................................................................. 17
2.4 Categorization of segmentation methods using statistical shape priors based on the optimizability of their energy functions. The ability to globally optimize an energy function implies that the space of possible segmentations can be efficiently searched in some way, and thus global uncertainty information could be extracted from that energy function (e.g. [30]). When only local minima are guaranteed, only local uncertainty information can be captured. We allow pose to be known a priori as globally optimizing shape and pose parameters simultaneously is a difficult problem and is largely unsolved. The methods are also categorized by segmentation representation and shape distribution, as these determine the domain of the energy function, and even convex energy functions can be difficult to optimize over non-convex domains. 18

2.5 A comparison of certain properties of popular algorithms. . . . . . . . . . . . 22

3.1 The mean and standard deviation of the DSC values calculated between the 20 testing segmentations and their corresponding projections onto the ALR- and ILR-PCA shape spaces, built from the 20 training images. Also shown are the p-values generated from a paired t-test to determine if the differences in DSC values are statistically significant. Bolded p-values pass the Holm-Bonferroni test (used to correct for multiple comparisons) at a significance level of 0.05. We note that ILR-PCA performs better when modeling the labels that do not share a boundary with the background ($K \in \{1, \ldots, 14\}$, see Fig. 3.10). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

5.1 A table of the results from Sec. 5.8. Adaptive fast RWIS chooses the number of eigenvectors to use online based on the user seeds. The results reported for fast RWIS are for a fixed number of eigenvectors $m$, chosen separately for each image to give the run-time and accuracy closest to its adaptive counterpart. Accuracy is reported using DSC. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 120

5.2 A table of the results from Fig. 5.14. Adaptive fast RWIR chooses the number of eigenvectors to use online based on the prior probabilities. The results reported for fast RWIR are for a fixed number of eigenvectors $m$, chosen separately for each image to give the run-time and accuracy closest to its adaptive counterpart. Accuracy is reported using MO. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 134
5.3 The results of running RWIR with different regularization matrices. RWIR-TP achieves less error than the other registrations while also ensuring topology preservation, while only requiring marginally longer computation time. 138

5.4 MOs and % of pixels with negative Jacobian from registering pairs of images, averaged across all regions and tests. RWIR-TP and RWIR-U achieve similar MOs, but RWIR-TP provides topology preserving transformations. 142
List of Figures

1.1 An overview of image analysis paradigms. The contributions of this thesis focus on (d) and (e). ......................................................... 4

2.1 A schematic of the medical image acquisition and segmentation process, highlighting in red where the uncertainty described in this section arises. ........... 12

2.2 A heuristic example of how uncertainty information is encoded in energy functions. While both $E_1$ and $E_2$ have the same minimizing segmentation, $E_1$ quickly increases for segmentations near the minimum, while in $E_2$ segmentations near the minimum all have a similar energy value. This may imply more uncertainty in the optimal segmentation when using $E_2$, as small perturbations in the image (e.g. additional noise) might noticeably change the minimum of the energy. ......................................................... 16

3.1 Performing PCA directly in the simplex results in a PCA mode (green line) that leaves the constrained space. Vertices correspond to “certain” probabilistic labels. See Fig. 3.4 and Fig. 3.7 for comparison, particularly Fig. 3.4 for a description of how the points were generated. Uncertainty is lost when points are projected back to the simplex. ................................. 38
3.2 Effect of perturbation and power transformation on compositional vectors.
(a) A set of compositional vectors (blue curve, arrow) are perturbed by various other points in the simplex (red to green dots). The perturbations result in new sets of compositional vectors (red to green curves). (b) Power transformation of the blue curve from (a) with scalars ranging from 0 (magenta curves) to 2 (cyan curves). Note power transformation by 1 leaves the curve unchanged (blue curve, arrow), power transformation by 0 reduces the curve to a point in the center of the simplex, and power transformation by large values shifts the curve to the edge of the simplex.

3.3 Orthogonal and parallel compositional lines. Blue compositional lines $\hat{q}_1(s) = q_1 \oplus (s \odot p_1)$ and $\hat{q}_2(s) = q_2 \oplus (s \odot p_1)$ are parallel. These lines represent how different probabilistic labels $q_1 = [0.38, 0.24, 0.38]^T$ and $q_2 = [0.2, 0.6, 0.2]^T$ are updated when an observation gives the likelihood $p_1 = [0.2, 0.3, 0.5]^T$. Note that although $\hat{q}_1(s)$ and $\hat{q}_2(s)$ converge, they never intersect, getting arbitrarily close as $s \to \infty$. Green compositional lines $\hat{q}_3(s) = q_3 \oplus (s \odot p_3)$ and $\hat{q}_4(s) = q_3 \oplus (s \odot p_4)$ are orthogonal, since $\langle p_3, p_4 \rangle_a = 0$. These lines represent how the probabilistic label $q_3$ is updated when an observation gives either the likelihood $p_3 \approx [0.48, 0.32, 0.20]^T$ or the likelihood $p_4 \approx [0.23, 0.53, 0.24]^T$. Since $p_3$ and $p_4$ are an orthogonal basis for $S3$, any other likelihood $r$ can be expressed as $r = (c_3 \odot p_3) \oplus (c_4 \odot p_4)$, where $c_3 = \langle p_3, r \rangle_a / \|p_3\|_a$ and $c_4 = \langle p_4, r \rangle_a / \|p_4\|_a$. Thus, any new observations are equivalent to making the observations corresponding to the likelihoods $p_3$ and $p_4$, $c_3$ and $c_4$ times respectively.

3.4 Performing PCA in LogOdds space results in a PCA mode (red line) that is skewed towards the last label. The points in (a) are found by: sampling 2000 points $[p_x, p_y]$ from an isotropic 2D Gaussian, skewing the $y$ component using $p_y < -0.7p_y(|p_y|^{0.8})$, applying a random rotation (fixed for all points), and mapping to the simplex using the inverse LogOdds transformation. The same point set is used in Fig. 3.1 and Fig. 3.7.
3.5 A demonstration of how the LogOdds and ILR transformations map three lines of equal length in $\mathbb{P}_3$ to $\mathbb{R}^2$. Under the ILR transformation all three lines are still of equal length, but the blue line is stretched under the LogOdds transformation. However, the LogOdds transformation has the useful property that the first two components of the probabilistic label map to the standard Euclidean axes.

3.6 The ILR transform and its inverse. (a) Orthogonal grid of lines, unit distance apart, in $\mathbb{R}^2$. (b) Correspondingly colored orthogonal compositional lines in $\mathbb{P}_3$ unit distance apart with respect to the Aitchison distance (see (3.18)).

3.7 Performing PCA in ILR space results in a PCA mode (blue line) that captures the variability in the data one would intuitively expect. The same point set is used in Fig. 3.1 and Fig. 3.4.

3.8 Quiver plots showing (a) $\nabla_1 \mathcal{H}$, (b) $\nabla_2 \mathcal{H}$, and (c) $\nabla_3 \mathcal{H}$ on $\mathbb{P}_3$. In (a), $\nabla_1 \mathcal{H}$ at each point is split into its component contained in the simplex (blue) and its component perpendicular to the simplex (red), emphasizing how $\nabla_1 \mathcal{H}$ points out of the simplex. Note how $\nabla_2 \mathcal{H}$ in (b) is elongated along the third (vertical) component. $\nabla_3 \mathcal{H}$ is both contained in the simplex and symmetric across the components.

3.9 A synthetic experiment detailing the differences between PCA in LogOdds and ILR space. (a) demonstrates how vectors in $\mathbb{P}_3$ are stretched along the first label. (b) plots the mean error of the changes of the label probabilities (see (3.57)) when taking a small step along the eigenmode of greatest variance, $\mathbf{q}_1$. (c) plots the CoV for $\{\lambda_2, \ldots, \lambda_{K-1}\}$. See text for further details.

3.10 An illustration of the PCA modes of greatest variation found using ALR transformed data (b) and ILR transformed data (c). (a) shows the segmentation of an axial slice of a thigh MRI. White corresponds to locations where the segmentation changes significantly when moving along the PCA mode (i.e. sum of the change in all probabilities when taking a small step away from the mean), and black corresponds to locations where the segmentation does not change. We see that PCA in ILR space captures variance in all of the boundaries, while PCA in ALR space focuses on variance in the background.
4.1 An example axial slice of a thigh MRI and the corresponding labeled ground truth. ................................................................. 63

4.2 An example of the columns of $A$ for the LogOdds, ILR, and GLR transformations for $K = 3$ labels. For the GLR transformation, the columns are chosen to satisfy a predefined pairwise distance function $D$. .......................... 66

4.3 An illustration of how the GLR transformation can be used to ensure small or thin structures are properly segmented. An image region may be segmented correctly into 3 regions (a) or incorrectly into 2 regions (b). This example assumes a regularization term penalizing the total change in the segmentation. Using no prior adjacency information such as the ILR transformation may result in the middle region being skipped, as the segmentation incurs too large of a regularization penalty (c). If it is known from the training data that regions 1 and 3 do not share boundaries with each other, but rather with the thin region 2, the GLR transformation can be constructed so that transitioning from region 1 to region 3 through region 2 does not incur any additional cost compared to skipping region 2 (d). Such situations occur in thigh muscle segmentation (e). ......................................................... 70

4.4 An example illustrating the GMM probabilities and presegmentation. In (b), we see the thresholded probabilities for GMM on “light”, “gray”, and “dark” labels. In (c), we see the results of the thresholded presegmentation of the image. ................................................................. 73

4.5 Regions in the thigh, and the average intensity profiles of the thighs. We see the background and cortical bone are dark, the muscle is medium, and the fat and bone marrow are light. ................................................................. 74

4.6 An example illustrating the anatomical alignment. In (a), we see the pre-segmentation into muscle, bone, and fat. In (b), we see the overlay of the presegmented muscle labels from two images (red and green, overlap in yellow), and in (e) we see the same muscle labels overlayed after being aligned. ................................................................. 76

4.7 The mean shape found using PCA. The 11 muscle labels and bone label are displayed in groups of 3 to aid in visualizing the probabilistic labels. ....... 78
4.8 A graphical illustration of the dual formulation of $E_{\text{bnd}}$ in (4.53). The example is shown in 2D instead of 3D for clarity. (a) shows the $p$ corresponding to a $\nabla \phi^i(x)$ parallel to $d_1(x)$. (b) shows the $p$ corresponding to a $\nabla \phi^i(x)$ perpendicular to $d_1(x)$. (c) shows the $p$ for an arbitrary $\nabla \phi^i(x)$, with the strength of the penalty being proportional to the length of the projection of $p$ onto $\nabla \phi^i(x)$.

4.9 An example showing the false boundaries created by intramuscular fat, as found by a Canny edge detector (b), with segmentation (i.e. true boundaries) shown in red for comparison (c).

4.10 An example showing true boundaries (green boxes) and false boundaries (red) (d), demonstrating the difficulty in distinguishing them. The appearances of the true and false boundaries overlap heavily, thus a powerful multi-modal classifier is necessary for accurate boundary detection.

4.11 An illustration of how the filters are shifted at each voxel to make the feature vector used when training the RF. The filters corresponding to the red direction should give the largest response. The feature vectors at each location are constructed by ordering the filter responses so that the direction with the largest expected response comes first. When applying the RF, each of the 18 possible orderings of the filters are tested.

4.12 An example of the boundary likelihoods found using our RF classifier. In (b), whiter shading corresponds to higher likelihoods. (c) shows the manual boundaries.

4.13 Examples of our segmentation algorithm applied to axial brain slice images. We see the shape prior greatly improves the segmentation. In most color figures of brain images seen here, red corresponds to CSF, green to gray matter, blue to white matter, and black to background. In gray scale images, lighter shades correspond to higher intensities/probabilities/entropies.
4.14 The segmentation of both an image and the same image partially obstructed, demonstrating the usefulness of our shape prior. The DSC of the segmentation without and with the obstruction is 0.8114 and 0.7995, respectively. Thus, despite the obstruction, the DSC only drops 1.5%. In 4.14d and 4.14h we note the increase in entropy under the obstruction (red arrow), corresponding to uncertainty in our probabilistic segmentation. Note that while such a glaring occlusion does not typically appear in MRI images, small artifacts and noise do. This example is designed to make the effect of the shape prior obvious.

4.15 An obstructed image is segmented into 2, 3, and 4 regions, showing how multi-region interactions improve the segmentation of individual regions. The bottom row shows the ground truth for the CSF and how the CSF’s segmentation improves as more regions are included.

4.16 Analysis of segmentation uncertainty. A close up of the CSF region in the center of an image for the purposes of understanding the sources of the uncertainty in the segmentation, shown by its entropy in the bottom right figure. We can infer the origin the of entropy of the segmentation (4.16h) from the intensity priors (4.16f) and the projected ground truth (4.16e). Areas of high intensity gradient (4.16c) create uncertainty in the intensity prior, and thus in the segmentation. Areas where the ground truth (4.16d) and the projected ground truth differ indicate features not representable by the shape prior, and again this affects the segmentation’s certainty.

4.17 The results of segmenting a brain slice using a non-convex the EM method [241] with different initializations. The initialization consists of an initial probabilistic segmentation in the shape space and an initial intensity inhomogeneity parameter for each pixel. When given the optimal initialization (bottom row), a good segmentation is found, having a DSC of 0.8337. However, when given an initialization of zeros (top row), the results are much worse, with the resulting segmentation having a DSC of 0.6766. The minimization gets stuck in a local minimum, as can often happen with non-convex energies. See the second row of Fig. 4.13 for our method’s performance on this image (requiring no initialization).
4.18 A summary of our method and the various techniques we compare against (see Algorithm 2). Colored blocks correspond to steps from Algorithm 2, and gray blocks correspond to one of our contributions being replaced by another technique.

4.19 A comparison of manual and automated segmentation results.

4.20 A comparison of GLR-Full to each of the other 5 methods in Fig. 4.18, summarized across each label and each image.

4.21 A comparison of the various algorithms, where positive values indicate GLR-Full performs better. See text for discussion. The results are summarized in Fig. 4.20.

4.22 An analysis of uncertainty information and error. (a) shows the voxels of top 5% most uncertainty (green), the erroneous voxels (red), and their intersection (yellow). (b) and (c) show the “error precision” (yellow/(yellow+green)) and “error recall” (yellow/(yellow+red)) for percentiles of uncertainty across all images. A general strategy for identifying errors in a crisp segmentation might be to simply take boundary voxels as being likely erroneous, so we present this as a baseline in (b) and (c). We see that the top 5% of voxels have about 20% higher precision but the same recall as the boundary voxels, indicating uncertainty is useful for error detection.

5.1 Comparison of results with and without precomputation for segmentation using priors on an image of size $N \approx 72,000$ pixels. For $m$, the number of eigenvectors used, we report the DSC between results from basic RWIS with priors and fastRWIS, $t_{on}$, the online time taken, and $t_{off}$, the offline time taken. We note that we are only concerned with $t_{on}$ and with $m = 80$ our method achieves excellent results in less than a fifth of the time taken when not using precomputation. Red and green correspond to different region boundaries.

5.2 Comparison of the runtimes of the basic RWIS (blue), fastRWIS run using Algorithm 3 (green), and fastRWIS run using the additional precomputation in Algorithm 4 (red) for different resolutions of the image in Fig. 5.1. Note the standard deviation for the online runtimes are all under 0.05 seconds.
5.3 Effect of $m$ and noise on segmentation accuracy. Compares the DSC between the segmentations found using basic RWIS and fastRWIS. Results are shown for two levels of noise and for multiple numbers of eigenvectors. 111

5.4 Effect of $m$ and noise on segmentation accuracy. Shows the DSC between the segmentations at varying levels of noise with $m = 200$ eigenvectors and 20 trials for each level of noise. We see that large enough $m$ lets us account for reasonable amounts of noise. 111

5.5 The bones of a knee segmented with $S = 100$ seeds in one of the bones. (a) was found using basic RWIS in 40.5 seconds and (b) was found using fastRWIS with $m = 350$ in 1.56 seconds. DSC between the two is 0.975. 112

5.6 A slice of a thigh MRI and several corresponding eigenvectors. The eigenvectors correspond to relaxed normalized cuts of the image, and thresholding them would result in a segmentation of certain structures, albeit perhaps not a useful one, as no user guidance has yet been given. 113

5.7 An example of the segmentations achieved by basic RWIS and fastRWIS with eigenvectors adapted to the seeds. Regions are outlined in color. fastRWIS takes only about 20% of time as basic RWIS (Fig. 5.8). 118

5.8 A comparison of run-time (as a fraction of basic RWIS) vs. accuracy for basic RWIS, fastRWIS with fixed numbers of eigenvectors, and fastRWIS with eigenvectors adapted automatically to the seeds. Our adaptive approach achieves excellent accuracy and run-time while automatically choosing the number of eigenvectors to use. See Table 5.1 for more details. 119

5.9 A comparison of fastRWIS accuracy when $\beta$ is known offline versus when $\beta$ is changed online and $\Lambda$ is updated using our technique. The eigenvectors from the Laplacian with $\beta' = 50$ are used as the base eigenvectors when $\beta$ is not known. The difference in DSC between the results of the two methods is reported. As expected, the DSC is higher when the Laplacian is known offline, though only slightly higher for a large range of values, particularly for the blood cell image. This indicates our technique allows $\beta$ to be adjusted online by at least a factor of 2 while sacrificing only minimal accuracy. 120
5.10 The average distance between the displacements generated by RWIR and fastRWIR for different values of $k$ and the average times taken for both RWIR and fastRWIR. For reference, the average distance between the RWIR and the ground truth displacements was $2.24 \pm 0.05$ mm, while for $k = 1000$ the average distance between the RWIR and the fastRWIR displacements is much smaller at $0.07 \pm 0.06$ mm, indicating that the approximation used in fastRWIR will have minimal effect on the registration error. RWIR runs in $2390 \pm 1700$ seconds (shown in red as a baseline), whereas fastRWIR with $k = 1000$ runs in $795 \pm 253$ seconds.

5.11 The DSCs of the 56 segmented brain regions change after registration, averaged across all pairs of images. Basic RWIR took $2716 \pm 1783$ secs. to perform the registrations, while fastRWIR took only $830 \pm 478$ secs. We note that both RWIR and fastRWIR (with $m = 1000$) typically improve the correspondence between anatomical regions, and perform very similarly.

5.12 The increase in DSC of each of the 56 segmented brain regions when registering with RWIR instead of fastRWIR with $m = 1000$. Some outliers above and below the current bounds have been excluded for readability.

5.13 An example of the registrations achieved by basic RWIR and fastRWIR with eigenvectors adapted to the priors. fastRWIR takes only about 30% of time as basic RWIR (Fig. 5.14).

5.14 A comparison of run-time (as a fraction of basic RWIR) vs. accuracy for basic RWIR, fastRWIR with fixed numbers of eigenvectors, and fastRWIR with eigenvectors automatically adapted to the prior probabilities. Our adaptive approach achieves excellent accuracy and run-times while automatically choosing the number of eigenvectors to use. See Table 5.2 for more details.

5.15 An example of aggregated graphs used to improve RWIR efficiency. Nearby nodes with similar prior probabilities are aggregated into super-vertices. The aggregated graphs have on average 18% and 15% as many vertices as the original graph for the thigh and brain images, respectively.
5.16 A comparison of RWIR techniques used on graphs with aggregated super-vertices. Accuracy and run-time are given in proportion to RWIR run on the full graph (Fig. 5.14 for absolute values). When run on the aggregated graph, RWIR achieves a significant speed-up with a minimal loss in accuracy, with further speed-up achieved by fastRWIR. The run times and MO values are given as a fraction of those achieved by the basic RWIS algorithm (Fig. 5.14 for absolute values). We note that using the eigenvectors from the aggregated graph (“□”) is not actually possible, since the aggregated graph is not known off-line, and is only included as a baseline.

5.17 A 1D example with circles representing pixels, unit distance apart, and arrows representing their displacement vectors. (i) The red pixels have crossing displacements and thus a negative Jacobian. (ii) After increasing regularization at the red pixels, their displacements do not cross, but the orange and purple pairs of pixels now have crossing displacements. (iii) Increasing the regularization of pixels based on their proximity to the location of the negative Jacobian in (i) results in a topology preserving transformation.

5.18 A histogram of the Jacobian values for the ground truth, RWIR-U, and RWIR-TP transformations, averaged over each of the tests, and with a log-scale on the y-axis for readability. While the ground truth and RWIR-TP Jacobians are mostly contained in the range [0.5, 1.5], the RWIR-U have Jacobian values far from 1 and even less than 0.

5.19 Mean TRE for different uncertainty percentiles. The mean TRE of the top i% most uncertain pixels was calculated, for i = {1, ..., 99}. The mean TRE consistently increases across percentiles, indicating uncertainty could be used to identify errors in a registration. Particularly, we see a sharp increase in TRE for the top 10% most uncertain pixels.

5.20 An example on real data demonstrating how our technique increases regularization locally where necessary without over-regularizing other areas and affecting accuracy.
# List of Terms

<table>
<thead>
<tr>
<th><strong>Imaging Modalities</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>CT</td>
</tr>
<tr>
<td>MRI</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>General Imaging</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
</tr>
<tr>
<td>PVE</td>
</tr>
<tr>
<td>TP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Medical Imaging Tasks</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>MIA</td>
</tr>
<tr>
<td>MIS</td>
</tr>
<tr>
<td>MIR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Algorithms</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>RW</td>
</tr>
<tr>
<td>RWIS</td>
</tr>
<tr>
<td>RWIR</td>
</tr>
<tr>
<td>RWIR-C</td>
</tr>
<tr>
<td>RWIR-U</td>
</tr>
<tr>
<td>RWIR-TP</td>
</tr>
<tr>
<td>GC</td>
</tr>
<tr>
<td>ACWE</td>
</tr>
<tr>
<td>EM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Evaluation</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>MO</td>
</tr>
<tr>
<td>DSC</td>
</tr>
<tr>
<td>TRE</td>
</tr>
</tbody>
</table>
## Misc.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>Principal component analysis</td>
</tr>
<tr>
<td>LR</td>
<td>Log-ratio</td>
</tr>
<tr>
<td>ALR</td>
<td>Additive LR</td>
</tr>
<tr>
<td>CLR</td>
<td>Centered LR</td>
</tr>
<tr>
<td>ILR</td>
<td>Isometric LR</td>
</tr>
<tr>
<td>GLR</td>
<td>Generalized LR</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum <em>a posteriori</em> probability (estimation)</td>
</tr>
<tr>
<td>GMM</td>
<td>Gaussian Mixture Model</td>
</tr>
<tr>
<td>SDM</td>
<td>Signed Distance Map</td>
</tr>
<tr>
<td>MRF</td>
<td>Markov random field</td>
</tr>
<tr>
<td>CRF</td>
<td>Conditional random field</td>
</tr>
<tr>
<td>COPD</td>
<td>Chronic Obstructive Pulmonary Disease</td>
</tr>
<tr>
<td>RF</td>
<td>Random Forest</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Thesis Context

Over the past several decades, there have been great advances in medical imaging technology. These imaging techniques allow doctors to non-invasively visualize the internal structures of patients. The ability to gain detailed anatomical and functional information without the need for exploratory surgery is invaluable to many clinical tasks. Many techniques for acquiring medical images are in common use, based on a variety of physical principles: ultrasound (US), positron emission tomography (PET), magnetic resonance imaging (MRI), X-ray computed tomography (CT), single-photon emission computed tomography (SPECT), etc. Each of these modalities produce images with unique properties, useful for different tasks. Medical images usually divide space into a rectilinear grid, with grid points known as pixels in 2 dimensional (2D) image slices and voxels (or volumetric pixels)\(^1\) in 3 dimensional (3D) volumetric images. Images assign a value to each pixel; this value could be a scalar or a more complex structure such as a vector of scalars from different times \([90]\) or a matrix of diffusion values \([22]\).

A single medical image provides information that aids in the diagnosis of medical conditions, treatment planning, treatment execution, monitoring of recovery, and a growing number of other tasks. A cohort of multiple images provides statistical data that can give insight into correlations between anatomical variability and medical conditions. These tasks require the analysis and subsequent extraction of vital information from medical images,

\(^1\)For brevity, when not specifically referring to 3D images we use the term pixel.
which requires the expertise of a highly trained human expert.

A key component of medical image analysis (MIA) is medical image segmentation (MIS), the identification of regions in an image corresponding to structures of interest (e.g. human organs) [201]. The information provided by a segmentation allows medical personnel to correlate image features to the underlying anatomical structures. MIS exhibits some unique characteristics that differentiate it from general segmentation problems. Medical images often exhibit relatively low signal-to-noise ratios (SNRs) [20, 29, 236, 320], and can have low resolutions compared to the size of structures of interest (e.g. blood vessels or small tumors on an organ’s surface). Further, there are stringent requirements on segmentation accuracy in clinical settings due to the importance of its uses. These difficulties are somewhat mitigated by the fact that medical images usually come with contextual information, so the structures that are expected to be present in the image are known a priori. This contextual information allows human experts to consistently generate adequate manual segmentations of medical images, but only after they have spent years training to do so.

Another key task in MIA is medical image registration (MIR), the task of bringing different images of similar anatomy into alignment by applying a deformation to one of the images [316]. Differences between images are typically due to either acquisition factors (e.g. pose, the translation and rotation of the anatomy) or shape variability (e.g. anatomical differences between patients). MIR facilitates a number of tasks, such as identifying changes in anatomy between images taken at different times, fusing images from multiple modalities to create a more complete picture, or combining a cohort of images from different patients into a single atlas that captures statistics on shape variability throughout a population. MIR can be performed manually by identification of landmark points, spatial locations in two images with corresponding anatomy.

Both MIS and MIR are typically formulated as optimization problems, where the delineation (in MIS) or the deformation (in MIR) is found by minimizing an energy functional. The energy functional combines terms that capture how well the proposed solution matches the image information (i.e. data terms) with terms that regularize the solution to ensure robustness to noise and missing information (i.e. regularization terms).

In clinical settings, accurate analysis of medical images is paramount to ensure patients receive proper diagnoses and treatments. This analysis can be time consuming, especially for large volumetric images, and the time of expertly trained radiologists is extremely valuable. Thus, the benefit of developing computational techniques to aid in image analysis tasks
such as MIS and MIR cannot be overstated. However, achieving sufficiently accurate results automatically is still infeasible for many MIA applications, and results generated manually by a human expert are typically considered the gold standard or ground truth (GT) (e.g. the unknown segmentation that correctly assigns every pixel its proper label [151]). As such, it is a common practice to evaluate automated methods by comparison to manual results. To aid in manual verification, many MIA algorithms include auxiliary information, such as uncertainty, where an algorithm attempts to identify potential errors in its own results. We note that uncertainty does not necessarily correlate with error; just as algorithms have varying levels of accuracy, algorithms may do a better or worse job quantifying their own certainty.

Since final diagnoses must be performed by radiologists, computational techniques used for MIA often require user interaction, where a clinician provides input to the algorithm, then waits for results while the algorithm runs, either to provide additional input or to verify the results [216]. For large volumetric images, this wait time may be significant, and thus not an optimal use of an expert clinician’s time (Fig. 1.1a).

Given the high demand for radiologists, there is often a delay between the acquisition of a medical image and when it receives attention from a user. Since interactive techniques require user input (Fig. 1.1b, Fig. 1.1c), they must run online (while a user waits), and cannot take advantage of this offline time. Fully automated analysis techniques, however, may be able to extract valuable image information offline. We note, though, that clinical results require human accountability, and fully automated techniques often cannot reliably achieve sufficient accuracy, so the focus of automated techniques should be on aiding subsequent analysis.

1.2 Thesis Contributions

In this thesis, we will explore two paradigms for automated MIA. The first paradigm involves techniques that produce results automatically, but also focus on including auxiliary information, such as uncertainty information or suspected pathologies, that can aid in subsequent user analysis of the results [134,187,332,346,361] (Fig. 1.1d). Specifically, we focus on incorporating anatomical shape models and statistically meaningful uncertainty values into a fully automatic image segmentation framework [86,138].
Figure 1.1: An overview of image analysis paradigms. The contributions of this thesis focus on (d) and (e).
The second paradigm involves the precomputation of data offline that is designed specifically to increase the online speed and accuracy of an MIA algorithm [126] (Fig. 1.1e). Specifically, we focus on increasing the speed and accuracy of the random walker image segmentation [124] and registration [72] frameworks through precomputation.

In Chapter 2, we review uses of uncertainty in MIA. We go on to review existing work in MIS and MIR related to our contributions.

In Chapter 3, we discuss the benefits of a new segmentation representation over existing representations. The LogOdds transformation is one of a family of log-ratio (LR) transformations that bijectively map the simplex of length $K$ probability vectors (or probabilistic labels) to $\mathbb{R}^{K-1}$. The LogOdds transformation introduces a bias towards the last label in multi-label segmentation; the isometric log-ratio (ILR) transformation, proposed by Egozcue et al., is another LR transformation that is symmetric between its components [100]. The ILR transformation was recently introduced as a segmentation representation [66], and we explore the benefits of using this representation for building shape models, specifically in applications with prominent foreground-foreground boundaries (i.e. boundaries between anatomical structures of interest). We also provide connections between the Euclidean inner product between ILR vectors and Bayesian inference on their corresponding probabilistic labels. We have made MATLAB code related to using the ILR transformation for MIA available to the public at ilr.cs.sfu.ca.

In Chapter 4, we address the first paradigm (Fig. 1.1d), formulating a segmentation framework that satisfies four important properties for fully automated MIS:

- The algorithm should be robust to initialization. This can be achieved by ensuring convex terms in an energy minimization framework.

- The algorithm should employ an anatomical shape model [45, 86, 87, 91, 138, 177, 326]. In automatic segmentation, prior shape models can compensate for the lack of human expert knowledge and missing or corrupt data.

- The algorithm should be designed to segment multiple structures simultaneously [24, 52, 122]. Even in applications where only one anatomical structure is of interest, segmenting multiple structures simultaneously provides global context by allowing structures to be identified by their relative locations.
The algorithm should provide a probabilistic segmentation, which encodes spatially localized uncertainty information that can be used to identify errors in the segmentation and pathologies in the image [59, 273–275, 324, 346].

A key step in incorporating these properties is using an LR representation, as introduced in Chapter 3, for our segmentation and building statistical shape models by applying principal component analysis (PCA) to training data. We will perform segmentation by minimizing an energy functional consisting of a regional term, encoding information about local image features; a boundary term, encouraging the segmentation to change from one label to another in the vicinity of boundaries detected in the image; and a shape term, enforcing a PCA shape model. By constructing each of these terms to be convex, we can ensure a globally minimizing segmentation can be found independent of initialization. By constructing each of these terms using statistics from training data, we can encode meaningful uncertainty information into our probabilistic segmentation.

We demonstrate our segmentation framework on the challenging problem of thigh muscle segmentation. To this end, we introduce several additional contributions. We will explore the use of generalized log-ratio (GLR) transformations that increase the influence of labels corresponding to smaller structures, ensuring they are better modeled by statistical analysis techniques. We will also explore using GLR transformations to penalize transitions between structures that should not share a boundary, ensuring more anatomically feasible segmentations. This contribution is a probabilistic analogue of similar transition penalties on crisp (non-probabilistic) labels [175, 308, 309]. We will introduce a technique for identifying boundaries between thigh muscles using a rotationally invariant random forest classifier. Our classifier is able to detect intermuscular boundaries with widely varying appearances while ignoring false boundaries arising from intramuscular fat. Finally, we introduce an anatomically motivated alignment scheme based on presegmentations of the images, removing pose variability to facilitate the construction of shape models.

In Chapter 5, we present techniques conforming to the second paradigm (Fig. 1.1e) for automated MIA by developing precomputation techniques for the popular random walker (RW) algorithm. The RW algorithm has been applied to image segmentation (RWIS) [123, 124] and image registration (RWIR) [72]. For RWIS, Grady and Sinop [126] presented a technique for precomputing data offline, before user interaction, in order to speed the segmentation online, when a user is providing input. We extend this technique, making it compatible with dynamic prior probabilities that are computed online based on seeds. We
propose a technique to quickly choose the amount of precomputed data to use online in order to maximize effectiveness for a given set of user input. We also propose a technique for updating the precomputed data to compensate for altered algorithm parameters.

We extend this precomputation technique to RWIR, precomputing data for one image (e.g., an atlas) in order to increase the speed at which it can be registered to other images. We propose a technique to quickly choose the amount of precomputed data to use online in order to maximize effectiveness based on the other image(s) being registered. We also propose a technique for updating the precomputed data when the image graph topology is changed online, e.g., in a multi-resolution registration approach. We provide a novel precomputation technique for RWIR that can be used to efficiently determine spatially adaptive regularization strengths that ensure a topology preserving transformation without having to run the full RWIR algorithm. We have made MATLAB code related to these RW techniques available to the public at fastrw.cs.sfu.ca.

1.3 Auto-Bibliography

The chapters of this thesis are largely based on the following publications.

Chapter 3

Chapter 4


Chapter 5


1.4 Other Contributions

To maintain the focus of this thesis, other contributions that were made over the course of the corresponding graduate studies have been omitted. These include:

- A convex, multi-object segmentation framework with geometric constraints [214].

---

3©2011 IEEE. Reprinted, with permission, from Shawn Andrews, Chris McIntosh, and Ghassan Hamarneh, Convex multi-region probabilistic segmentation with shape prior in the isometric log-ratio transformation space, IEEE ICCV, 2011

• A technique for using uncertainty to improve registration accuracy [182].

• Analysis of white matter injury in the developing brain using diffusion tensor MRI data [9, 48].
Chapter 2

Review

In this chapter, we review works related to our contributions in this thesis. Uncertainty is a unifying theme throughout this thesis, with all of our algorithms generating probabilistic results. We begin by surveying works related to incorporating and utilizing uncertainty in image segmentation and registration in Sec. 2.1. In Sec. 2.2, we review popular segmentation algorithms, organized by the segmentation representation used in order to provide context for our novel segmentation representations. In Sec. 2.3, we review techniques for image registration, focusing on methods for enforcing regularity on deformations.

2.1 Uncertainty in Image Analysis

In MIS, manual expert segmentations are considered a realization of the ground truth (GT), so the goal of many automated MIS methods is to replicate manual results by providing a labeling of an image. However, no automated technique is currently reliable enough to be completely trusted in sensitive clinical settings, where human verification and accountability are considered paramount. A human expert must at the very least review and sign off on automated results, and significant user interaction is often still required to achieve usable results [243]. Thus the goal, at least short term, of automated MIS methods should be to increase the efficiency, accuracy, and reproducibility of human experts, not replace them. While there have been many works discussing how to validate the accuracy of a MIS method (i.e. the “correctness” of segmentations provided by a method) [37, 114, 158, 195, 240, 358–360], how to best quantify uncertainty is not a very well studied problem. In fact, many MIS methods do not explicitly attempt to quantify uncertainty, nor convey uncertainty
information to a user, though we will discuss how measures of uncertainty exist implicitly in many methods.

There have been, however, previous works addressing reasons why segmentation uncertainty should be explicitly encoded [34, 184, 293, 295, 307, 332, 349], and other works have presented methods for conveying uncertainty information to users [185, 220, 273, 275, 324]. MIR is another fundamental task in MIA, and a key step in many clinical applications [300]. A few of the prominent clinical uses of image registration include fusing information from multiple imaging modalities, tracking the progression of a medical condition by registering images taken at different times, and constructing anatomical atlases for use in segmentation [344]. Uncertainty quantification in MIR is useful to help a user identify errors, update parameters, or guide subsequent analyses [112, 117, 120, 146, 170, 182, 206, 222, 227, 231, 255, 256, 289, 317]. Some MIR techniques explicitly encode uncertainty to facilitate this quantification [72, 257, 290].

As anatomical structures are being viewed indirectly through the various mechanisms involved in the imaging process, inaccuracies and approximations are introduced. Because of these, even highly trained human experts often encounter some difficulty in MIA. This difficulty spawns both inter-operator (different experts) and intra-operator (the same expert at different times) variability. Thus the value of uncertainty quantification is not limited to automated results.

In this section, we review uncertainty in MIA. We discuss methods for leveraging uncertainty to aid in MIA in Sec. 2.1.1, discuss how uncertainty is inherent in an image in Sec. 2.1.2, and discuss how uncertainty arises in automated MIA methods in Sec. 2.1.3. Fig. 2.1 provides a schematic of the image acquisition and analysis process, indicating where the uncertainty discussed in Sec. 2.1.2 and Sec. 2.1.3 arises.

2.1.1 Uses of Uncertainty

Uncertainty information is useful for a wide range of medical image analysis tasks [163], and has been utilized to help solve many specific problems. In this section, we provide examples of uncertainty being leveraged to improve results in medical imaging applications. These works serve to exemplify the need to quantify uncertainty in segmentation results, either as additional information for the user or as input into subsequent analysis techniques.

Brain MRIs provide a challenging segmentation problem, as they suffer from large noise levels, low contrast between structures, and partial volume effect (PVE) (see Sec. 2.1.2).
Methods have been proposed to quantify uncertainty in brain segmentation results [30, 111, 290]. These uncertainty values are used, for example, to identify poor segmentations and discard them from subsequent statistical analyses. The diagnostic uses of uncertainty in cardiac MRI segmentations are discussed by Shi et al. [287]. Uncertainty is inherent in radiation dosages in radiotherapy. Combining this uncertainty with the uncertainty in the existence of tumors and other physical pathologies can be used in treatment planning and to avoid excessive radiation exposure [107, 145, 256, 338]. Dynamic positron emission tomography (dPET) brain images often have especially low signal-to-noise ratio (SNR) and prevalent PVE. Uncertainty values have been leveraged in dPET segmentation results to both improve the segmentation in an iterative procedure and to fit kinetic models to images [274, 276]. Uncertainty encoded in pixel labelings has been used to aid in breast cancer detection [59,189]. The uncertainty in diffusion values are important to the analysis of diffusion tensor (DT) MRIs [49,50,110,146].

Table 2.1 categorizes works according to how they utilize uncertainty information. Even though many applications utilize uncertainty information already, improved methods for extracting, quantifying, or visualizing meaningful uncertainty information may result in
even more widespread usage. To this end, we will now overview how uncertainty arises in 
MIS tasks and how it can be quantified.

2.1.2 Inherent Uncertainty

While anatomical structures typically have exact, well-defined boundaries, medical images 
are only approximations of the physical reality. As stated, manual segmentations will of-
ten suffer from inter- and even intra-operator variability, and thus work has been done 
towards capturing uncertainty even in manual segmentations [346]. This variability may 
be attributable to inherent uncertainty in an image. With uncertainty existing even in 
GT manual segmentations, every automated segmentation method must also suffer from 
some uncertainty in its results. When computational or physical phantoms (for which the 
GT segmentation is known) are used [73, 76, 136, 168], inherent uncertainty still exists, as 
the phantoms model the errors introduced in image acquisition. Thus, including accurate 
uncertainty information can be beneficial for almost any segmentation result.

One source of inherent uncertainty is imaging noise. In general, a medical imaging 
technique discretizes space into a rectilinear grid, with each grid element referred to as a 
pixel. Each pixel is assigned some value (often scalar) representing a measurement made by 
the imaging tool at that location.

Unfortunately, the image acquisition tools are not perfect and can provide erroneous 
values. This measurement error often takes the form of random noise in the image and 
can be modeled by statistical distributions [20, 29, 320]. Image noise artifacts also results 
from movement of the subject being imaged. Such movement is almost impossible to avoid 
completely, as even sedated humans exhibit movement (e.g. when breathing) [202, 296, 322].

Although there exists methods that attempt to remove noise [54, 63, 236, 270, 272, 347],

<table>
<thead>
<tr>
<th>Uses of Uncertainty in Medical Image Analysis</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visualization</td>
<td>[184, 185, 220]</td>
</tr>
<tr>
<td>Segmentation Improvement</td>
<td>[59, 273–276, 324]</td>
</tr>
<tr>
<td>Brain MRI Analysis</td>
<td>[30, 111, 290]</td>
</tr>
<tr>
<td>Cardiac MRI Analysis</td>
<td>[287]</td>
</tr>
<tr>
<td>Radiation Therapy</td>
<td>[107, 145, 256, 338]</td>
</tr>
<tr>
<td>Mammogram Analysis</td>
<td>[59, 189]</td>
</tr>
<tr>
<td>DTMRI Analysis</td>
<td>[110, 146]</td>
</tr>
</tbody>
</table>

Table 2.1: Sample works utilizing uncertainty for various purposes.
the result is at best an approximation of the noise-free image. Around the boundary of structures of interest, it can be impossible to determine with absolute certainty if a pixel is part of that region or just appears that way as the result of noise. If one neglects uncertainty information in a segmentation, some information will be lost.

Another source of inherent uncertainty is PVE and discretization errors. As stated, medical imaging techniques discretize space into pixels. As a result, it is inevitable that some of these pixels will lie on the boundary between one or more structures of interest. The measurement recorded at such a boundary pixel is likely to be a blending of the measurements that would have been taken at its component structures. This blurring is referred to as partial volume effect (PVE), and results in pixels that are intrinsically part of more than one structure \[250,278,298\]. Instead of assigning pixels to multiple structures, a more descriptive technique is to allow pixels to be fractionally assigned to structures, indicating how much of a pixel’s area or volume belongs to each structure.

However, even in a noise-free image, PVE may result in segmentation uncertainty, as the relationship between the imaging measurement made at a pixel and the fractional make up of that pixel is not always clear. Further, even with fractional label assignments, how a pixel is geometrically divided is still uncertain.

2.1.3 Automated Image Analysis Errors

As stated above, while automated MIS methods continue to advance, no method can yet compete with the ability of a human expert. Thus, the aims of MIS methods are to assist, not replace, an expert user. The more information made available to the user, the more useful the method can be. It is valuable for a method to not only be able to provide a plausible segmentation, but provide a measure of certainty in the results.

A human expert can construct a segmentation by drawing on a rich knowledge of anatomy to execute a complex decision making process. This decision making process is not fully understood, so automated MIS methods can only approximate it by using simplified models. In general, every automatic segmentation framework consists of three components: a way to represent segmentations of an image, a measure that evaluates the correctness of a segmentation given an image, and a scheme to find a segmentation that is in some way optimal with respect to this measure. Below, we discuss in general how uncertainty can arise in automated MIS methods as a result of decisions made for each of these three components. Specific methods are discussed in more detail in Sec. 2.2.
CHAPTER 2. REVIEW

Fuzzy Shape Representations

<table>
<thead>
<tr>
<th>Landmark Points</th>
<th>[55, 183, 228, 343]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medial Axes</td>
<td>[219, 345]</td>
</tr>
<tr>
<td>Active Contours</td>
<td>[55, 144]</td>
</tr>
<tr>
<td>Pixel Labelings</td>
<td>[66, 67, 82, 86, 123, 124, 126, 167, 241, 286, 291, 293, 294, 332]</td>
</tr>
</tbody>
</table>

Table 2.2: Fuzzy versions of popular segmentation representations.

Fuzzy Representations

A segmentation consists of representations for one or more structures in an image. We call a representation crisp if it assigns a discrete label, corresponding to a structure of interest, to every point in the image domain\(^1\). Crisp representations do not explicitly encode uncertainty, so if a method is concerned with leveraging uncertainty information for analysis, quantifying uncertainty for a user, or utilizing uncertainty in some other way, that method would have to extend crisp representations in some way. The most direct way to do this is to use a fuzzification of a crisp representation [36, 144, 219, 332]. The two primary methods for fuzzification are by assigning probabilistic fractional labels to spatial locations or by assigning probabilistic spatial distributions to structural boundaries. Table 2.2 categorizes works based on the probabilistic shape representation they use. Similar fuzzifications can also be performed to registration representations, e.g. by assigning a spatial distribution to each pixel [72].

Energy Minimization

The measure by which a segmentation (or registration) is evaluated can be used to implicitly encode uncertainty information. This measure can usually be cast as an energy function over the space of possible segmentations, which in turn can be split into two components. The first component measures how well a segmentation matches features in an image. This is often described as the external energy and includes all energy terms that vary with the image data. The second component measures how closely shapes conform to prior models, and is often described as the internal energy. For an image \(J\) and a space of possible

\(^1\)The image domain can be treated as a discrete set of pixels or a continuous subset of a Euclidean space.
Figure 2.2: A heuristic example of how uncertainty information is encoded in energy functions. While both $E_1$ and $E_2$ have the same minimizing segmentation, $E_1$ quickly increases for segmentations near the minimum, while in $E_2$ segmentations near the minimum all have a similar energy value. This may imply more uncertainty in the optimal segmentation when using $E_2$, as small perturbations in the image (e.g. additional noise) might noticeably change the minimum of the energy.

User input into a segmentation method is either used to constrain $S(J)$ during the optimization or incorporated into the external energy term. We note that in general, finding $S^*$ may be computationally infeasible, so often a local minimum of the energy must suffice. Similarly, in MIR, for a pair of images $J_1$ and $J_2$ and a space of possible deformations of $D(J_1)$, the pairwise (as opposed to groupwise) image registration problem can be written as

$$D^* = \arg\min_{D \in D(J_1)} E(D, J_1, J_2) = \arg\min_{D \in D(J_1)} (E_{external}(D, J_1, J_2) + E_{internal}(D)) , \quad (2.2)$$

where $E_{external}$ measures how well the deformed images match and $E_{internal}$ attempts to ensure a plausible (e.g. smooth) deformation. More specific examples of these energy terms for segmentation will be discussed in Sec. 2.2 for MIS and in Sec. 2.3 for MIR; for now we keep the formulation general.

While $E(S, J)$ need only be some metric measuring the “correctness” of a segmentation, it can be interpreted as a distribution over segmentations [30, 312] by invoking the Boltzmann
CHAPTER 2. REVIEW

<table>
<thead>
<tr>
<th>Internal Energy</th>
<th>External Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Anatomical/</td>
</tr>
<tr>
<td></td>
<td>Acquisition</td>
</tr>
<tr>
<td></td>
<td>Statistics Over</td>
</tr>
<tr>
<td></td>
<td>Intensity Values</td>
</tr>
<tr>
<td>Anatomical/</td>
<td>[14, 21, 43, 55]</td>
</tr>
<tr>
<td>Acquisition</td>
<td>[12, 33, 38, 39]</td>
</tr>
<tr>
<td>[58, 64, 101, 103]</td>
<td>[82, 123, 234, 249]</td>
</tr>
<tr>
<td>[159, 175, 286, 311]</td>
<td>[265, 353, 354, 361]</td>
</tr>
<tr>
<td>Gaussian Distributions</td>
<td>[45, 80, 86, 177]</td>
</tr>
<tr>
<td>Over Shape Reps</td>
<td>[77, 78]</td>
</tr>
<tr>
<td></td>
<td>[172, 306]</td>
</tr>
<tr>
<td></td>
<td>[28, 89, 173, 174]</td>
</tr>
<tr>
<td>[241, 254, 326, 327]</td>
<td>[230, 282, 357]</td>
</tr>
<tr>
<td>Other Distributions</td>
<td>[65, 79, 84, 87]</td>
</tr>
<tr>
<td>Over Shape Reps</td>
<td>[91, 109, 239, 266]</td>
</tr>
<tr>
<td></td>
<td>[188, 334]</td>
</tr>
<tr>
<td></td>
<td>[42, 47]</td>
</tr>
<tr>
<td>Distributions Over</td>
<td>[297]</td>
</tr>
<tr>
<td>Shape Features</td>
<td>[96, 148, 154]</td>
</tr>
<tr>
<td></td>
<td>[264, 340]</td>
</tr>
</tbody>
</table>

Table 2.3: Categorization of segmentation methods based on their internal and external energy terms. While energy terms based on models of the anatomy (e.g. boundary smoothness) or the image acquisition process (e.g. noise distribution) may sometimes provide better results, statistically motivated energy terms can give a more accurate measure of uncertainty when treated as a distribution over segmentations (see (2.5)). It is often difficult to model distributions over image data or shape representations directly, therefore sometimes distributions are modeled over certain features. We also categorize methods based on the complexity of the distribution they use to model shape variability; while energy terms derived from Gaussians are simple and relatively easy to optimize, more complex distributions may model the variability in shapes more accurately.

distribution, \( Pr(S|J) = \exp(-E(S, J))/Z(J) \), where \( Z(J) \) is a normalization term. The minimization problem in (2.1) becomes a maximum a posteriori (MAP) problem:

\[
S^* = \underset{S \in \mathcal{S}(J)}{\text{argmax}} \ Pr(S|J) \tag{2.3}
\]

\[
S^* = \underset{S \in \mathcal{S}(J)}{\text{argmax}} \frac{Pr(J|S) \cdot Pr(S)}{Z(J)} \tag{2.4}
\]

\[
S^* = \underset{S \in \mathcal{S}(J)}{\text{argmax}} \frac{\exp(-E_{\text{external}}(S, J)) \cdot \exp(-E_{\text{internal}}(S))}{Z(J)} . \tag{2.5}
\]

This general probabilistic formulation of the segmentation problem makes clear the implicit uncertainty in segmentation algorithms: \( Pr(S|J) \) represents a quantification of our belief that \( S \) is the correct segmentation, and while the MAP estimate \( S^* \) may maximize this belief, other segmentations still have non-zero probability, and thus we are not certain that \( S^* \) is correct. When only the MAP segmentation is considered, information encoded in the distribution is lost. By considering the behaviour of the distribution \( Pr(S|J) \), we gain
Table 2.4: Categorization of segmentation methods using statistical shape priors based on the optimizability of their energy functions. The ability to globally optimize an energy function implies that the space of possible segmentations can be efficiently searched in some way, and thus global uncertainty information could be extracted from that energy function (e.g. [30]). When only local minima are guaranteed, only local uncertainty information can be captured. We allow pose to be known a priori as globally optimizing shape and pose parameters simultaneously is a difficult problem and is largely unsolved. The methods are also categorized by segmentation representation and shape distribution, as these determine the domain of the energy function, and even convex energy functions can be difficult to optimize over non-convex domains.

information about how relatively confident we are that a segmentation is correct, which can be used to quantify uncertainty [30]. A similar interpretation can be applied to registration algorithms using (2.2) [170,258]

We note that many methods using an energy minimization framework make no claim as to the statistical validity of the underlying distribution given by (2.5), instead relying on energy terms derived from models of the underlying anatomy or image acquisition process. This does not preclude using their energy functions to measure uncertainty, as the values assigned by the energy function are assumed to still be a measure of segmentation correctness. For example, if minor changes in the image data tend to noticeably change the minimum of $E(S, J)$, this indicates uncertainty in $S^*$ (Fig. 2.2). Other methods, however, do base their energy terms on rigorous probabilistic models derived from statistical analysis of training data, giving more meaningful uncertainty interpretations. Table 2.3 categorizes segmentation methods based on how their energy terms are calculated.
Global and Local Optima

If an energy function can be globally minimized, then it is often possible to sample from the corresponding probabilistic model (see (2.5)), as global minimization requires implicit knowledge of the entire space of possible solutions. For example, if the minimization scheme used to minimize an energy does not get stuck in local minima, then some Markov chain Monte Carlo methods should also be able to avoid them, remaining ergodic and exploring the entire solution space. These probabilistic models can be used to sample segmentations or registrations according to our belief they are correct, thus giving a single uncertainty quantification for the result (as opposed to spatially localized uncertainty).

Unfortunately, many descriptive energy functions or segmentation spaces are non-convex, meaning it may not be possible to find a global minimizer. In this case, a local minimum of $E$ must suffice. Which local minimum is found depends on the user input and on the optimization scheme used. The lack of knowledge about the global minimum of the energy leads to uncertainty in the results, as it may be the case that the global minimum is significantly better than the best local minimum found. This uncertainty is difficult to quantify, however, as it would require knowledge about the global minimum of the energy. The best that can be done is to evaluate how successfully the optimization scheme is at finding good local minima. The probabilistic models corresponding to these energy functions may still be used to samples from the neighborhood around a local minimum, giving local uncertainty information. In Table 2.4 we categorize popular segmentation methods based on whether or not they are guaranteed to find a global optima of their energy functions.

2.2 Fully Automated and Interactive Segmentation

MIS is the task of identifying anatomical structures in an image by assigning labels to the pixels. In this section, we overview fully and semi-automatic segmentation schemes. We begin by identifying four properties we deem important for an automated MIS algorithms to possess. These properties will provide context for which to discuss fully automated MIS.

1. The algorithm should be robust to initialization. If an algorithm’s results depend on how it is initialized, then a good initialization is required to ensure accurate results. However, any required initialization must be set automatically, which introduces the new problem of how to automatically choose a good initialization. Further, if the
same initialization is not chosen consistently, the results may not be consistent. Many segmentation algorithms can be formulated as energy minimization problems. If the energy being minimized is strictly convex, a single globally optimal solution can be found regardless of initialization, guaranteeing robustness.

2. The algorithm should employ an anatomical shape model [45, 86, 87, 91, 138, 177, 326]. In medical images, different anatomical structures may be difficult to distinguish based on image features alone (e.g. due to similar intensity profiles, partial volume effects obscuring boundaries). However, the shapes of anatomical structures are often consistent across populations, so prior shape knowledge can greatly assist in accurate segmentation. Interactive segmentation algorithms benefit from the prior anatomical knowledge of human experts, and thus can often avoid explicitly utilizing shape models.

3. The algorithm should be designed to segment multiple structures simultaneously [24, 52, 122]. Even in applications where only one anatomical structure is of interest, segmenting multiple structures simultaneously provides global context by allowing structures to be identified by their relative locations, particularly in conjunction with property 2. Multi-label segmentation also compensates for structures having similar image features by allowing all choices of labels for a voxel to be evaluated simultaneously. Again, in interactive segmentation this global context is implicitly provided by a user.

4. The algorithm should provide auxiliary information useful for subsequent analysis. Since fully automated segmentation algorithms are not yet sufficiently accurate and reliable to provide results usable in many clinical settings without subsequent verification and correction by human experts, auxiliary information is useful. We specifically focus on probabilistic segmentations, which assign each pixel a distribution over possible labels and thus provide spatially localized uncertainty information that can be used to identify errors in the segmentation and pathological anatomy (e.g. if an anatomical shape model does not match an image) [59, 273–275, 324, 346]. A key part of ensuring statistically meaningful uncertainty is to propagate information to the segmentation from statistical shape and image feature models.
While many segmentation techniques incorporate some of these four properties, the properties are all interrelated and incorporating them all into a useful segmentation framework is challenging. The ease with which these properties can be included in a segmentation algorithm depends heavily on how a segmentation is represented, e.g. if the space of possible segmentations is not convex, it may be difficult to globally optimize an energy defined over that space. Table 2.5 summarizes what properties are included in some popular automatic segmentation algorithms.

While we focus on these four properties due to their relevance to automated MIS and the lack of existing methods incorporating all four, there are many other important properties to consider when constructing a MIS algorithm:

- The algorithm should consistently produce accurate results (verified using a large, anatomically representative data set).
- The algorithm should be efficient, running as quickly as possible.
- The algorithm should be useful for a variety of segmentation applications.
- The algorithm should minimize the amount of user interaction.
- The algorithm should employ a minimal number of hyper-parameters that need to be tuned, be insensitive to the values of these hyper-parameters, or the hyper-parameters should have clear intuitive meanings.

These properties depend on how the “optimal” segmentation is defined, given a space of possible segmentations; often, trade-offs must be made between them. For example, an energy that can be optimized quickly and efficiently may produce less accurate results, or an energy that performs quickly and accurately for a certain problem may not generalize to other problems. Thus, these properties must be all be considered when building a segmentation algorithm. In particular, there has been significant work done towards how to best incorporate user input into MIS.

No fully automated MIS algorithm provides adequate accuracy in all medical applications, and results must be verified by a human expert (hence the inclusion of uncertainty as a property). Further, even some fully automated MIS algorithms require user input “external” to the algorithm, in the form of parameter setting or initialization techniques. Thus semi-automatic techniques are often necessary. Many semi-automatic techniques employ repeated
CHAPTER 2. REVIEW

Table 2.5: A comparison of certain properties of popular algorithms.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Convex Energy</th>
<th>Shape Prior</th>
<th>Multi-label</th>
<th>Probabilistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>[63]</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>[147]</td>
<td>✓</td>
<td>X</td>
<td>✓†</td>
<td>X</td>
</tr>
<tr>
<td>[334]</td>
<td>✓</td>
<td>✓†</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>[340]</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>[96,237]</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>[175,238,356]</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>[297]</td>
<td>✓</td>
<td>✓†</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>[51,52]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[241]</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[123]</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[86]</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
</tbody>
</table>

† Allows only limited regional interaction terms.
‡ Only applicable to restricted classes of shapes.

user interaction to guarantee the accuracy required for medical applications [156, 216]. At a high level, interactive algorithms can be divided into several classes, based on how they accept user input.

1. The user specifies an approximate boundary, which evolves towards the correct segmentation by minimizing a cost function derived from shape priors and image information [64,159].

2. The user specifies sequential points on or near the boundary, and then the boundary is filled in between these points using a minimal path approach [13,21,74].

3. The user provides seeds, or pixels within specific regions, and then uses these seeds as a basis for the segmentation [38,123].

These different classes of user interaction and the four properties of fully automated segmentation algorithms are both related to the representation used for a segmentation. Thus, the remainder of this section reviews popular segmentation representations and provides context to the points discussed above.
2.2.1 Point Distribution Models

A shape can be represented by specifying a set of points, often on its boundary, sometimes referred to as landmark points. Landmark points come in three general types [80]:

1. anatomically meaningful locations,
2. geometric locations (e.g. leftmost point of an object),
3. points interpolated from the first two types.

Landmarks are in general a point set, along with connectivity information, that defines a 2D or 3D shape [80, 138].

Shape uncertainty can be incorporated into a point set by specifying a distribution over the displacements at each point relative to some fixed coordinate frame. Such a distribution is often constructed using a set of training point sets, and is termed a point distribution model (PDM). When constructing a distribution over some shape representation, it is often desirable to remove variability due to the pose of the shape (e.g. translation, rotation, scale) and only capture the remaining variability in the shape. Thus, the point sets are often aligned using the Procrustes method, and then principal component analysis (PCA) is performed on their coordinates to extract the principal modes of variation [80]. As PCA assumes a Gaussian distribution for its input data, which does not always hold for the variability found in anatomical shapes, other statistical techniques have been applied to more accurately capture shape variability in PDMs [79, 84, 94, 292, 331, 333]. Once a distribution for PDMs has been constructed, it can be used as an internal energy term to constrain segmentations to viable shapes.

Matching a landmark point set to appropriate image features remains a difficult problem, as the pose aligning a PDM to an image must be determined. External energy terms can be leveraged to find an appropriate pose. Some external energy terms are boundary-based, where image features are used to determine appropriate landmark locations [28, 42, 78, 80, 172, 173, 230]. Given a good initialization, these priors are used to iteratively search for likely landmark points and realign the shape model. More recently, methods have been developed to mitigate initialization dependence by training classifiers to find potential landmark positions anywhere in the image automatically [88, 105, 341].

Regional features from the interior of training shapes have also been used to align PDMs to structures of interest in an image. One of the most popular of these methods is active
appearance models, introduced by Cootes et al. [77], where the entire interior of the image is used as a feature. Other works propose different regional features and distributions over these features, either to increase efficiency or more accurately model the structures of interest [89, 174, 282, 306, 357].

Since the distributions over boundary and regional features can be constructed in a probabilistic framework [137], the energy terms determining the optimal location for the point set in an image can be used to derive a measure of uncertainty. This concept has been extended to probabilistic landmark points, where a discrete point location is replaced by a distribution over locations [183, 228, 343]. Of course, since only a locally minimizing point set is guaranteed, it is difficult to quantify how certain a shape boundary is compared to other local minima (see Table 2.4).

2.2.2 Medial Representations

Medial axes are another popular way to represent segmentations. In this representation, introduced by Blum [35], shapes are uniquely defined by a medial axis, represented by a set of points equidistant from two or more points on the shape boundary, and a radius function giving the distance to the boundary for each point on the medial axis. Medial representations can often provide a more compact shape description, and for some applications, they are better at explicitly representing common shape deformations, such as bending or elongation.

Medial representations were extended to 3D M-Reps by Pizer et al. [235] and utilized in segmentation methods [47, 154, 200, 234]. Fletcher et al. introduce principal geodesic analysis to build statistical M-Rep shape priors [109, 288]. Fuzzy versions of medial axis representations have also been developed [219, 345].

2.2.3 Explicit Deformable Models

A deformable model is a segmentation representation that can be continuously deformed under the influence of forces, often defined as the negative gradients of energy functionals with respect to the representation. We use the term explicit to refer to deformable models that provide a direct mapping from some parameter space to a shape’s boundary.

One of the first and most well known uses of deformable models for image segmentation was introduced by Kass et al. [159], where the boundary of a 2D structure is explicitly represented by a parametric close contour, \( \mathbf{v}(t) = (x(t), y(t)) \in \mathbb{R}^2 \), for \( t \in [0, 1] \). This
contour is often referred to as a *snake*. A snake is initialized somewhere in the image and then deformed according to the gradients of energy terms (see (2.1)). External energy terms attempt to align the snake with edges in the image, where edges are defined, for example, by large intensity gradients:

\[ E_1(v) = w_1 \int_0^1 |\nabla(G * J(v(t)))| dt , \]

where \( w_1 \) is a scalar weight, \( J \) is the image, \( G \) is a Gaussian smoothing filter, and \( \nabla \) is a gradient operator. Internal energy terms avoid sharp edges in the snake by penalizing large derivatives of \( v \) with respect to \( t \), for example:

\[ E_2(v) = w_2 \int_0^1 |v'(t)| dt , \]

where \( w_2 \) is a scalar weight. Given energy functionals of the form (2.6) and (2.7) (or other similar forms), variational calculus can be used to derive update equations for \( v \), thus deforming the snake from some initial state to a local minimum of the energy:

\[ \frac{\partial v}{\partial \tau} = -\frac{\partial E_1}{\partial v} - \frac{\partial E_2}{\partial v} , \]

where \( \tau \) is an artificial time parameter.

Snakes are simple and direct representations of a structure’s boundary, and thus lend themselves to manual manipulation and fine tuning [159]. However, the segmentation accuracy is often dependent on the initialization, as only local minima of the energy are guaranteed. Thus, various additional energy terms have been proposed to improve the accuracy of the local minimum found. An inflation term has been used to push the snake outward from an initialization inside the target shape in order to avoid spurious local minima [75]. Simulated annealing has been applied to improve the minimization process [127, 242]. Other boundary-based energy terms have been introduced [311], along with techniques for including region-based energy terms [60, 113, 128, 149, 153]. Snakes have been extended to vector valued images [279]. The “live-wire” segmentation method uses a shape representation similar to snakes but is initialized by the user placing a set of points along the structure boundary [21, 209].

While snakes and similar shape representations are continuous in theory, discretizations must be used in practice. Discretization is usually done by expressing the shape representation in terms of some set of basis functions. Various choices of basis functions exist with differing
numerical properties such as finite differences [248], geometric splines [106], Fourier bases [18,302], and others [179,180].

Even with improved energy terms, the snake shape representation suffers from some inherent drawbacks. Snakes cannot represent more topologically complex shapes (e.g. shapes with multiple parts or holes). However, more advanced frameworks have been developed to mitigate this problem [171, 199, 353]. As a parametric curve, the snake representation of a boundary is not unique, and some basic functions over shapes (e.g. the distance between two shapes) may be parameterization dependent. Work has been done on developing alternative continuous parametric representations for 2D closed curves that are less sensitive to the choice of parameterization [301].

Many applications in medical imaging require 3D shape representations, whereas the basic snake representation only handles 2D segmentation. Boundary-based deformable models were generalized to 3D [321], where 3D shapes are fit to 2D profiles. One solution for volumetric image segmentation is to perform slice by slice segmentation of a volumetric image, using the segmentation of one slice as the initialization of the next [75]. Deformed 3D shapes, like spheres and cylinders, have been used as a 3D analogue to a snake [75, 197, 203], and volumetric boundaries have been represented using simplex meshes [95]. Various other 3D boundary-based deformable shape representations have been proposed, many based on different choices of basis functions to be used in the discretizations [303, 313]. Also, many of the extensions to 2D snakes have analogues for these 3D representations, such as topological adaptability [196] and the live-wire paradigm [135, 244].

Some works have focused on probabilistic versions of snakes that incorporate uncertainty information [14, 55, 144]. Other works [198, 229] have cast snake energy minimization in a probabilistic framework (see (2.5)), which implicitly provides some measure of uncertainty. In many cases, however, the energy terms are not statistically based, but rather intuitively defined (see (2.6) and (2.7)).

### 2.2.4 Implicit Deformable Models

Implicit deformable models are a popular alternative to the explicit deformable models discussed above. As opposed to explicit representations, implicit shape representations define some function over the entire image and encode shapes using features of this function.

The most popular implicit shape representation is the level set method, introduced as a technique for curve evolution by Osher and Sethian [218], where the boundary of a shape
is defined as the zero level set of a function, and the inside and outside of the shape have opposite signs. If $\Omega$ is the image domain and $\Omega_1 \subset \Omega$ is the region occupied by the structure of interest, then a level set function is defined as $\phi : \Omega \rightarrow \mathbb{R}$ such that

$$\Omega_1 = \{x \in \Omega | \phi(x) > 0\}.$$  \hspace{1cm} (2.9)

Note that in some works, the sign of $\phi$ is flipped, but in this report we use the above convention. An energy functional $E(\phi)$ is defined over the space of level set functions of an image (see (2.1)), and a level set function is evolved according to the gradients of the energy:

$$\frac{\partial \phi}{\partial \tau} = -\frac{\partial E}{\partial \phi},$$ \hspace{1cm} (2.10)

where $\tau$ is an artificial time parameter.

Level set representations solve several of the issues of explicit deformable models, most notably they are parameterization free, they can represent arbitrary topologies without re-initialization, and they extend seamlessly to higher dimensional images. Energy functionals used in level set based formulations have been shown to often have less local minima than energy functionals over explicit shape representations, and thus level set formulations are less sensitive to initialization [44, 85]. More recently, work has been done connecting the update equations for level set functions to the corresponding snake update equations [69].

Level set functions have seen significant use in image segmentation [57, 190, 251, 352]. Snake-like energy terms were soon formulated in a level set framework [58, 160]. Caselles et al. [58] use some “edge detector” function $g$ and update the level set function $\phi$ by

$$\frac{\partial \phi}{\partial \tau} = |\nabla \phi| \text{ div} \left( g(J) \frac{\nabla \phi}{|\nabla \phi|} \right),$$ \hspace{1cm} (2.11)

where div is a divergence operator. This method is known as Geodesic Active Contours, as the energy can be interpreted as the length of the level set (i.e. the shape boundary) in a Riemannian space whose metric is constructed based on the image. Various other boundary-based energy terms have been proposed, many based on the total variation (TV) of a level set function [43, 175]. For a differentiable function $\phi : \Omega \rightarrow \mathbb{R}$, the TV is defined as

$$TV(\phi, \Omega) = \int_{\Omega} |\nabla \phi(x)| dx.$$ \hspace{1cm} (2.12)
TV has several favorable properties, for example it can be efficiently minimized using dual formulations [61], and it extends to discontinuous functions [62].

Chan and Vese provided an alternative to boundary-based energy terms when they introduced Active Contours Without Edges (ACWE) [64], where the external energy term is based on the regional intensities inside and outside of the level set contour, and does not explicitly force the contour to lie on an image edge. Their proposed external energy functional utilizes the Heaviside function $H$ to transform the level set function $\phi$ into an indicator function for the structure of interest:

$$E_{\text{ext-ACWE}}(\phi) = \int_{\Omega} H(\phi)|c_1 - J(x)|^2 + (1 - H(\phi))|c_2 - J(x)|^2 dx,$$

where $c_1$ and $c_2$ are constant approximations to the intensity inside and outside the shape, respectively, and are iteratively recalculated from the current shape. The internal energy attempts to minimize the boundary length:

$$E_{\text{int-ACWE}}(\phi) = \int_{\Omega} |\nabla H(\phi(x))| dx,$$

making this energy minimization problem equivalent to the piecewise constant version of the popular Mumford-Shah image segmentation objective function [211].

When cast as a probabilistic formulation (see (2.5)), we can think of (2.13) as modeling Gaussians over the intensity distributions inside and outside a structure, and updating their means through an expectation-maximization (EM) process. We can think of (2.14) as an exponential distribution over the boundary length. This probabilistic interpretation helps obviate the uncertainty in the ACWE shape boundary. Chan et al. [63] showed that given fixed $c_1$ and $c_2$ and a properly regularized Heaviside function, an energy functional given as a weighted sum of $E_{\text{int-ACWE}}(\phi)$ and $E_{\text{ext-ACWE}}(\phi)$ can be globally minimized, thus removing the uncertainty due to local minima. Other regional intensity based energy terms have been introduced [221, 267, 328, 363], where the regional statistics are either computed from the current shape estimation or trained a priori.

The level set framework was extended to assign $K$ different labels to an image by using $K$ different level sets [103, 221, 326, 354, 362], and a more efficient formulation using only $\log_2(K)$ level sets has been introduced as well [336]. Though convex binary problems are not necessarily convex when extended to multi-label segmentation, some works have developed convex minimization schemes for multi-label level set energies by embedding the problem in a higher dimensional space [51, 237].
Internal energy terms for deformable models have been extended to enforce more complex restrictions on shape boundaries. Some of these amount to penalizing geometric features of the boundary other than the length [260], but many involve more complex statistical shape priors derived from training data. Leventon et al. [177] perform PCA on signed distance maps (SDMs) of training shapes to find their mean and principal eigenmodes of variation, and then penalize deviation from these using internal energy terms. While SDMs do not form a linear space, and thus linear combinations of the eigenmodes give non-SDM functions, the level sets of linear combinations of SDM functions still give useful results. Tsai et al. [327] extended these ideas, restricting the segmentation to the span of the eigenmodes, and applied the framework to multi-label segmentation. Other works also apply PCA to implicit shape representations [45], equivalent to assuming a Gaussian distribution over the shape space. More complex distributions have also been introduced [91, 239, 254, 266, 268]. Other works have extended these by only applying level set shape priors locally [65, 87].

Statistical shape priors on implicit shape representations face the challenge of pose estimation, much like shape priors over landmark points. Shape priors must either include information on pose, which is difficult with implicit representations, or include pose parameters in their energy minimization framework, which often results in locally minimizing segmentations. Some works compromise these two approaches, performing a rough image alignment before building a shape prior and implicitly incorporating the remaining pose variability into the shape model [254]. Finding globally optimal pose and shape parameters is still largely an open problem (see Table 2.3), with many current solutions employing relaxations to approximate an optimal solution [122, 341].

The methods discussed above are mainly concerned with finding distributions with meaningful MAP segmentations, and are not directly concerned with the properties of the distributions themselves. Some methods utilize relaxations of crisp segmentations in intermediate steps in their minimization process [63], but, while these relaxed segmentations could be thought of as encoding uncertainty, these methods are not typically concerned with conveying this information to a user. Nevertheless, some recent works have extended level set functions to explicitly encode uncertainty, for example, Pohl et al. interpret a vector of SDMs of multiple shapes as the logarithm of odds (LogOdds) transformation of the probabilistic label at each image location [241]. Probabilistic labels have also recently been incorporated into variational frameworks [178, 210]. In the next section we explore working directly with probabilistic labels.
2.2.5 Discrete Shape Representations

In the previous representations, the image domain is treated as a continuous subset of a Euclidean space, with discretizations only being used for implementation. In discrete segmentation representations, the image is explicitly treated as a collection of pixels, and shapes are represented by labels assigned to these pixels. Labels can be crisp, assigning each pixel to a single structure, or probabilistic, fractionally assigning each pixel to multiple structures. We assume $\Omega$ is discretized with $N = |\Omega|$ pixels. We denote a crisp assignment of $K$ labels to an image as a function $q : \Omega \rightarrow B^K$, where $B^K = [e^1, \ldots, e^K]$ and $e^i \in \mathbb{R}^K$ is the $i^{th}$ standard basis function. Likewise, a probabilistic assignment of $K$ labels can be defined as a function $q : \Omega \rightarrow \mathbb{P}_K$, where $\mathbb{P}_K$ is the unit simplex of dimension $K$:

$$\mathbb{P}_K = \left\{ p = [p_1, p_2, \ldots, p_K]^\top \in \mathbb{R}^K \left| p_i > 0, 1 \leq i \leq K, \sum_{i=1}^{K} p_i = 1 \right. \right\} . \quad (2.15)$$

We note probabilistic labels contain more information than crisp labels, and probabilistic labels can be converted to crisp ones via thresholding. Crisp labels have the benefit of simplicity, taking only $K$ discrete possible values per pixel, but probabilistic representations have the advantage of explicitly encoding uncertainty in a form easily interpretable by a user.

Energy functions for discrete segmentations often take the form of Markov Random Fields (MRFs) or conditional random fields (CRFs) defined over the pixel labels. As such, the energy is a weighted sum of unary terms (dependent on a single label), binary terms (dependent on a pair of labels), and higher order terms (dependent on three or more labels). Unary terms can encode image information in the form of regional priors and binary terms can encode image information regarding image edges. With only unary and binary terms, an energy can often be globally minimized [166], but due to the expressive nature of higher order terms, they are sometimes included even if it means the loss of global optimality.

When only unary and binary terms are employed, a graph structure is often used to represent the problem, with vertices corresponding to pixels and weighted edges corresponding to binary terms. In this image graph, vertices are usually only locally connected, and the edge weights often decrease as the corresponding local image gradient increases. The exact method for determining the edge weights is application dependent, and may be fixed ahead of time or learned from training data [314].

Boykov and Jolly [38] cast the segmentation problem as finding the minimum cut in
CHAPTER 2. REVIEW

an image graph. Such a cut can be efficiently and optimally calculated with max $st$-flow algorithms [165], and the resulting segmentation method is termed Graph Cuts (GC). There have been many extensions to this work, for example, efficient algorithms for computing multi-way cuts, simultaneously segmenting an image into multiple regions [40]. Various features and user input methods have been developed to improve GC results, e.g. by training features to use for edge weights or allowing users to specify relative spatial priors between structures [11, 12, 33, 39, 96, 249, 265]. As GC methods solve for the boundary of a shape, they result in a crisp labeling, though work has been done toward extracting uncertainty from GC results [164].

Normalized cuts are a graph-based segmentation method introduced by Shi and Malik [286] that provides a probabilistic labeling by solving a generalized eigenvalue problem based on the image graph’s Laplacian matrix. Another popular graph-based probabilistic labeling technique is the Random Walker (RW) segmentation algorithm introduced by Grady [123, 124]. The RW algorithm can be interpreted as an interactive version of normalized cuts [126]. The edge weights in the RW algorithm are interpreted as probabilities for random walks on the graph. The RW algorithm also assigns probabilistic labels to pixels, and the probabilistic labeling can be optimally and efficiently found, even for multi-label segmentations.

The GC and RW frameworks have been unified under one larger framework along with several other graph-based segmentation methods [82, 83, 291, 339]. These graph-based segmentation algorithms provide several important advantages over the variational frameworks of deformable models (for example the ability to employ powerful combinatorial solvers that can efficiently find global optima) but they also have some drawbacks (for example metrication errors due to the discretized image domain [162]). However, as continuous deformable models must be discretized when implemented, many of the level set methods are analogous to an optimization scheme over discrete labels, and explicitly treating them as such can provide some computational advantages [101, 125].

Incorporating shape priors into discrete frameworks has been an area of active research [25, 67, 188, 280, 340]. While higher order energy terms can be descriptively powerful [148, 264], they often require that convexity be sacrificed, making their optimization more difficult and initialization dependent. However, there are some exceptions to this rule, where higher order constraints are included while maintaining a guarantee of global optimality [147, 148, 167, 252]. Shape priors for more specific segmentation problems have also achieved success, for example multi-label shape priors built for nested or layered surfaces [297], or priors used
for shapes exhibiting star-convexity [334].

Cremers et al. [86] used PCA based shape priors similar to those used on SDMs for probabilistic pixel labelings and incorporated into a globally optimizable segmentation framework. While the space of probabilistic labelings is convex (unlike the space of SDMs), moving along the PCA eigenmodes may still result in invalid labelings (i.e. pixels assigned label fractions greater than 1 or less than 0). Nevertheless, this method enforces a statistically-based shape prior on probabilistic labelings while maintaining global optimality.

Statistical Analysis of Probabilistic Labels

The probabilistic label of a pixel is an instance of what is known in other areas as a compositional vector, i.e. a positive real vector with some fixed sum. Much research has been done on the statistical analysis of compositional vectors, starting with Aitchison’s work [1]. In this paper, it is shown that the space $\mathbb{P}_K$ is an inner product space with respect to the Aitchison inner product. Aitchison proposed transforming $\mathbb{P}_K$ to a Euclidean space before performing statistical analysis (e.g. PCA). Specifically, Aitchison recommended the centered log-ratio (CLR) transformation and the additive log-ratio (ALR) transformation, both based on taking different ratios of logarithms of the components of compositional vectors. The ALR transformation is in fact equivalent to the LogOdds transformation used by Pohl et al. [241] to interpret SDMs as probabilistic labels. Egozcue et al. [100] propose the isometric log-ratio (ILR) transformation, isometrically mapping $\mathbb{P}_K$ to $\mathbb{R}^{K-1}$. Changizi and Hamarneh applied the ILR transformation to probabilistic labels in shape analysis [66], but outside of this work, the wealth of research on the statistical analysis of general compositional data [3, 6, 32, 108, 193, 194, 223] has been largely passed over in MIS literature.

Comparison of Probabilistic Segmentations

The Dice similarity coefficient (DSC) is often used to evaluate how similar crisp segmentations are, which is used in evaluating automated MIS algorithms via comparison with the GT. Here we describe the DSCs extension to probabilistic segmentations. For crisp sets $\mathcal{S}_1, \mathcal{S}_2 \subset \Omega$, the DSC is defined as

$$D(\mathcal{S}_1, \mathcal{S}_2) = \frac{2|\mathcal{S}_1 \cap \mathcal{S}_2|}{|\mathcal{S}_1| + |\mathcal{S}_2|}. \quad (2.16)$$

We define probabilistic sets as sets that assign a value in $[0, 1]$ to each element of $\Omega$, indicating probability of inclusion. In order to extend the DSC to two probabilistic sets, $q^1, q^2$ :
Ω → [0, 1], we must generalize the notions of set size and set intersection. The standard generalization of set size is to simply sum the probabilities of each element [34]:

\[ |q| = \sum_{x \in \Omega} q(x). \tag{2.17} \]

Set intersection is usually generalized using \textit{t-norms} [34]; here, we use the simple product t-norm, defined for \( a, b \in [0, 1] \) as \( t(a, b) = a \cdot b \). The resulting probabilistic DSC function is then given by:

\[ D(q^1, q^2) = \frac{2 \sum_{x \in \Omega} q^1(x) \cdot q^2(x)}{\sum_{x \in \Omega} q^1(x) + q^2(x)}. \tag{2.18} \]

### 2.3 Deformation Representations for Image Registration

Just as MIS algorithms are dependent on the segmentation representation used, MIR algorithms depend on how deformations are represented. In this thesis, we focus primarily on the regularization of deformations (the internal energy in (2.2)) as opposed to image matching criteria (the external energy in (2.2)). In general, regularization is enforced to ensure a deformation is \textit{anatomically feasible}. How feasibility is defined varies between applications and algorithms. A common requirement is that a deformation be \textit{smooth}, with continuous derivatives up to a certain degree. A stricter requirement is that a deformation \textit{preserves topology}, so that its inverse transformation exists. Finally, some algorithms provide \textit{inverse consistency}, ensuring the resulting deformation is independent of image ordering [16,70]. In this section, we review various techniques for ensuring regularized deformations in MIR. We then review graph-based MIR algorithms in more detail, as they are a prime focus of this thesis.

#### 2.3.1 Regularizing Deformations

**Implicit Regularization**

A common regularization technique is to constrain the space of possible deformation to diffeomorphisms [15, 27, 98] or a subset thereof, implicitly enforcing regularization. One way this is done is to represent a deformation using a set of control points and interpolate between them using basis functions [142]. Examples of popular basis functions include multi-quadric splines [142], thin plate splines [191], and B-splines [121, 271, 330]. Deformations
have been represented using finite elements, discretizing the image domain into a mesh, then finding the optimal displacement at each mesh node \([212, 245]\). Deformations represented using discrete Fourier transformations or wavelets can be regularized by suppressing high-frequency bases \([142, 186, 365]\). Some applications in MIR involve articulation, and benefit from partitioning the image and representing deformations in each partition separately \([232]\).

**Explicit Regularization**

Another strategy for regularizing a deformation is to explicitly penalize irregular deformations in an energy minimization formulation. Many algorithms penalize a Sobolev semi-norm of the deformation \([364]\). Regularizers involving first order derivatives include total variation regularizers \([17]\) and diffusion regularizers \([143]\), penalizing the gradient norm and squared gradient norm, respectively. Penalties involving higher order derivatives include curvature regularizers involving the Laplacian of the deformation \([56, 208]\), physics-based elastic regularizers involving the divergence of the deformation \([46, 259]\), and regularizers involving the curl of the deformation \([68, 299]\).

Other regularizers encourage a deformation to have specific properties, such as a Jacobian close to 1 \([263, 271]\), mass preservation (with image intensity interpreted as mass density) \([92, 133]\), or explicit inverse consistency \([139, 140, 152, 207, 335, 350]\). Alternatively, post-processing techniques such as scaling and squaring \([141]\) can convert a transformation to one that preserves topology.

**Spatially Inhomogeneous Regularization**

The trade-off between explicit regularization and image matching can often be controlled by changing algorithm parameters. Desirable deformation properties, such as topology preservation, are usually defined locally, so a globally constant regularizer may result in over-regularization in some image regions to ensure sufficient regularization in other regions. To prevent this, spatially inhomogeneous regularization techniques have been proposed, allowing regularization strength to be set locally. Spatially inhomogeneous regularization techniques fall into three main categories \([176]\):

1. Each pixel is assigned a different regularization strength, so more confident pixels contribute more to the image matching and less confident pixels allow themselves to be guided \([115, 233, 310]\).
2. Regions in an image are identified and each region is regularized with a different strength, perhaps with regional interfaces regularized weakly [53,93,116,129,155,176,253,262,304,305,351].

3. Different regions are assigned completely different deformation models (e.g. rigid deformations in some regions, elastic deformations in others) [99,131,181,269].

Many graph-based registration algorithms introduced in the next section fall into the first category, allowing the strength of the regularization to be explicitly specified at each graph vertex.

2.3.2 Graph-Based Image Registration

Recently, several registration techniques have moved towards performing registration in the discrete domain [72,121,140,318,319,342], where pixels are assigned labels from a predefined set of displacement vectors. The optimal deformation is found by minimizing a Markov random field (MRF) defined over the image graph. Powerful and efficient discrete optimization techniques (such as linear programming algorithms [121] or GC based algorithms [319]) can be leveraged to optimize these energies, often avoiding problems often encountered in continuous registration formulations, such as becoming stuck in local minima. Unary MRF terms usually encode image similarity information, while binary (or pairwise) MRF terms enforce regularization [81]. Higher order MRF terms have also been used [169,217]. In Sec. 5.2, we will focus on the random walker image registration (RWIR) algorithm [72], a graph-based registration algorithm employing a Gaussian MRF to enforce regularization. The RWIR energy can be globally optimized and provides a probabilistic deformation.

One trade-off of discrete techniques is that the predefined set of displacement vectors needs to be quite large, particularly in 3D tasks, in order to achieve adequate spatial resolution, leading to high computational costs. The requirement for a large label set has been mitigated by iterative approaches designed to incrementally update an initially sparse label set [72,140,317], though such iterative approaches often sacrifice global optimality. Alternatively, a sparser, image-adaptive discretization of the image domain can be used [222,246,315] instead of the standard uniform voxel-based discretization, so fewer spatial locations need to be solved for. These techniques usually employ a multi-resolution approach, starting with a sparse graph and a large set of labels and moving to a fuller graph and a smaller set of labels.
2.4 Chapter Summary

In this chapter we reviewed and summarized previous works that will form the basis for this thesis. Specifically, Chapter 3 will build upon the review of uncertainty in Sec. 2.1, presenting techniques for performing statistical analysis of probabilistic labels and demonstrating the benefits of using techniques appropriate for a given task. Chapter 4 presents a novel fully automatic segmentation framework based on a novel segmentation representation and focused on the four properties for fully automated segmentation outlined in Sec. 2.2. Chapter 5 presents precomputation techniques designed to increase the speed and accuracy of the interactive random walker image segmentation and image registration algorithms, discussed in Sec. 2.2 and Sec. 2.3.
Chapter 3

Statistical Analysis of Probabilistic Labels

As we discussed in Sec. 2.1.3, probabilistic labels for image segmentation encode uncertainty information that has multiple uses in subsequent analyses, such as quantifying error or detecting potential pathologies. Anatomical shape models built using principal component analysis (PCA) on training data are popular in automated medical image segmentation, providing context that would normally be given by a human expert. However, the space of probabilistic labels is constrained, and thus movement along the PCA modes can result in probabilistic labels outside of the simplex (e.g. negative fractions), which require projections back to the simplex (Fig. 3.1). Projections may discard relative uncertainty information, particularly in multi-label probabilistic segmentations, since many probabilities may be projected back to 0.

Fortunately, techniques have been established that avoid this issue. The general idea is to map the probabilistic labels at each voxel to an unconstrained vector space, perform statistical analysis, and then map the results back to the simplex. In this chapter, we describe existing transformations used to perform this mapping, provide intuition behind these transformations in the context of probabilistic image segmentation, and explore how the choice of transformation affects statistical anatomical shape models.
CHAPTER 3. STATISTICAL ANALYSIS OF PROBABILISTIC LABELS

38

Figure 3.1: Performing PCA directly in the simplex results in a PCA mode (green line) that leaves the constrained space. Vertices correspond to “certain” probabilistic labels. See Fig. 3.4 and Fig. 3.7 for comparison, particularly Fig. 3.4 for a description of how the points were generated. Uncertainty is lost when points are projected back to the simplex.

3.1 Aitchison Geometry

3.1.1 Vector Space Structure

We let $K$ represent the number of labels (including background) in a given segmentation task. We define the unit simplex for $K$ probabilistic labels as

$$\mathbb{P}_K = \left\{ p = [p_1, p_2, \ldots, p_K]^\top \in \mathbb{R}^K \mid p_i > 0, 1 \leq i \leq K, \sum_{i=1}^{K} p_i = 1 \right\}. \quad (3.1)$$

$\mathbb{P}_K$ is defined such that each component is non-zero. Cromwell’s rule [150] states that no probabilities should be exactly 0 in a Bayesian framework, as this does not allow for a change in one’s belief, regardless of the evidence observed. A probabilistic segmentation is a function $q : \Omega \rightarrow \mathbb{P}_K$, where $\Omega$ is the discrete image domain, with $N = |\Omega|$ pixels.

The problem of how to best perform statistical analysis on probabilistic labels has been tackled in fields outside of image processing. Probabilistic labels are a type of compositional data, multivariate data where the absolute values of the data points are either unmeasurable or unimportant, and we are instead only concerned with the ratios of values (e.g. how much more likely is one label than another). Compositional data arises in a variety of fields, such as geology (mineral compositions of rocks), environmetrics (pollutant compositions), and economics (household budget compositions).
The insight of Aitchison that compositional data represents relative quantities, and thus should be dealt with in terms of ratios [1, 2], lead to a *stay-in-the-simplex* approach to compositional data analysis [3, 6, 7, 19, 31]. This approach advocates imposing a Hilbert space structure on the simplex, defining intrinsic simplicial distances and inner products [4, 5, 32, 100, 226].

Aitchison showed that $\mathbb{P}_K$ forms a $(K-1)$-dimensional vector space [2] with the *perturbation* and *power transformation* operators acting as vector addition and scalar multiplication, respectively [31, 32, 223, 224]. For $p, q \in \mathbb{P}_K$, perturbation is defined as

$$p \oplus q = C([p_1 q_1, \ldots, p_K q_K]^\top), \quad (3.2)$$

and for a scalar $\alpha$, power transformation is defined as

$$\alpha \odot p = C([p_1^\alpha, \ldots, p_K^\alpha]^\top). \quad (3.3)$$

$C(\cdot)$ is a renormalization operator (also known as “closure”):

$$C(v) = \frac{v}{Z(v)}, \quad Z(v) = \sum_i v_i. \quad (3.4)$$

Perturbation and power transformation satisfy the usual properties of vector addition and scalar multiplication [225]. Fig. 3.2 illustrates the effect of perturbation and power transformation on a set of compositional vectors. The analogue of subtraction is defined as

$$p \ominus q = p \oplus ((-1) \odot q). \quad (3.5)$$

Perturbation and power transformation satisfy the usual properties of vector addition and scalar multiplication. For $p, q, r \in \mathbb{P}_K$, the following properties hold [225]:

1. Associativity of addition: $(p \oplus q) \oplus r = p \oplus (q \oplus r)$,
2. Commutativity of addition: $p \oplus q = q \oplus p$,
3. Additive identity: $((\frac{1}{K}) \odot 1) \oplus p = p$,
4. Additive inverse: $p^{-1} = (-1) \odot p$, so $p \oplus p^{-1} = ((\frac{1}{K}) \cdot 1)$.

For $p, q \in \mathbb{P}_K$ and $\alpha, \beta \in \mathbb{R}$ the following properties hold:

1. Associativity of scalar multiplication: $\alpha \odot (\beta \odot p) = (\alpha \cdot \beta) \odot p,$
Figure 3.2: Effect of perturbation and power transformation on compositional vectors. (a) A set of compositional vectors (blue curve, arrow) are perturbed by various other points in the simplex (red to green dots). The perturbations result in new sets of compositional vectors (red to green curves). (b) Power transformation of the blue curve from (a) with scalars ranging from 0 (magenta curves) to 2 (cyan curves). Note power transformation by 1 leaves the curve unchanged (blue curve, arrow), power transformation by 0 reduces the curve to a point in the center of the simplex, and power transformation by large values shifts the curve to the edge of the simplex.

2. Distributivity of vector sums: \( \alpha \odot (p \oplus q) = (\alpha \odot p) \oplus (\alpha \odot q) \),

3. Distributivity of scalar sums: \( (\alpha + \beta) \odot p = (\alpha \odot p) \oplus (\beta \odot p) \),

4. Multiplicative identity: \( 1 \odot p = p \).

In applications where a compositional vector represents ratios of different quantities, perturbation can correspond to changes through some exponential process (e.g., the decay or growth of bacterial concentrations) [6]. As an analogy to scalar multiplication, power transformation can control the amount or strength of these changes. We can use the power transformation to parameterize the change in a composition caused by a perturbation,

\[ q(s) = q_0 \oplus (s \odot p) , \]

(3.6)

where \( s \) is a time parameter. Such a curve through the simplex is called a compositional line (Fig. 3.3).
CHAPTER 3. STATISTICAL ANALYSIS OF PROBABILISTIC LABELS

Figure 3.3: Orthogonal and parallel compositional lines. Blue compositional lines $\hat{q}_1(s) = q_1 \oplus (s \odot p_1)$ and $\hat{q}_2(s) = q_2 \oplus (s \odot p_1)$ are parallel. These lines represent how different probabilistic labels $q_1 = [0.38, 0.24, 0.38]^T$ and $q_2 = [0.2, 0.6, 0.2]^T$ are updated when an observation gives the likelihood $p_1 = [0.2, 0.3, 0.5]^T$. Note that although $\hat{q}_1(s)$ and $\hat{q}_2(s)$ converge, they never intersect, getting arbitrarily close as $s \to \infty$. Green compositional lines $\hat{q}_3(s) = q_3 \oplus (s \odot p_3)$ and $\hat{q}_4(s) = q_3 \oplus (s \odot p_4)$ are orthogonal, since $\langle p_3, p_4 \rangle_a = 0$. These lines represent how the probabilistic label $q_3$ is updated when an observation gives either the likelihood $p_3 \approx [0.48, 0.32, 0.20]^T$ or the likelihood $p_4 \approx [0.23, 0.53, 0.24]^T$. Since $p_3$ and $p_4$ are orthogonal for $S3$, any other likelihood $r$ can be expressed as $r = (c_3 \odot p_3) \oplus (c_4 \odot p_4)$, where $c_3 = \langle p_3, r \rangle_a / \|p_3\|_a$ and $c_4 = \langle p_4, r \rangle_a / \|p_4\|_a$. Thus, any new observations are equivalent to making the observations corresponding to the likelihoods $p_3$ and $p_4$, $c_3$ and $c_4$ times respectively.
When a compositional vector corresponds to a probabilistic label, another interpretation for perturbation and power transformation arises in the context of Bayesian inference on probabilistic segmentations. Following Pohl et al., we show perturbation can be used to update a prior discrete distribution given new evidence, and that power transformation controls our confidence in a distribution [241]. We let \( L = \{\ell_1, \ldots, \ell_K\} \) be the set of possible labels; we assume a pixel \( x \in \Omega \) has some unknown ground truth label represented by the random variable \( G_x \in L \). A probabilistic segmentation \( q \) gives a discrete distribution for each pixel:

\[
q_i(x) = \Pr(G_x = \ell_i)
\]  

(3.7)

Now we are given an observation, \( t \), and we wish to update the prior distribution \( q(x) \) to the posterior distribution \( \hat{q}(x) \). We let \( T \) be a random variable for \( t \) and apply Bayes’ rule:

\[
\hat{q}_i(x) = \Pr(G_x = \ell_i | T = t) = \frac{\Pr(T = t | G_x = \ell_i) \Pr(G_x = \ell_i)}{\Pr(T = t)}.
\]  

(3.8)

We will keep the exact form of \( t \) general, but as an example, the observation could be a label assigned to \( x \) by a human expert. We may then model \( \Pr(T = t | G_x = \ell_i) \) by assuming some small probability \( \epsilon \) that the label \( t \) is incorrect, giving

\[
\Pr(T = t | G_x = \ell_i) = \begin{cases} 
1 - \epsilon & \text{if } t = \ell_i \\
\epsilon/(K - 1) & \text{otherwise}.
\end{cases}
\]  

(3.10)

Letting \( p_i(x, t) = \Pr(T = t | G_x = \ell_i)/\Pr(T = t) \) and \( p(x, t) = C \left( [p_1(x, t), \ldots, p_L(x, t)]^\top \right) \), we rewrite (3.8) as

\[
\hat{q}_i(x) = p_i(x, t)q_i(x)
\]  

(3.11)

\[
\hat{q}(x) = p(x, t) \oplus q(x).
\]  

(3.12)

That is, calculating a posterior distribution at a pixel is equivalent to performing a perturbation between the prior distribution and the normalized likelihood function. We note that perturbation has built in renormalization (see (3.2)), so the posterior distribution is guaranteed to sum to 1.
CHAPTER 3. STATISTICAL ANALYSIS OF PROBABILISTIC LABELS

In practice one may wish to incorporate spatial dependencies between neighboring pixels when updating the label probabilities, instead of updating each pixel’s probabilities separately as is done in (3.8). The above framework is meant only as an example to provide an interpretation of perturbation.

If, \( m \) times consecutively, we make the same observation \( t \), we perform the perturbation in (3.12) \( m \) times, or equivalently, we perturb \( q(x) \) by \( (m \odot p(x, t)) \). Thus, the power transformation weights how much influence an observation has. For example, if \( t_1 \) is input provided by a reliable human expert and \( t_2 \) is similar input provided by an untrained human, we could use the power transformation to give \( t_1 \) greater influence.

3.1.2 Hilbert Space Structure

By defining an inner product on the simplex along with its associated norm and distance metric, we obtain a Hilbert space structure. The most prevalent choice is the Aitchison inner product [3], which is used in many important works concerning compositional data analysis (though not specific to image analysis) [31,32,100,223,224]. We first define \( \langle \cdot, \cdot \rangle \) as the standard Euclidean inner product and \( \mu(\cdot) \) as the arithmetic mean of a vector. That is, for vectors \( \mathbf{v}, \mathbf{w} \in \mathbb{R}^K \),

\[
\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{i=1}^{K} v_i w_i \tag{3.13}
\]

\[
\mu(\mathbf{v}) = \frac{1}{K} \sum_{i=1}^{K} v_i . \tag{3.14}
\]

The Aitchison inner product between two vectors \( \mathbf{p}, \mathbf{q} \in \mathbb{P}_K \) is defined as:

\[
\langle \mathbf{p}, \mathbf{q} \rangle_a = \langle \log(\mathbf{p}), \log(\mathbf{q}) \rangle - K \cdot \mu(\log(\mathbf{p})) \cdot \mu(\log(\mathbf{q})) , \tag{3.15}
\]

where \( \log(\cdot) \) is the natural logarithm. In a slight abuse of notation, \( \log(\cdot) \) is applied to vectors, denoting a component-wise logarithm:

\[
\log(\mathbf{p}) = [\log(p_1), \ldots, \log(p_K)]^\top . \tag{3.16}
\]

The associated norm and distance are given by:

\[
\|\mathbf{p}\|_a^2 = \langle \mathbf{p}, \mathbf{p} \rangle_a \tag{3.17}
\]

\[
d_a(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} \ominus \mathbf{q}\|_a . \tag{3.18}
\]
The function \(d_a(\cdot, \cdot)\) satisfies the standard properties of a distance metric over a vector space, specifically with respect to vector addition and scalar multiplication (i.e., perturbation and power transformation). That is, for any \(p, q, r \in \mathbb{P}_K\) and \(\alpha \in \mathbb{R}\), \(d_a(\cdot, \cdot)\) satisfies:

1. Translation invariance: \(d_a(r \oplus p, r \oplus q) = d_a(p, q)\),

2. Homogeneity: \(d_a(\alpha \odot p, \alpha \odot q) = |\alpha|d_a(p, q)\).

A distance is in general a quantification of “how much” must be added to one point to get to another. The Aitchison distance between probabilistic labels \(q(x)\) and \(\hat{q}(x)\) is a measure of how much \(q(x)\) must be perturbed to get \(\hat{q}(x)\). Equivalently, returning to the Bayesian interpretation of the Aitchison geometry, \(d_a(q, \hat{q})\) measures how much evidence must be observed to convince ourselves of a posterior \(\hat{q}(x)\) starting from a prior \(q(x)\).

If we assume a likelihood where \(\|p(x, t)\|_a \leq C\) for some constant \(C\) and for all \(t\) (e.g. using the likelihood in (3.10)), then we can make a more concrete statement about the Aitchison distance: to get from prior \(q(x)\) to posterior \(\hat{q}(x)\), we must observe corroborating evidence at least \((d_a(q(x), \hat{q}(x))/C)\) times. A graphic example of perturbing by vectors with different Aitchison norms is seen in Fig. 3.3.

Translation invariance leads to the useful property, shown by Tunc et al. [329], that if we have two posterior distributions derived from the same prior but different observations, \(\hat{q}_1 = p(t_1) \oplus q\) and \(\hat{q}_2 = p(t_2) \oplus q\), the Aitchison distance between them is independent of the prior used: \(d_a(\hat{q}_1, \hat{q}_2) = d_a(p(t_1), p(t_2))\).

An inner product provides a notion of orthogonality. For example, if we have a set of compositional vectors \(P = \{p_1, \ldots, p_K\}\), and another compositional vector \(r\) is orthogonal to each vector in \(P\), then no linear combination (with respect to perturbation and power transformation) of vectors in \(P\) can result in \(r\). If the vectors in \(P\) and \(r\) all correspond to likelihoods from different observations (see (3.12)), \(r\) provides information that is complementary to the information provided by the vectors in \(P\); the evidence provided by \(r\) neither supports nor refutes the evidence provided by \(P\). A graphic example of how the Aitchison inner product can be used to identify parallel and orthogonal likelihood functions is seen in Fig. 3.3.

Instead of working directly with probabilistic labels, a typical strategy is to use a transformation from \(\mathbb{P}_K\) to \(\mathbb{R}^{K-1}\) that maps the Aitchison geometry to standard Euclidean geometry. In the next section we review several such transformations.
CHAPTER 3. STATISTICAL ANALYSIS OF PROBABILISTIC LABELS

3.2 Probabilistic Segmentation Representations

3.2.1 LogOdds Transformation

The LogOdds transformation was introduced for the statistical analysis of probabilistic anatomical shapes by Pohl et al. [241]. In this section we discuss the LogOdds transformation and introduce concepts in simplicial data analysis that we will expand upon in subsequent sections.

To facilitate the creation of statistical models over probabilistic segmentations, Pohl et al. proposed applying the LogOdds transformation to probabilistic labels, mapping them to a vector space. The LogOdds transformation is also referred to as the additive log-ratio (ALR) transform by Aitchison [2], and for a probabilistic label \( p \in \mathbb{P}_K \) it is defined as

\[
\text{LogOdds}(p) = \left[ \log\left(\frac{p_1}{p_K}\right), \ldots, \log\left(\frac{p_{K-1}}{p_K}\right) \right]^\top \tag{3.19}
\]

\[
\text{LogOdds}^{-1}(v) = C\left(\left[\exp(v_1), \ldots, \exp(v_{K-1}), 1\right]^\top\right). \tag{3.20}
\]

Again, \( C(\cdot) \) is a renormalization operator, and \( \exp(\cdot) \) is the natural exponential, the inverse of \( \log(\cdot) \), and is understood to be applied component-wise to vectors. We will adopt the convention of referring to a vector resulting from applying the LogOdds transformation to a probabilistic label as a “LogOdds vector” and the space of all such vectors as “LogOdds space”.

LogOdds provides a bijection between \( \mathbb{P}_K \) and \( \mathbb{R}^{K-1} \), and standard statistical techniques can be performed in LogOdds space without requiring any projections. Further, the LogOdds transformation maps perturbation and power transformation to standard Euclidean vector addition and scalar multiplication. For \( p, q \in \mathbb{P}_K \) and scalar \( \alpha \),

\[
\text{LogOdds}(p \oplus q) = \text{LogOdds}\left(\frac{1}{Z}[p_1q_1, \ldots, p_Kq_K]^\top\right), \text{ where } Z \text{ normalizes}, \tag{3.21}
\]

\[
= \left[ \log\left(\frac{Z}{Z\frac{p_1q_1}{p_Kq_K}}\right), \ldots, \log\left(\frac{Z}{Z\frac{p_{K-1}q_{K-1}}{p_Kq_K}}\right) \right]^\top \tag{3.22}
\]

\[
= \left[ \log\left(\frac{p_1}{p_K}\right) + \log\left(\frac{q_1}{q_K}\right), \ldots, \log\left(\frac{p_{K-1}}{p_K}\right) + \log\left(\frac{q_{K-1}}{q_K}\right) \right]^\top \tag{3.23}
\]

\[
= \text{LogOdds}(p) + \text{LogOdds}(q). \tag{3.24}
\]
LogOdds(α ⊙ q) = LogOdds\left(\frac{1}{Z}[q_1^\alpha, \ldots, q_K^\alpha]^\top\right), \text{ where } Z \text{ normalizes}, \hspace{1cm} (3.25)

= \left[\log \left(\frac{Z q_1^\alpha}{Z q_K^\alpha}\right), \ldots, \log \left(\frac{Z q_{K-1}^\alpha}{Z q_K^\alpha}\right)\right]^\top, \hspace{1cm} (3.26)

= \left[\alpha \log \left(\frac{q_1}{q_K}\right), \ldots, \alpha \log \left(\frac{q_{K-1}}{q_K}\right)\right]^\top, \hspace{1cm} (3.27)

= \alpha \text{LogOdds}(q). \hspace{1cm} (3.28)

Thus, addition and scalar multiplication between LogOdds vectors can be interpreted as performing inference on the underlying probabilistic labels.

Pohl et al. construct a shape model by performing PCA on sets of M probabilistic training segmentations by first applying the LogOdds transformation to their probabilistic labels. With N pixels and K labels, probabilistic segmentations mapped by the LogOdds transformation can be represented as vectors in $\mathbb{R}^{N(K-1)}$. Performing PCA on the M training segmentations provides the mean segmentation and the m < M eigenmodes of greatest variance, which best capture how the training segmentations vary from the mean. The PCA mean and modes approximate the space of feasible segmentations, which can be used, for example, to guide the segmentation of a novel image.

Probabilistic training data is often not available, as is the case when training segmentations are constructed manually. Such “crisp” (non-probabilistic) segmentations could be represented by assigning a binary indicator vector of (K − 1) 0’s and a single 1 to each pixel, but vectors in $\mathbb{P}_K$ cannot contain 0’s, and thus the LogOdds transformation cannot be applied. There are several possibilities for converting a crisp segmentation to a probabilistic one.

Pohl et al. take the approach of constructing probabilistic labels by calculating the signed distance maps (SDMs) of each crisp label (other than the background) and treating the resulting vector of SDMs at each pixel as a LogOdds vector which can be mapped back to a probabilistic label. This approach provides intuitive results because the LogOdds transformation has the convenient property that the $i^{th}$ component of LogOdds($p$) increases and decreases monotonically with $p_i$, so pixels closest to the center of a crisp shape are given probabilities for label $i$ close to 1 and pixels far away from the shape are given probabilities for label $i$ close to 0.

A potential drawback of the LogOdds transformation is that it is symmetric in the first $K - 1$ labels, but not in the $K^{th}$ label (usually the background in segmentation, see (3.19)).
CHAPTER 3. STATISTICAL ANALYSIS OF PROBABILISTIC LABELS

Figure 3.4: Performing PCA in LogOdds space results in a PCA mode (red line) that is skewed towards the last label. The points in (a) are found by: sampling 2000 points $[p_x, p_y]$ from an isotropic 2D Gaussian, skewing the $y$ component using $p_y < -0.7p_y(|p_y|^{0.8})$, applying a random rotation (fixed for all points), and mapping to the simplex using the inverse LogOdds transformation. The same point set is used in Fig. 3.1 and Fig. 3.7.

With $I$ as the $(K - 1) \times (K - 1)$ identity matrix, $1$ as the vector of $(K - 1)$ 1’s, and using $[\ldots]$ to denote a block matrix, we can rewrite (3.19) as the matrix-vector product

$$\text{LogOdds}(p) = A \log(p) \quad (3.29)$$

$$A = [I, -1], \quad (3.30)$$

recalling $\log(\cdot)$ is applied to vectors component-wise.

The LogOdds transformation is a linear combination of vectors in $\mathbb{R}^{K-1}$ with weights $\log(p_i)$. The first $(K - 1)$ vectors are the standard basis vectors of $\mathbb{R}^{K-1}$, $\{e_1, \ldots, e_{K-1}\}$, whereas the $K^{th}$ vector is $-1$. Since $\|e_i\| = 1$ and $\|-1\| = \sqrt{K-1}$, for the binary case $K = 2$ the LogOdds transformation is symmetric between the two labels, but for $K > 2$, changes in the last label’s (background’s) probability are magnified by $\sqrt{K-1}$ when mapped to LogOdds space. This asymmetry primarily manifests itself in probabilistic image segmentation in two ways:

1. Regularizing segmentations typically involves penalizing large gradients or changes in the labeling function. Thus, in LogOdds space foreground-background boundaries will be more strongly regularized than foreground-foreground boundaries.
CHAPTER 3. STATISTICAL ANALYSIS OF PROBABILISTIC LABELS

2. When PCA is used to construct shape models, greater variance is detected around foreground-background boundaries than around foreground-foreground boundaries, leading to these boundaries being better represented by the resulting shape models. These biases are demonstrated in Sec. 3.2.3.

This asymmetry means that distances between LogOdds vectors do not correspond to the (symmetric) Aitchison distance between the underlying probabilistic labels, but rather to a skewed distance metric. Fortunately, Egozcue et al. developed another log-ratio transformation that does map the Aitchison distance to the Euclidean distance, and is thus known as the isometric log-ratio transformation [100].

3.2.2 Isometric Log-Ratio Transformation

Egozcue et al. proposed the isometric log-ratio (ILR) transformation [100], defined as

$$\text{ILR}(\mathbf{p}) = B^\top \log(\mathbf{p})$$
$$\text{ILR}^{-1}(\mathbf{v}) = \mathcal{C}(\exp(B\mathbf{v})),$$

where $\mathcal{C}(\cdot)$ renormalizes and $B = [b_1, \ldots, b_{K-1}]$ is an orthonormal basis for the hyperplane of $\mathbb{R}^K$ orthogonal to $\mathbf{1}$. Thus, the block matrix $\begin{bmatrix} B & \mathbf{1}/\sqrt{K} \end{bmatrix}$ is orthogonal, so

$$\begin{bmatrix} B & \mathbf{1}/\sqrt{K} \end{bmatrix}^\top \begin{bmatrix} B & \mathbf{1}/\sqrt{K} \end{bmatrix} = B^\top B + \frac{\mathbf{1}\mathbf{1}^\top}{K} = I.$$ (3.33)

Similar to the LogOdds transformation, we will refer to a vector resulting from applying the ILR transformation to a probabilistic label as an “ILR vector”.

Like the LogOdds transformation, the ILR transform is a linear combination of vectors in $\mathbb{R}^{K-1}$ with weights $\log(p_i)$. In ILR, the vectors are the columns of $B^\top$, which are all equidistant from each other and from the origin (forming the vertices of a generalized tetrahedron), making ILR symmetric in all $K$ labels. An illustration of the difference between ALR and ILR is seen in Fig. 3.5. Many equally valid choices for $B$ exist due to rotational symmetry; in this thesis we use a basis suggested by Egozcue et al.:

$$b_i = \frac{1}{\sqrt{i + 1}} \begin{bmatrix} \frac{1}{\sqrt{i}}, \ldots, \frac{1}{\sqrt{i}}, -i, 0, \ldots, 0 \end{bmatrix}^\top_{i \text{ elements}} \begin{bmatrix} \sqrt{i} \end{bmatrix}^{(K-i-1) \text{ elements}}.$$ (3.34)
Figure 3.5: A demonstration of how the LogOdds and ILR transformations map three lines of equal length in \( \mathbb{P}_3 \) to \( \mathbb{R}^2 \). Under the ILR transformation all three lines are still of equal length, but the blue line is stretched under the LogOdds transformation. However, the LogOdds transformation has the useful property that the first two components of the probabilistic label map to the standard Euclidean axes.

The ILR transform is called “isometric” because it maps the Aitchison inner product to the Euclidean inner product:

\[
\langle \text{ILR}(p), \text{ILR}(q) \rangle = \langle \log(p) \rangle^\top B B^\top \log(q)
\]

\[
= \langle \log(p) \rangle^\top (I - \mathbf{1} \mathbf{1}^\top / K) \log(q) 
\]

\[
= \langle \log(p), \log(q) \rangle - \frac{(1, \log(p))(1, \log(q))}{K} 
\]

\[
= \langle \log(p), \log(q) \rangle - K \cdot \mu(\log(p))\mu(\log(q)) 
\]

Note that step (3.36) follows from (3.33), and is the reason that a similar derivation will not work for the LogOdds transformation. Similarly, the Aitchison distance and norm are also mapped to the standard Euclidean distance and norm, respectively. Based on Sec. 3.1.2, distances between ILR vectors can be interpreted as a measure of the “amount” of evidence that must be observed get to a certain posterior distribution from a prior distribution. Fig. 3.6 illustrates the equivalence of the Aitchison and Euclidean distances under the ILR transformation, and Fig. 3.7 demonstrates the intuitive results associated with PCA in ILR space.

We note that while each component of a LogOdds vector corresponds directly to one of the \((K-1)\) foreground label probabilities, the components of an ILR vector do not have such
Figure 3.6: The ILR transform and its inverse. (a) Orthogonal grid of lines, unit distance apart, in $\mathbb{R}^2$. (b) Correspondingly colored orthogonal compositional lines in $\mathbb{P}_3$ unit distance apart with respect to the Aitchison distance (see (3.18)).

A clear interpretation; changes in an ILR vector component will likely result in increasing several of the label probabilities. Thus, ILR vectors are less intuitive to manipulate directly, and tasks such as converting a set of SDMs to a probabilistic segmentation (see Sec. 3.2.1) are better suited for the LogOdds transformation. Further, in imaging applications where foreground-foreground boundaries are unlikely, statistical analysis results should be similar in LogOdds and ILR space, so the more intuitive LogOdds transformation may be the more appropriate choice.

The ILR transformation was suggested as being potentially useful in medical imaging analysis previously by Changizi and Hamarneh [66], but deeper analysis of how various imaging tasks will benefit from the use of the ILR transformation is still needed. In Sec. 3.2.3, we provide examples for both synthetic data and real medical images, comparing statistical analysis done in both LogOdds and ILR space. These examples highlight how the symmetry of the ILR transform leads to more intuitive results in some applications.

First, however, we establish a connection between the Aitchison geometry and information theory.
Figure 3.7: Performing PCA in ILR space results in a PCA mode (blue line) that captures the variability in the data one would intuitively expect. The same point set is used in Fig. 3.1 and Fig. 3.4.

**Relationship to Entropy**

The entropy of a probabilistic vector is a measure of the uncertainty associated with that vector. It is defined by Shannon [284] as the expected value of the information contained in a vector $\mathbf{q} \in \mathbb{P}_K$, specifically

$$H(\mathbf{q}) = -\sum_{k=1}^{K} q_k \log q_k ,$$

(3.40)

As in previous sections, we maintain the use of the natural log function. Entropy is important for the analysis of segmentation results as it characterizes the quality of the results in different areas; regions with high entropy are very uncertain and may require further user input to improve results, whereas in regions with low entropy, the user can be confident in the results [247, 274, 323, 332].

In this section, we establish a connection between the LogOdds and ILR transformations and the gradient of the entropy. The definition of the gradient depends on how the simplex

(a) Simplex Points

(b) ILR PCA
Figure 3.8: Quiver plots showing (a) $\nabla_1 \mathcal{H}$, (b) $\nabla_2 \mathcal{H}$, and (c) $\nabla_3 \mathcal{H}$ on $P_3$. In (a), $\nabla_1 \mathcal{H}$ at each point is split into its component contained in the simplex (blue) and its component perpendicular to the simplex (red), emphasizing how $\nabla_1 \mathcal{H}$ points out of the simplex. Note how $\nabla_2 \mathcal{H}$ in (b) is elongated along the third (vertical) component. $\nabla_3 \mathcal{H}$ is both contained in the simplex and symmetric across the components.

is parameterized. Taking the gradient with respect to the individual probabilities gives:

$$\nabla_1 \mathcal{H}(q) = \left[ \frac{\partial \mathcal{H}}{\partial q_1}, \ldots, \frac{\partial \mathcal{H}}{\partial q_K} \right]^\top$$

$$= [-1 - \log q_1, \ldots, -1 - \log q_K]^\top$$

$$= -(1 + \log q).$$

This gradient is defined in $\mathbb{R}^K$, and does not lie in the plane of the simplex since it does not sum to 0 (Fig. 3.8). Parameterizing the simplex by $K - 1$ variables will provide a gradient that does not point out of the simplex.

We can parameterize the simplex by the first $K - 1$ probabilities:

$$\mathcal{H}(q) = - \left( \sum_{i=1}^{K-1} q_i \log q_i \right) - \left( 1 - \sum_{i=1}^{K-1} q_i \right) \log \left( 1 - \sum_{i=1}^{K-1} q_i \right)$$

$$\nabla_2 \mathcal{H}(q) = \left[ \frac{\partial \mathcal{H}}{\partial q_1}, \ldots, \frac{\partial \mathcal{H}}{\partial q_{K-1}} \right]^\top$$

$$= - \left[ \log \left( \frac{q_1}{q_K} \right), \ldots, \log \left( \frac{q_{K-1}}{q_K} \right) \right]^\top$$

$$= - \text{LogOdds}(q).$$

This definition for the gradient is, like the LogOdds transformation, not symmetric in the components of $q$. 
A symmetric parameterization of the simplex is given by using the columns of $B$ as a basis (since each of the columns sum to 0). The corresponding gradient can be calculated by projecting $\nabla_{1} \mathcal{H}(q)$ from (3.43) onto the columns of $B$:

$$\nabla_{3} \mathcal{H}(q) = \left[ \langle \nabla_{1} \mathcal{H}(q), b_{1} \rangle, \ldots, \langle \nabla_{1} \mathcal{H}(q), b_{K-1} \rangle \right]^{T}$$

(3.48)

$$= - \left[ \langle 1 + \log q, b_{1} \rangle, \ldots, \langle 1 + \log q, b_{K-1} \rangle \right]^{T}$$

(3.49)

$$= - \left[ (1, b_{1}) + \langle \log q, b_{1} \rangle, \ldots, (1, u_{K-1}) + \langle \log q, u_{K-1} \rangle \right]^{T}$$

(3.50)

$$= - \left[ \log q, b_{1} \rangle, \ldots, \log q, b_{K-1} \rangle \right]^{T} \quad \text{since } \langle 1, b_{i} \rangle = 0$$

(3.51)

$$= \left[ \log q, b_{1} \rangle, \ldots, \log q, b_{K-1} \rangle \right]^{T}$$

(3.52)

$$= B^{T} \log(q)$$

(3.53)

$$= - \text{ILR}(q).$$

(3.54)

An illustration of the three gradient definitions is seen in Fig. 3.8. While we do not explore this relationship further in this thesis, it provides a connection between the Aitchison geometry and information theory that we hope to later build upon.

### 3.2.3 Shape Model Experiments

PCA is a commonly used method for constructing a shape model from existing segmentations. In this section, we build shape models by applying the LogOdds (i.e. the ALR) and ILR transformations to rigidly aligned training segmentations, using PCA to capture variability in the transformed segmentations, and then mapping the results back to the simplex of probabilistic labels. For brevity, we refer to the LogOdds transformation as the ALR transformation and we denote these shape models as ALR-PCA and ILR-PCA. For crisp training segmentations, we use the method outlined in Sec. 3.2.1, where vectors of SDMs of the crisp shapes are treated as ALR vectors and mapped to the simplex to create probabilistic segmentations. A minimum probability of 0.01 is imposed.

To construct a PCA shape model, a set of ALR or ILR transformed training segmentations are cast to vectors in $\mathbb{R}^{N(K-1)}$ and PCA is performed to find a mean segmentation $\bar{q}$, a set of eigenmodes of greatest variance $Q = [q^{1}, \cdots, q^{m}]$, and the variances associated with the modes, $\lambda_{1} \geq \cdots \geq \lambda_{m}$. Since the segmentations are roughly aligned, the PCA modes $Q$ should capture how the boundaries of shapes tend to deform across the training data.

Since the mean can be calculated using only vector addition and scalar multiplication, $\bar{q}$ is the same in ALR and ILR space. The modes, however, depend on distances, and thus
CHAPTER 3. STATISTICAL ANALYSIS OF PROBABILISTIC LABELS

Figure 3.9: A synthetic experiment detailing the differences between PCA in LogOdds and ILR space. (a) demonstrates how vectors in $P_3$ are stretched along the first label. (b) plots the mean error of the changes of the label probabilities (see (3.57)) when taking a small step along the eigenmode of greatest variance, $q_1$. (c) plots the CoV for $\{\lambda_2, \cdots, \lambda_{K-1}\}$. See text for further details.

will be different in the ALR- and ILR-PCA shape models. We note that if all of the modes are taken, the set of probabilistic segmentations in the span of the modes will be the same for ALR- and ILR-PCA, but the associated variances of the modes will differ. Since changes in the last component (the background) of a probabilistic label are magnified by a factor of approximately $\sqrt{K-1}$ when mapped by the ALR transformation, these changes affect the variance much more than changes in the other components, and thus we expect the ALR-PCA shape model to focus more on modeling changes in the background.

Synthetic Data

We begin with a simple synthetic example to demonstrate how the asymmetry in ALR can affect PCA results. We uniformly sample (with respect to the Euclidean distance) 10,000 random probabilistic labels from $P_K$, for each $K \in \{3, \cdots, 15\}$. We then take each sampled probabilistic label $p$ and skew it away from the center of the simplex by pushing the 1st label’s probability, $p_1$, away from 0.5 using a sigmoid function,

$$p_1 \leftarrow \left(1 + \exp(-30(p_1 - 0.5))\right)^{-1}, \quad (3.55)$$

then renormalizing. An example of this skew for $P_3$ is seen in Fig. 3.9a. We map the skewed labels using the ALR and ILR transformations and perform PCA on both to find $m = K - 1$ modes.
Since the probabilities are skewed along one label only, and are symmetric in the remaining labels, we expect that:

1. mode $q^1$ corresponds mostly to changes in the 1st label,

2. variances $\{\lambda_2, \cdots \lambda_{K-1}\}$ are all similar.

We design a test to quantify both of these expectations, and repeat these tests 100 times on different randomly generated data, with the results summarized in Fig. 3.9.

To quantify the first expectation, we calculate how the label probabilities change when taking a small step along $q^1$ away from the center of the simplex (the origin in ALR and ILR spaces). We denote the change in probability $p_i$ as $\delta_i$. If $q^1$ does indeed correspond mostly just to changes in $p_1$, we expect the other label probabilities to change minimally, by about $-\delta_1/(K-1)$ each, in order to maintain that the probabilities sum to one. That is, we expect

$$\frac{\delta_i}{\delta_1} = -\frac{1}{K-1}.$$  \hspace{1cm} (3.56)

Thus we define the “error” in the change of $p_i$, $i \geq 2$, as

$$E_i = \left| \frac{\delta_i}{\delta_1} - \frac{-1}{K-1} \right|.$$  \hspace{1cm} (3.57)

We summarize these errors for a given test by taking the mean of $\{E_2, \ldots, E_K\}$. Fig. 3.9b plots these mean errors. In ILR-PCA, the mean error is always close to 0, as expected. In ALR-PCA, the error increases for larger $K$, since $q^1$ focuses more on capturing the increased variance in the last label.

To quantify the second expectation, we take a normalized measure of dispersion known as the coefficient of variation (CoV) (the standard deviation divided by the mean) of $\{\lambda_2, \cdots \lambda_{K-1}\}$. We expect the CoV to be close to 0, indicating these variances are close in value. Fig. 3.9c plots the CoV for $\{\lambda_2, \cdots \lambda_{K-1}\}$. In ILR-PCA, the CoV is around 0, as expected. In ALR-PCA, the CoV reaches a peak around $K = 8$, where the ALR space variance of the 1st and last components are about the same, and both the first and second modes are required to explain this variance.

While PCA in ILR space gives the expected results, PCA in ALR space does not, especially for larger $K$. This results from changes in the last label being magnified in ALR space by $\sqrt{K-1}$ and the PCA modes capturing this variation.
CHAPTER 3. STATISTICAL ANALYSIS OF PROBABILISTIC LABELS

(a) Thigh Segmentation  
(b) LogOdds PCA Mode  
(c) ILR PCA Mode

Figure 3.10: An illustration of the PCA modes of greatest variation found using ALR transformed data (b) and ILR transformed data (c). (a) shows the segmentation of an axial slice of a thigh MRI. White corresponds to locations where the segmentation changes significantly when moving along the PCA mode (i.e. sum of the change in all probabilities when taking a small step away from the mean), and black corresponds to locations where the segmentation does not change. We see that PCA in ILR space captures variance in all of the boundaries, while PCA in ALR space focuses on variance in the background.

PCA Model Generalizability - Thigh Muscles

In this section we provide an example illustrating the advantages of using the ILR transform to model anatomical shapes in real medical data. We build anatomical shape models using PCA in both ALR and ILR space, and then examine how accurately these shape models generalize to unseen shapes. We use a data set consisting of 40 MRIs of thighs with $250 \times 250 \times 40$ voxels, each segmented into subcutaneous fat, intermuscular fat, cortical bone, bone marrow, 11 thigh muscles, and background [132], and rigidly aligned [8] (Fig. 3.10). More thorough experiments focused on the segmentation and registration of this data set are performed in subsequent chapters, and more details on its acquisition and background are available in Sec. 4.1.1. In this chapter this data set is used because it provides a good example of a practical multi-label segmentation problem, and because most of the boundaries are between adjacent anatomical regions, and not with the background, so the differences between ALR and ILR will be apparent. Since ALR-PCA gives extra weight to variations in the background, we expect the shapes of the muscle and bone labels to be modeled more accurately by the ILR-PCA shape model, since in ILR transformed data all label boundaries are treated equally. Conversely, we expect the shape of the background and the label it borders (the subcutaneous fat) to be more accurately modeled by the ALR-PCA
CHAPTER 3. STATISTICAL ANALYSIS OF PROBABILISTIC LABELS

<table>
<thead>
<tr>
<th>Type</th>
<th>$k$</th>
<th>LogOdds Proj. DSCs</th>
<th>ILR Proj. DSCs</th>
<th>$p$-Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Muscle</td>
<td>1</td>
<td>0.64 ± 0.15</td>
<td>0.71 ± 0.16</td>
<td><strong>0.0021</strong></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.73 ± 0.11</td>
<td>0.73 ± 0.13</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.76 ± 0.092</td>
<td>0.78 ± 0.14</td>
<td>0.30</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.72 ± 0.14</td>
<td>0.75 ± 0.14</td>
<td><strong>3.5 \times 10^{-5}</strong></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.79 ± 0.061</td>
<td>0.82 ± 0.040</td>
<td>0.027</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.76 ± 0.11</td>
<td>0.80 ± 0.10</td>
<td><strong>0.001</strong></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>0.83 ± 0.073</td>
<td>0.86 ± 0.068</td>
<td>0.0073</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0.57 ± 0.17</td>
<td>0.64 ± 0.12</td>
<td>0.033</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0.83 ± 0.051</td>
<td>0.84 ± 0.058</td>
<td>0.027</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.71 ± 0.10</td>
<td>0.72 ± 0.093</td>
<td>0.13</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>0.85 ± 0.030</td>
<td>0.88 ± 0.022</td>
<td><strong>1.0 \times 10^{-5}</strong></td>
</tr>
<tr>
<td>Bone</td>
<td>12</td>
<td>0.57 ± 0.21</td>
<td>0.61 ± 0.17</td>
<td>0.34</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>0.51 ± 0.26</td>
<td>0.54 ± 0.22</td>
<td>0.49</td>
</tr>
<tr>
<td>Fat</td>
<td>14</td>
<td>0.36 ± 0.12</td>
<td>0.41 ± 0.13</td>
<td><strong>0.00066</strong></td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.80 ± 0.10</td>
<td>0.79 ± 0.12</td>
<td>0.23</td>
</tr>
<tr>
<td>Background</td>
<td>16</td>
<td>0.9794 ± 0.0080</td>
<td>0.9761 ± 0.0094</td>
<td><strong>0.00012</strong></td>
</tr>
</tbody>
</table>

Table 3.1: The mean and standard deviation of the DSC values calculated between the 20 testing segmentations and their corresponding projections onto the ALR- and ILR-PCA shape spaces, built from the 20 training images. Also shown are the $p$-values generated from a paired $t$-test to determine if the differences in DSC values are statistically significant. Bolded $p$-values pass the Holm-Bonferroni test (used to correct for multiple comparisons) at a significance level of 0.05. We note that ILR-PCA performs better when modeling the labels that do not share a boundary with the background ($K \in \{1, \ldots, 14\}$, see Fig. 3.10).

We test the ability of the ALR- and ILR-PCA shape models to generalize to new images. We performed PCA on a training set of 20 of the images (chosen at random) to construct a shape model. $m$ PCA modes were found, where $m$ was chosen to account for at least 90% of the variability in the training data (11 modes for both ALR- and ILR-PCA). The remaining 20 segmentations were used to test the generalizability of the shape models. Each testing segmentation was projected onto the span of the PCA modes. The projected segmentations represent the closest approximation to the original testing segmentations in the PCA shape space. For a shape model mean $\bar{q}$, modes of variation $Q$, and variances $\Lambda$, if $q_{gt}$ is the ground truth segmentation, its projection onto the shape space $q_{proj}$ is given by

$$q_{proj}(x) = LR^{-1}(\bar{q} + Q(x)Q(x)^\top (LR(q_{gt}(x)) - \bar{q}(x))),$$

(3.58)
We evaluated the generalizability of the shape models by comparing the original segmentations to their projections. We compare them on a label by label basis using the extension of the Dice similarity coefficient (DSC) to probabilistic labels from (2.18). We use a probabilistic DSC instead of thresholding the segmentations so as not to lose uncertainty information. The DSCs for each label are summarized by their mean and standard deviation in Table 3.1, along with the $p$-values generated from a paired $t$-test to test the null hypothesis that the difference in DSC between ALR- and ILR-PCA has an expected value of zero. We see the background label is more accurately modeled by ALR-PCA, but ILR-PCA models almost all of the foreground labels more accurately. The subcutaneous fat (label 15) is modeled equally well by both shape ALR- and ILR-PCA since half of its boundary is with the background and half with the other foreground labels.

### 3.3 Chapter Summary and Discussion

In this chapter, we introduced techniques for mapping probabilistic labels to a vector space in order to facilitate algebraic manipulation and statistical analysis. A well known method for performing this mapping is the LogOdds (ALR) transformation, which has several useful properties. However, the LogOdds transformation is asymmetric when there are multiple foreground labels, and changes in the background probability cause a greater change in the LogOdds transformed data than equivalent changes in the foreground probabilities. Making use of established methods for compositional data analysis, we demonstrated that using the symmetric ILR transformation to map probabilistic labels more accurately captures foreground-foreground boundaries in statistical shape models. Thus, the ILR transformation is a more appropriate mapping in applications containing many foreground-foreground boundaries.

We note that an alternative approach to performing statistical analysis on probabilistic labels is to employ a non-Euclidean simplicial metric (e.g. the Fisher information metric or the Bhattacharyya metric) and use an extension of PCA to Riemannian manifolds (e.g. principal geodesic analysis [109]). In the future, we will explore the relationship between these metrics and the Aitchison metric in the context of shape analysis, building upon the connection we established between entropy and the ILR transformation in (3.52). We note, however, several advantages the Aitchison metric provides that make it a strong choice when
performing PCA on probabilistic segmentations:

1. It is isometric to the Euclidean metric through the ILR transform, simplifying the implementation of PCA.

2. The connections between the Aitchison metric and Bayesian inference, highlighted in Sec. 3.1, motivate the use of the Aitchison geometry.

3. Many other simplicial metrics are compact, and would require projections or constraints to ensure probabilities stay in the simplex.
Chapter 4

Fully Automated Segmentation

In the previous chapter, we discussed techniques for performing statistical analysis on probabilistic labels and demonstrated the differences in segmentation results when using the LogOdds and ILR transformations, showing how the bias in the LogOdds transformation affects statistical shape models. In this chapter, we incorporate these transformations into a fully automatic segmentation framework. Of course, every segmentation framework requires some user input (e.g. in order to choose energy terms, set parameters). By fully automatic, we refer to the algorithm running without intervention each time a novel image of a certain class (e.g. a thigh MRI) is acquired. Our framework fills a niche in that it is, to our knowledge, the first segmentation method to satisfy four important properties for fully automatic segmentation (discussed in more detail in Sec. 2.2):

1. Initialization independence (convexity);
2. Enforcing an anatomical shape model;
3. Simultaneous multi-label segmentation;
4. Statistically meaningful uncertainty information.

The use of the log-ratio segmentation representations are key to satisfying these properties, specifically in terms of encoding meaningful uncertainty information in a multi-label shape model.

In order to highlight all of our contributions and give context to the benefits of the above four properties, we focus primarily on the difficult task of thigh muscle segmentation from volumetric MRI, which we discuss in detail in the following section.
4.1 Application: Thigh Muscle Segmentation

4.1.1 COPD Data Set

In patients with chronic obstructive pulmonary disease (COPD), skeletal muscle weakness is common [192, 277, 281, 283, 337]. Lower limb muscles are often the most affected due to lack of use [283]. The skeletal muscle weakness related to COPD may be associated with the loss of muscle mass [283]. Reduced muscle mass is among several factors that lead to reduced force production, also including changes in the muscle contractile apparatus and neuromuscular activation [277]. The relative effects of these factors are an ongoing area of study [192]. Thus, size and shape measurements are required to study the contribution of muscle mass reduction to force loss. Recently [132], a non-uniform distribution of atrophy and size changes was found across knee extensors and flexors in patients with COPD, which may be reflective of localized factors such as denervation, limited recruitment, or atrophy of specific muscle fibers, rather than systemic factors contributing to muscle atrophy. Further, classifiers trained on 3D shape descriptors of thigh muscles have been able to detect COPD [204].

MRI can be used to distinguish muscle from the surrounding regions, and can generate multiple image slices from which volume and shape properties of individual muscles can be estimated [102]. Information regarding the specific muscles or muscle regions that are most atrophied together with functional assessment will enable therapeutic interventions to be targeted to the affected regions [132, 204, 325] rather than the prescription of a generalized approach that may prove ineffectual.

A sample of patients was obtained with moderate to severe COPD on the basis of the Global Initiative for Chronic Obstructive Lung Disease guidelines [213] and of greater than 50 years of age. The healthy control group were of age greater than 50 years, free of lung disease and non-smokers.

A 1.5T MRI scanner (1.5T Horizon Echospeed Scanner; GE Healthcare, Milwaukee, WI) was used to acquire 5 mm-thick, contiguous, axial slices from the anterior superior iliac spine to the tibial plateau while the subjects lower extremities were strapped to a foam block to minimize movement. Images were T1-weighted magnetic resonance (echo time, 8 ms; repetition time, 650 ms) with a 40 cm$^2$ field of view and a 512×384 pixel matrix (in-plane resolution, 0.78 × 0.78 mm). For each subject, two sets of images, one for the upper and another for the lower thigh regions, were collected in immediate succession without a change in the subjects position. A landmark (vitamin E capsule) at the mid-thigh (half the distance
between the anterior inferior iliac spine and the superior margin of patella) was identified on the MRI scans to register the two images into a single image of the entire thigh. The vitamin E landmark facilitated the identification of overlap between the two sets of images. The MRI scan yielded a total of approximately 100 slices for each participant, which were merged into a single 3D image using the Merge module in the Amira 3.1 software package (Mercury Computer Systems, Inc, Chelmsford, MA), with the coincidence of the landmark verified visually after merging. ITK-SNAP 1.6.0.1 [355] was used for the manual slice-by-slice segmentation of individual muscles of the left thigh from the merged axial MRIs. Segmentation was performed by a physical therapist with expert anatomic knowledge. The images were semi-automatically cropped down to \(250 \times 250 \times 40\), so that only the left thigh was contained, and the slices started just above the knee and extended the upper thigh, so that the muscles of interest were all almost entirely contained. The images are normalized to take values in the range \([0, 1]\).

### 4.1.2 Thigh Muscle Segmentation Overview

An important precursor to any volume measurement or shape analysis is segmentation. Manual segmentation in 3D medical images is extremely time consuming, tedious, and suffers from inter- and intra-operator variability. Highly automated segmentation is important for studies involving a large cohort of subjects to reduce manual labour and variability, and to improve the efficiency in analyzing large groups of data. Encoding uncertainty in automated segmentations through the use of probabilistic labels aids in subsequent user analysis and identification of segmentation errors [273,275].

There are three main issues that complicate the task of knee extensor and flexor (thigh) muscle segmentation. The first issue is that each of the thigh muscles has similar intensity and texture, so they cannot be differentiated using local image features alone. The second issue is that the boundaries of intermuscular fat between neighboring muscles often have thickness on the order of the imaging resolution, and tend to be obscured in many areas by imaging noise and partial volume effects. Lastly, the texture and intra-muscular fat in the muscles can create intensity gradients greater than nearby intermuscular boundaries (Fig. 4.1). Certain imaging procedures can partially alleviate these issues [26], but as in our case, such images are not always available.

When manually segmenting an image, a clinician often must rely on their prior anatomical knowledge of the relative muscle locations in order to distinguish them; automated
segmentation methods must somehow encode similar anatomical information to achieve adequate accuracy. A diverse set of techniques for encoding this information has been proposed in the muscle segmentation literature. Gilles et al. deformably register template meshes of muscles and other thigh anatomy to a novel image, matching model edges to boundaries in the images and employing regularization terms designed to model plausible deformations in real anatomy \cite{118,119}. Essafi et al. propose a wavelet-based encoding for calf muscles represented using landmarks, which provides a hierarchical encoding of shape variability \cite{104}. Wang et al. capture muscle shape variability using the relative positions of small cliques of nearby landmark points, and segment novel images by training random forest classifiers for the landmark locations based on image features \cite{341}. The above methods rely on matching surface models to muscle boundaries and can suffer from difficult to detect or false boundaries that often occur in thigh MRI data. Further, none of these methods guarantee a globally optimal solution, and thus can be sensitive to improper initialization.

Baudin et al. employ several techniques for implicit segmentation of thigh muscles that achieve excellent results. First, they present a technique to automatically detect voxels inside the various muscles utilizing a boundary map calculated from local intensity variance, and use these voxels as “seeds” to guide a subsequent segmentation \cite{23}. However, the detection of enough boundaries to ensure seeds lie in the interior of muscle regions may be difficult. Baudin et al. also present a technique for encoding shape models learnt from training segmentations into a random walker (RW) segmentation framework \cite{124}, which benefits from RW’s robustness to missing edges \cite{24,25}. These works exhibit the greatest similarity

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig4_1.png}
\caption{An example axial slice of a thigh MRI and the corresponding labeled ground truth.}
\end{figure}
to our proposed approach, as the segmentation step is convex and it provides a probabilistic solution. However, our data exhibits many false edges from intramuscular fat (Fig. 4.10, Fig. 4.23), which can cause errors in RW based approaches due to their dependence on local gradient information for regularization. Also, they employ a deformable registration step prior to the construction of their shape models. Deformable registration is difficult to perform optimally, and may remove some of the shape variability that we wish to capture in order to differentiate COPD and healthy muscle structure.

We build a fully automatic thigh segmentation technique based on the four properties discussed above (a convex, multi-label segmentation framework incorporating a shape model and uncertainty information). We rely on multi-label shape models to achieve adequate segmentation accuracy and robustness in the presence of low quality image data. Our technique incorporates three further contributions specifically designed for thigh muscle segmentation:

1. We train a statistical shape model capturing the relative thigh muscle shapes and locations using a novel generalized log-ratio (GLR) transformation for segmentation representation. We show that we can improve segmentation accuracy by implicitly encoding anatomical volume and adjacency information in our segmentation representation.

2. We introduce a technique for automatically presegmenting images into bone, fat, and muscle classes, with all knee extensor and flexor muscles combined under one label. We use this presegmentation to perform an anatomical alignment of the images, removing pose and size variance so only the variance in higher order shape moments remains.

3. We train a rotationally invariant random forest classifier to robustly detect likely boundary locations in the images and the expected orientation of these boundaries. An anisotropic smoothing of the boundary likelihoods using this orientation reinforces accurately detected boundary voxels and suppresses erroneous boundary voxels.

Information from the presegmentation, the shape model, and the boundary likelihoods are incorporated into a strictly convex energy functional that is globally minimized to produce a probabilistic multi-label segmentation.

In the following section, we introduce the GLR transformation and its use as a segmentation representation.
4.2 Segmentation Representation

In the previous chapter, we explored the uses of the LogOdds and ILR transformations for the statistical analysis of probabilistic segmentations. We presented theory and results showing that statistical analysis using the LogOdds transformation gives greater weight to changes in the background label, and so is appropriate in applications with few foreground-foreground boundaries, while the ILR transformation treats all labels equally, and thus is an appropriate choice when lacking prior knowledge about the segmentation labels. However, in applications such as thigh muscle segmentation, with many neighboring structures of different shapes and volumes, we may not want all boundaries to be treated equally. In this section, we generalize the LogOdds and ILR transformations, and demonstrate how we can design transformations that enforce specified distances between certain labels (Fig. 4.2). These transformations can encode anatomical volume and adjacency information learned from training data, and can influence how much different labels influence a PCA shape model and how strictly transitions between pairs of labels are penalized when regularizing the segmentation.

4.2.1 Generalized Log-Ratio Transformation

We introduce a family of transformations that generalize the LogOdds and the ILR transformations, known as generalized log-ratio (GLR) transformations. A GLR transformation of the unit simplex is a function GLR : \( \mathbb{P}_K \to \mathbb{R}^{K-1} \) given by

\[
\text{GLR}(\mathbf{q}) = A \log(\mathbf{q})
\]

\[
A \cdot \mathbf{1} = \mathbf{0},
\]

where \( \mathbf{q} \in \mathbb{P}_K \). \( A = [a_1, \ldots, a_K] \) is a \((K-1) \times K\) matrix, \( \mathbf{1} \) and \( \mathbf{0} \) are vectors of all 1’s and 0’s, respectively. Setting \( A = [I, -1] \) gives the LogOdds transformation (see (3.29)), and setting \( A = B^T \) gives the ILR transformation (see (3.31)). The condition (4.2) on \( A \) ensures that vector addition and scalar multiplication correspond to perturbation and power.
Figure 4.2: An example of the columns of $A$ for the LogOdds, ILR, and GLR transformations for $K = 3$ labels. For the GLR transformation, the columns are chosen to satisfy a predefined pairwise distance function $D$.

transformation, respectively. That is, for $p, q \in \mathbb{P}_K$ and $\alpha \in \mathbb{R}$,

$$\text{GLR}(p \oplus q) = \text{GLR}(C([p_1q_1, \ldots, p_Kq_K]^\top))$$

$$= A \log \left( \frac{1}{Z} [p_1q_1, \ldots, p_Kq_K]^\top \right)$$

$$= A(\log(p_1) + \log(q_1), \ldots, \log(p_K) + \log(q_K))^\top - \log(Z)A1$$

$$= A \log(p) + A \log(q) - 0, \text{ from (4.2)}$$

$$= \text{GLR}(p) + \text{GLR}(q), \quad (4.7)$$

$$\text{GLR}(\alpha \odot q) = \text{GLR} \left( C \left( [q_1^\alpha, \ldots, q_K^\alpha]^\top \right) \right)$$

$$= A \log \left( \frac{1}{Z} [q_1^\alpha, \ldots, q_K^\alpha]^\top \right)$$

$$= A(\alpha[0, \ldots, 0]) - \log(Z)A1$$

$$= \alpha A \log(q) - 0, \text{ from (4.2)}$$

$$= \alpha \text{GLR}(q). \quad (4.12)$$

$Z$ is the normalizing scalar associated with $C(\cdot)$ that ensures a vector sums to 1.

The condition (4.2) also results in a simple form for the inverse transformation, given by

$$\text{GLR}^{-1}(\phi) = C \left( \exp \left( A^+ \phi \right) \right).$$

To show this, from (4.2) we know $A$ has 0 as an eigenvalue with corresponding right eigenvector $1$. We assume $A$ has no other zero eigenvalues, i.e. its columns are all linearly
independent. If $A^+$ is the pseudo-inverse of $A$, then

$$A^+A = I - \frac{11^\top}{K}, \quad (4.14)$$

where $I$ is the identity matrix. This allows us to define the inverse of a GLR transformation:

$$\text{GLR}^{-1}(\text{GLR}(q)) = C(\exp(A^+ \text{GLR}(q))) \quad (4.15)$$

$$= C(\exp(A^+ A \log(q))) \quad (4.16)$$

$$= C\left(\exp\left(\left(I - \frac{11^\top}{K}\right) \log(q)\right)\right) \quad (4.17)$$

$$= C\begin{pmatrix}
\exp(\log(q_1)) / \exp(1^\top \log(q)/K) \\
\vdots \\
\exp(\log(q_K)) / \exp(1^\top \log(q)/K)
\end{pmatrix} \quad (4.18)$$

$$= C\begin{pmatrix}
\frac{q_1}{\beta} \\
\vdots \\
\frac{q_K}{\beta}
\end{pmatrix}, \quad \text{where } \beta = \exp\left(\frac{1^\top \log(q)}{K}\right) \quad (4.19)$$

$$= q, \quad \text{since } C(\cdot) \text{ removes scaling factors.} \quad (4.20)$$

### 4.2.2 Label Transition Penalties

We now demonstrate how the matrix $A$ can be chosen to encode prior anatomical volume and adjacency information directly into a segmentation representation.

A GLR transformation is a weighted sum of the columns of $A$, with weights given by the log of the probabilities, so there is a correspondence between the columns $A$ and the probabilistic labels. To make this correspondence clearer, let $r^i = C(e^i + \epsilon \cdot 1) \in \mathbb{P}_K$ for $i \in \{1, \ldots, K\}$, where $e^i$ is the $i^{th}$ standard basis vector and $\epsilon$ is a small scalar, so $r^i$ represents a probabilistic labeling where label $i$ is very likely. Taking advantage of the fact
that \( A \cdot 1 = 0 \) and letting \( \delta_{i,j} \) be the Kronecker delta, we see that

\[
\text{GLR}(r^i) = A \log \left( C \left( e^i + \epsilon \cdot 1 \right) \right) = A \log \left( \frac{1}{Z} \left( \frac{1}{\epsilon} \cdot e^i + 1 \right) \right) = A \log \left( \frac{1}{\epsilon} \cdot e^i + 1 \right) - A \cdot (\log(\epsilon) \cdot 1) = A \log \left( \frac{1}{\epsilon} \cdot e^i + 1 \right) \]

\[
= \left[ a_1, \ldots, a^K \right] \begin{bmatrix}
\log(\delta_{1,i}/\epsilon + 1) \\
\log(\delta_{K,i}/\epsilon + 1)
\end{bmatrix}
= \sum_{k=1}^{K} \log(\delta_{k,i}/\epsilon + 1) a^k
= \log \left( \frac{1 + \epsilon}{\epsilon} \right) a^i
\approx \log \left( \frac{1}{\epsilon} \right) a^i.
\]

Thus, the magnitude of \( \text{GLR}(r^i) \) is proportional to \( \|a^i\| \), the Euclidean norm of \( a_i \), and the distance between \( \text{GLR}(r^i) \) and \( \text{GLR}(r^j) \) is proportional to \( \|a^i - a^j\| \). By adjusting the value of \( \|a^i - a^j\| \), we can encode pairwise label distances into GLR transformations that determine how much the transformation is affected by changes from “likely label \( i \)” to “likely label \( j \)”. There are two ways that this pairwise distance will affect our segmentation algorithms:

- To regularize a segmentation, we penalize large gradient values in the segmentation’s components (see Sec. 4.3.3 for more details), and the gradient values between an image region that is “likely label \( i \)” and an image region that is “likely label \( j \)” will be proportional to \( \|a^i - a^j\| \). Increasing (decreasing) \( \|a^i - a^j\| \) will deter (encourage) boundaries between labels \( i \) and \( j \) in the segmentation. This can ensure, for example, that a thin label is not neglected by discouraging its neighboring labels from sharing a boundary (Fig. 4.3).

- Increasing \( \|a^i - a^j\| \) will result in greater variance in GLR space near boundaries between labels \( i \) and \( j \), and thus these boundaries will have a greater influence in PCA.
When performing PCA on training segmentations, if a pixel $x$ is near the boundary between labels $i$ and $j$, training segmentations may assign $x$ high probability for either label $i$ or $j$. A greater distance between the $x$'s training probabilities in GLR space results in a greater variance being detected at $x$, so increasing $\|a^i - a^j\|$ results in $x$ having a greater influence on the PCA modes. The resulting PCA shape model will capture variations in the boundary between labels $i$ and $j$ more accurately.

In order to better capture the shapes of smaller structures, we want to choose a distance function between labels that encodes the relative volumes and adjacencies of the structures. At a high level, the idea is to construct a distance function between labels that varies inversely with the surface area of the shared boundaries of the structures corresponding to those labels. Smaller structures will have smaller boundaries, and thus greater distances and increased influence on a PCA shape model. From the training segmentations, we calculate the average number of voxels in the boundary between labels $i$ and $j$ (approximately the average surface area), $\sigma_{i,j}$ (with $\sigma_{i,i} = 0$). We define $\sigma$ as the matrix of all surface areas and $\bar{\sigma}$ as the mean across all pairs of labels:

$$\bar{\sigma} = \frac{2}{K \cdot (K-1)} \sum_{i=1}^{K} \sum_{j=i+1}^{K} \sigma_{i,j}.$$  \hspace{1cm} (4.29)

We want to define a squared-distance matrix $D$ with components varying inversely to the corresponding components of $\sigma$; for example we use

$$D_{i,j} = \frac{\bar{\sigma}}{\sigma_{i,j} + \bar{\sigma}} \text{ if } i \neq j, \quad D_{i,i} = 0.$$  \hspace{1cm} (4.30)

The smaller the boundary shared by a pair of labels, the greater the distance between them will be under the GLR transformation, discouraging transitions between them in the segmentation (Fig. 4.3).

Given such a $K \times K$ matrix $D$ of pairwise squared distances between labels, we would like to choose $A$ such that $\|a^i - a^j\|^2 = D_{i,j}$. It is known that $D$ encodes a Euclidean distance if and only if $T = -\frac{1}{2}CDC$ is positive semidefinite, where $C = I - \frac{1}{K}11^\top$ is a centering matrix [175]. If $T$ is positive semidefinite, then the desired $A$ is given by $T = A^\top A$. The distance encoded in $D$ from (4.30) may not be Euclidean, in which case $T$ will have negative eigenvalues. Let $\psi_1$ be the smallest eigenvalue of $T$. If $\psi_1 < 0$, we add $(-2\psi_1)$ to the off diagonal components of $D$ to get $\hat{D} = D - 2\psi_1(11^\top - I)$. $\hat{D}$ corresponds to a Euclidean
Figure 4.3: An illustration of how the GLR transformation can be used to ensure small or thin structures are properly segmented. An image region may be segmented correctly into 3 regions (a) or incorrectly into 2 regions (b). This example assumes a regularization term penalizing the total change in the segmentation. Using no prior adjacency information such as the ILR transformation may result in the middle region being skipped, as the segmentation incurs too large of a regularization penalty (c). If it is known from the training data that regions 1 and 3 do not share boundaries with each other, but rather with the thin region 2, the GLR transformation can be constructed so that transitioning from region 1 to region 3 through region 2 does not incur any additional cost compared to skipping region 2 (d). Such situations occur in thigh muscle segmentation (e).
distance, which we show by proving \( \hat{T} = -\frac{1}{2} \hat{D} \hat{C} \) is positive semidefinite. First, we establish several properties of \( C \):

\[
C1 = 1 - \frac{1}{K} \mathbf{1}\mathbf{1}^\top \\
= 1 - \frac{K}{K} \mathbf{1} \\
= 0 ,
\]

\[
CC = I - \frac{2}{K} \mathbf{1}\mathbf{1}^\top + \frac{1}{K^2} \mathbf{1}\mathbf{1}^\top \mathbf{1}\mathbf{1}^\top \\
= I - \frac{2}{K} \mathbf{1}\mathbf{1}^\top + \frac{K}{K^2} \mathbf{1}\mathbf{1}^\top \\
= C .
\]

Now, since \( C1 = 0, \mathbf{1} \) is a zero eigenvalue of \( T \) corresponding to the eigenvector \( \mathbf{1} \). Let \( T = Q\Psi Q^\top \) be the eigenvector decomposition of \( T \), with \( Q \) a matrix of orthonormal eigenvectors and \( \Psi \) a diagonal matrix of eigenvalues. Then

\[
\hat{T} = -\frac{1}{2} C(\hat{D})C \\
= -\frac{1}{2} (CDC - 2\psi_1 C\mathbf{1}\mathbf{1}^\top C - 2\psi_1 CC) \\
= -\frac{1}{2} (CDC + 2\psi_1 C) \\
= Q\Psi Q^\top - \psi_1 \mathbf{1} + \frac{\psi_1}{K} \mathbf{1}\mathbf{1}^\top \\
= Q(\Psi - \psi_1 \mathbf{1})Q^\top .
\]

\( Q(\Psi - \psi_1 \mathbf{1})Q^\top \) is an eigenvector decomposition of \( \hat{T} \), and since all of the diagonal components of \( (\Psi - \psi_1 \mathbf{1}) \) are greater than or equal to zero (the smallest component of \( \Psi \) is \( \psi_1 \)), \( \hat{T} \) is positive semi-definite. We use this technique because only the relative values of the distances in (4.30) are meaningful (i.e. which labels are closer and which are farther), and adding a constant preserves this relative ordering (e.g. \( D_{i,j} > D_{k,l} \) implies \( \hat{D}_{i,j} > \hat{D}_{k,l} \)). Algorithm 1 provides an overview of this technique for constructing a GLR transformation.

In the following sections, we use the definition in (4.30) for the GLR transformation in our thigh muscle segmentation framework. Results on the thigh muscle data set demonstrating the benefits of representing a segmentation using an application dependent GLR transformation are presented in Sec. 4.4.2. We note that (4.30) is one of many ways of constructing a GLR transformation, and while this technique is well suited for segmentation
Algorithm 1 Choosing the Matrix $A$ for GLR Based on Boundary Surface Areas

1: Input: Surface Areas $\sigma$
2: Compute $D$ according to (4.30)
3: Compute $T = \frac{1}{2}CDC$
4: Calculate the eigendecomposition $T = Q\Psi Q^\top$, $\Psi = \text{diag}([\psi_1, \ldots, \psi_{K-1}]^\top)$, $\psi_1 \leq \ldots \leq \psi_K$
5: If $\psi_1 \geq 0$
6: \hspace{1em} Output: $A = \Psi^{1/2}Q^\top$
7: Else
8: \hspace{1em} Output: $A = (\Psi - \psi_1 I)^{1/2}Q^\top$ (see (4.41))

when training data is available, other applications may benefit from a differently constructed GLR transformation.

4.3 Energy Minimization Segmentation Framework

We formulate our segmentation algorithm as an energy minimization problem. We represent a $K$ label segmentation as a function $\phi : \Omega \to \mathbb{R}^{K-1}$, where $\phi(x)$ is a GLR vector. Since $\Omega$ is a discrete set of $N$ pixels, segmentations can be interpreted as vectors in $\mathbb{R}^{N(K-1)}$, useful when performing PCA. We define our energy functional as a weighted sum of terms:

$$E(\phi) = E_{\text{rgn}}(\phi) + w_1 E_{\text{shp}}(\phi) + w_2 E_{\text{bnd}}(\phi),$$  \hspace{1em} (4.42)

where $w_1$ and $w_2$ are scalar weights controlling the trade-off between terms. The setting of the weights is largely application dependent and is discussed in more detail later. The first term incorporates local regional image information into the segmentation by performing a presegmentation, as described in Sec. 4.3.1. The second term, incorporating the information from a PCA shape model, is described in Sec. 4.3.2. The third term forces the segmentation boundaries to conform to the intermuscular boundaries and is described in Sec. 4.3.3.

We ensure robustness to initialization by defining each energy term to be convex, allowing $E(\cdot)$ to be globally minimized. It is important to note that a convex energy functional may still have multiple global optima; if one of the terms is strictly convex, however, a single global optimum is guaranteed.
4.3.1 Regional Energy Terms

The high level approach to defining a convex regional energy term is to define a family of distributions over the segmentation labels, parameterized by local image features. A probabilistic segmentation can be derived from this family of distributions, which can then be transformed into GLR space, denoted \( \pi : \Omega \rightarrow \mathbb{R}^{K-1} \). We can then utilize the Aitchison geometry to define a convex energy term measuring the similarity between \( \phi \) and \( \pi \), for example:

\[
E_{\text{rgn-1}}(\phi) = \sum_{x \in \Omega} -\langle \phi(x), \pi(x) \rangle \\
E_{\text{rgn-2}}(\phi) = \sum_{x \in \Omega} \| \phi(x) - \pi(x) \|^2 ,
\]

where \( \langle \cdot, \cdot \rangle \) is the standard Euclidean inner product and \( \| \cdot \| \) is the standard Euclidean norm. The term used would depend on the application.

Thigh Presegmentation

While it is difficult to distinguish thigh muscles from each other, it is much easier to distinguish muscle tissue from other anatomical structures, even by intensity alone, as seen in Fig. 4.4a. We take advantage of this to perform a presegmentation and identify which voxels are muscle (regardless of which particular muscle) to guide the segmentation. While we are only eventually interested in segmenting the muscles, it is useful to segment the fat tissue
CHAPTER 4. FULLY AUTOMATED SEGMENTATION

Figure 4.5: Regions in the thigh, and the average intensity profiles of the thighs. We see the background and cortical bone are dark, the muscle is medium, and the fat and bone marrow are light.

and the femur bone for the purpose of anatomical alignment (detailed in Sec. 4.3.2). Since the presegmentation does not require a shape model, it can be used to align the training images prior to shape model construction.

To perform the presegmentation, we note that the thigh images exhibit three main intensity classes (Fig. 4.4a, Fig. 4.5), “dark” (background, cortical bone), “gray” or “medium” (muscle), and “light” (fat, bone marrow). We assign a probability for each of these classes to each voxel by fitting a Gaussian mixture model (GMM) to the voxel intensities. Since we know roughly the means and variances of the intensities of these three classes from training data, we can provide a good initialization to the GMM, and because the three classes are so distinct, we have found the GMM to converge to the same distribution when the initialization is perturbed. The result is a length 3 probability vector, mapped to $\mathbb{R}^2$ using the ILR transformation, assigned to each voxel, $\pi_{\text{GMM}} : \Omega \rightarrow \mathbb{R}^2$. Note we do not have strong prior knowledge to incorporate into a GLR transformation, so we use the ILR transformation.

To account for artifacts in the presegmentation due to noise and PVE (Fig. 4.4b), we incorporate a regularization step:

$$
\pi_{\text{opt}} = \arg\min_{\pi} E_{\text{preseg}}(\pi)
$$

$$
E_{\text{preseg}}(\pi) = \sum_{x \in \Omega} \| \pi(x) - \pi_{\text{GMM}}(x) \|^2 + w_3 \sum_{x \in \Omega} \sum_{i=1}^{2} \| \nabla \pi_i(x) \|,
$$
where $w_3$ is a scalar weight and $\nabla$ is the spatial gradient operator. The second term in (4.46) is a standard total variation (TV) term on the components of $\phi$, which can be globally optimized using primal-dual techniques [62] (discussed further in Sec. 4.3.3) and creates a smooth presegmentation (Fig. 4.4c).

Identifying the femur bone is useful both for the alignment of images and to provide anatomical context to the shape model described in Sec. 4.3.2. To extract the bone from the presegmentation, we use the following post processing technique, which we have empirically found to give consistently accurate results:

1. Convert $\pi_{opt}$ to $\mathbb{P}_3$ and threshold to create a crisp (non-probabilistic) segmentation;
2. Find the convex hull of the “gray” (muscle) label;
3. Label the largest connected component of the “dark” region inside this convex hull as cortical bone;
4. Find the “light” region voxels inside the cortical bone region and label it as bone marrow.

Following this procedure, we have an estimate of the location of muscle tissue (“gray” regions), bone, and fat (the remaining “light” regions, see Fig. 4.4c). Incorporating bone and fat into the segmentation allows us to encode which muscles border bone and which border fat in the GLR transformation.

Using the probabilities from $\pi_{opt}$ and the bone and fat locations found using the above procedure, we construct a 4 label probabilistic segmentation (muscle, bone, fat, and background). We convert this to a 14 label segmentation by “spreading” the muscle probability at each voxel evenly among all 11 muscles labels, and apply the GLR transformation to the resulting probabilities, denoting the result $\pi_{\text{rgn}}$. For the remainder of the chapter, we fix the number of labels as $K = 14$ when discussing thigh muscle segmentation.

We define the regional energy term using (4.43):

$$E_{\text{rgn}}(\phi) = -\sum_{x \in \Omega} \langle \phi(x), \pi_{\text{rgn}}(x) \rangle.$$  \hspace{1cm} (4.47)

We use an inner product instead of a distance term (as in (4.46)) since we do not want $\phi(x)$ to be close to $\pi_{\text{rgn}}(x)$ (which has a maximum muscle probability of 1/11) but to only follow its guidelines as to what voxels are and are not muscle.
4.3.2 Shape Energy Terms

We use shape energy terms based on PCA shape models, as described in Chapter 3. To review, we assume we have a set of \( M \) manually generated training segmentations. Since manual segmentations are usually crisp, we convert them to probabilistic segmentations using the technique introduced by Pohl et al., calculating the SDMs of the foreground labels, treating the SDM values as a LogOdds vector, and converting the vector back to the simplex of probabilistic labels [241]. We then apply an application dependent GLR transformation to each of the training segmentations to get a set of training segmentations in GLR space, \( \{ \eta^1, \ldots, \eta^M \} \), \( \eta^i \in \mathbb{R}^{N \cdot (K-1)} \). We use PCA to extract the mean \( \bar{\eta} \in \mathbb{R}^{N \cdot (K-1)} \), the \((N \cdot (K - 1)) \times m\) matrix of shape modes \( Q \), and the \(m \times m\) diagonal matrix of variances \( \Lambda = \text{diag}([\lambda_1, \ldots, \lambda_m]^\top) \), where \( \lambda_1 \geq \cdots \geq \lambda_m \). \( m < M \) is usually chosen to account for a certain percentage of the variability (e.g. 90%).

To construct our shape term, we decompose \( \phi = \phi_{\text{PC}} + \phi_{\text{orth}} + \bar{\eta} \), where \( \phi_{\text{PC}} \) lies in the span of the shape modes \( Q \) and \( \phi_{\text{orth}} \) is orthogonal to each of the shape modes (\( Q^\top \phi_{\text{orth}} = 0 \)). We divide our shape energy term into two parts to separately penalize the two components of \( \phi \). The first part is given by a squared Mahalanobis distance:

\[
E_{\text{PC}}(\phi) = (\phi_{\text{PC}})^\top Q \Lambda^{-1} Q^\top (\phi_{\text{PC}})
= (\phi - \bar{\eta})^\top Q \Lambda^{-1} Q^\top (\phi - \bar{\eta}).
\]

We note (4.49) does not penalize \( \phi_{\text{orth}} \). A common practice is to constrain \( \|\phi_{\text{orth}}\| = 0 \).
To allow for more flexible segmentations, we introduce a second shape energy term penalizing $\phi_{\text{orth}}$:

$$E_{\text{orth}}(\phi) = \|\phi_{\text{orth}}\|^2 = \|(\phi - \bar{\eta}) - QQ^\top (\phi - \bar{\eta})\|^2.$$  \hspace{1cm} (4.50)

We combine these to get

$$E_{\text{shp}}(\phi) = E_{\text{PC}}(\phi) + w_4 \lambda_m E_{\text{orth}}(\phi),\hspace{1cm} (4.51)$$

where $E_{\text{orth}}$ is multiplied by $\lambda_m^{-1}$ to ensure that variations out of the span of the shape modes are penalized at least as much as variations within the shape modes. $w_4 \to \infty$ ensures the segmentation lies on the span of the shape modes. Typically, we will simply set $w_4 = 1$, and penalize the out-of-modes variation as much as the smallest mode, allowing for a more flexible segmentation. We note this energy term is strictly convex, and thus will guarantee the total energy functional is strictly convex.

One issue with implicit shape models like the one described here is that they are poor at capturing variability in lower order shape moments, such as pose and size, but strong at capturing variability in higher shape moments. To account for this, a preprocessing step is usually applied to images to align their poses. However, general image alignment requires a registration step, which is expensive and often not convex. To circumvent this issue in our thigh segmentation problem, we employ an anatomically motivated alignment step based on our presegmentation from Sec. 4.3.1.

### Thigh Muscle Shape Model

Besides acting as a regional prior for the segmentation, we use the presegmentation $\pi_{\text{opt}}$ to guide the alignment of images for the purpose of building an anatomical shape prior. We use $\pi_{\text{opt}}$ to find an affine transformation for each image designed to remove pose and size variability. In Sec. 4.1.1, the images are cropped from just above the knee to just below the hip using a supervised method, so the slices in each image already correspond. Further, the out-of-plane resolution is much greater than the in-plane resolution, so out-of-plane rotations result in significant image blurring. For these reasons, we focus on in-plane alignment. This leaves 5 degrees of freedom to eliminate for each slice: 2 for translation, 1 for rotation, and 2 for scaling.

The translation is determined by aligning the centroid of the bone to a common location. The rotation and scaling are determined by fitting an ellipse to the detected muscle region,
rotating the image around the bone so the major axes of the ellipses align, and scaling the slices anisotropically along the ellipse axes so the muscles are roughly the same size (Fig. 4.6). This procedure may result in adjacent image slices being transformed differently and no longer being aligned, so we take each of the 5 transformation parameters for each slice and fit to them a linear function of the slice index, so the transformations vary linearly across slices, resulting in a 3D affine transformation. This alignment scheme has the benefit of being straightforward and initialization independent. It is possible that a general non-rigid registration algorithm could provide a better alignment, but this would introduce user-dependence or free parameters into the algorithm. Further, after a non-rigid alignment the remaining shape variability captured by the PCA shape model will be less indicative of actual anatomical variability and more a function of the registration errors.

In thigh muscle segmentation, we want our shape model to focus on capturing intermuscular boundaries correctly, as the muscle-background boundaries are relatively easy to detect from image gradients. Implicit PCA shape models often capture the variability in larger structures more accurately than smaller structures, simply because larger structures contain more voxels, which increases their overall influence on the PCA modes. We can offset this bias towards larger structures by constructing our GLR transformation, using the technique from Sec. 4.2.2. An example of a mean thigh muscle segmentation from a PCA shape model is seen in Fig. 4.7.

4.3.3 Boundary Energy Terms

When constructing a boundary term, we wish to encourage the segmentation to change from one label to another in the vicinity of anatomical boundaries in the image. Often, anatomical boundaries can be detected using image gradient information or other simple
image features. In the problem of thigh muscle segmentation, such simple techniques are not sufficient, as we will discuss later in this section. For now, however, we assume we have a boundary likelihood function \( h : \Omega \rightarrow [0, 1] \) and a vector field \( \mathbf{d}^1 : \Omega \rightarrow \mathbb{S}^2 \) normal to the most likely boundary plane at each voxel (where \( \mathbb{S}^2 \) is the unit sphere) Setting \( h = 0 \) will result in an image independent boundary term. If \( \mathbf{d}^1 \) is unavailable, it could be estimated using gradients of \( h \). Note that, for convenience, we focus specifically on 3D images in this section, though all of the concepts apply directly to 2D segmentation.

We define \( \mathbf{d}^2, \mathbf{d}^3 : \Omega \rightarrow \mathbb{S}^2 \) as two vector fields that, along with \( \mathbf{d}^1 \), form orthonormal bases at each voxel (thus \( \mathbf{d}^2(x) \) and \( \mathbf{d}^3(x) \) span the plane tangent to the detected boundary at \( x \in \Omega \)). To incorporate \( h \) and \( D(x) = [\mathbf{d}^1(x), \mathbf{d}^2(x), \mathbf{d}^3(x)] \) into an energy term, we penalize gradients in the segmentation’s components in directions other than orthogonal to the boundary, and penalize changes orthogonal to the boundary according to the likelihood \( h \). Specifically, defining the diagonal matrix \( H(x) = \text{diag}([(1 - h(x)), 1, 1]) \), we define our energy term as

\[
E_{\text{bnd}}(\phi) = \sum_{x \in \Omega} \sum_{i=1}^{K-1} \|H(x)^{-1} \cdot D(x)\cdot \nabla \phi^i(x)\|_p^\ell,
\]

where \( \ell = \{1, 2\} \). We ensure \( (1 - h(x)) > \epsilon > 0 \) for numerical stability. By projecting \( \nabla \phi^i(x) \) onto \( D \), we penalize the component of \( \nabla \phi^i(x) \) parallel to \( \mathbf{d}^1(x) \) by a factor of only \( (1 - h(x)) < 1 \) as much as the component of \( \nabla \phi^i(x) \) in the span of \( \{\mathbf{d}^2(x), \mathbf{d}^3(x)\} \), encouraging the segmentation to change labels across the detected muscle boundaries. Choosing \( \ell = 2 \) provides a more smoothly varying probabilistic segmentation and results in a simpler optimization, as (4.52) becomes a convex quadratic when discretized. Choosing \( \ell = 1 \), however, results in a modified total variation term that provides better results in regards to matching large changes in the probabilities with detected image boundaries.

### Boundary Term Optimization

This modified total variation term can be efficiently globally minimized using established primal dual techniques \([62, 308]\):

\[
E_{\text{bnd}}(\phi) = \sum_{x \in \Omega} \sum_{i=1}^{K-1} \max_{p^i(x) \in C(x)} p^i(x)^\top \nabla \phi^i(x)
\]

\[
C(x) = \left\{ p \in \mathbb{R}^3 \left| \left(\frac{p^\top \mathbf{d}^1(x)}{1 - h(x)}\right)^2 + \left(\frac{p^\top \mathbf{d}^2(x)}{1 - h(x)}\right)^2 + \left(\frac{p^\top \mathbf{d}^3(x)}{1 - h(x)}\right)^2 \le 1 \right. \right\}.
\]
The dual variables \( P = [p^1, \ldots, p^{K-1}] \) are constrained to the ellipsoids \( C(x) \) at each voxel, which is flattened along the direction \( d^1(x) \) by a factor of \( 1/(1 - h(x)) \) (Fig. 4.8).

To optimize (4.53), we employ a primal-dual proximal point method, which can guarantee convergence given small enough step size [237, 261]. We re-write our total energy functional as an optimization problem of the form

\[
\phi^* = \arg\min_{\phi} \max_{P \in C_P} E_{\text{cnv}}(\phi) + E_{\text{bnd-tot}}(\phi, P)
\]

\( C_P = \left\{ P = [p^1, \ldots, p^{K-1}] \mid p^i(x) \in C(x), \forall i \in \{1, \ldots, K - 1\}, \ x \in \Omega \right\} \). \hspace{1cm} (4.56)

\( E_{\text{cnv}} \) encompasses the convex regional and shape energy terms, and can be expressed as a quadratic:

\[
E_{\text{cnv}}(\phi) = \frac{1}{2} \phi^T M \phi + b^T \phi,
\]

where \( M \) is a positive semi-definite matrix and \( b \in \mathbb{R}^{N(K-1)} \). \( E_{\text{bnd-tot}} \) equals \( E_{\text{bnd}} \) with the implicit maximization of \( P \) removed:

\[
E_{\text{bnd-tot}}(\phi, P) = \sum_{x \in \Omega} \sum_{i=1}^{K-1} p^i(x)^T \nabla \phi^i(x)
\]

\[
= \sum_{x \in \Omega} \sum_{i=1}^{K-1} (\text{div } p^i(x)) \phi^i(x),
\]

where \( \text{div} \) is the divergence operator [62].
We solve (4.55) using a proximal point method [237, 261]. The idea is to iteratively optimize for the primal variables $\phi$ and the dual variables $P$, adding a soft constraint to prevent the solutions from changing too much in a given step. Given solutions $\phi_j = [\phi^1_j, \ldots, \phi^{K-1}_j]$ and $P^j = [p^1_j, \ldots, p^{K-1}_j]$ at iteration $j$:

**Primal Step:**

$$
\phi_{j+1} = \arg\min_{\phi \in \mathbb{R}^{N \cdot (K-1)}} E_{\text{cnv}}(\phi) + E_{\text{bnd-tot}}(\phi, P^j) + \frac{\|\phi - \phi_j\|^2}{2\tau_1}.
$$

(4.60)

With a fixed $P_j$, $E_{\text{bnd-tot}}(\phi, P_i)$ is linear in $\phi$ (from (4.59)), so (4.60) is a convex quadratic, and can be globally minimized.

**Dual Step:**

$$
P_{j+1} = \arg\max_{P \in C_P} E_{\text{bnd-tot}}(\phi_{j+1}, P) - \frac{\|P - P_j\|^2}{2\tau_2}.
$$

(4.61)

$$
\|P - P_j\|^2 = \sum_{i=1}^{K-1} \|p^i - p^i_j\|^2.
$$

(4.62)

With a fixed $\phi_{j+1}$, $E_{\text{bnd-tot}}(\phi, P_i)$ is linear in $P$ (from (4.58)), so (4.61) is a concave quadratic. $C_P$ is a hyper-ellipse, a convex set that can be projected onto. Thus the dual step can also be globally optimized.

This algorithm converges for small enough $\tau_1$ and $\tau_2$ [237]. Though we do not yet have a theoretical guarantee of convergence for specific values, empirically we found $\tau_p = \tau_q = 0.1$ provided fast and reliable convergence.

**Thigh Intermuscular Boundary Detection**

Accurate detection of intermuscular boundaries in thigh MRI data is confounded by several factors. First, the layer of intermuscular fat dividing the muscle tissue can often be of a sub-voxel thickness, resulting in PVE obscuring the boundary. Second, intramuscular fat and other textures inside muscle regions can create many false edges (Fig. 4.9, Fig. 4.10). Third, the intermuscular boundaries vary greatly in both thickness and appearance, even within a single slice of an MRI (Fig. 4.10a). In this section, we develop techniques for calculating the boundary likelihood function $h$ for thigh MRI data. We propose 3 techniques for detecting intermuscular boundaries, each building on and improving the previous.

**Curvature Filters:** Our first technique is based on the observation that intermuscular boundaries tend to have a gray-light-gray profile (Fig. 4.10), and thus a curvature-based filter might be effective. Our technique consists of several steps:
Figure 4.9: An example showing the false boundaries created by intramuscular fat, as found by a Canny edge detector (b), with segmentation (i.e. true boundaries) shown in red for comparison (c).

1. Blur the image with a Gaussian of fixed width.

2. Find the Hessian at each voxel $x$ and find its eigenvalues $\lambda_1(x) \geq \lambda_2(x) \geq \lambda_3(x)$ and corresponding eigenvectors $v^1(x)$, $v^2(x)$, and $v^3(x)$. $v^1(x)$ and $\lambda_1(x)$ correspond to the direction of maximum curvature and its curvature value.

3. Intermuscular boundaries should be sheet-like, and for voxels on sheet-like boundaries, we expect $\lambda_1(x) \gg \lambda_2(x), \lambda_3(x)$ [97]. Thus, we take

$$h_0(x) = \sqrt{\frac{\lambda_1(x)^2}{\lambda_1(x)^2 + \lambda_2(x)^2 + \lambda_3(x)^2}}$$  \hspace{1cm} (4.63)$$
as a boundary indicator at $x$.

4. For $x$ on an intermuscular sheet of fat, $v^2(x)$ and $v^3(x)$ will approximate the plane tangent to the sheet, and other nearby voxels in this plane should also have a high value for $h$. Thus, we anisotropically blur $h$ in this plane, denoting the result $h_1$. Voxels with a high value after this blurring are likely on a boundary.

5. Further, voxels in the direction of $v^1$ will be further away from the boundary, and thus should have a lower value for $h$. Thus, we blur $h$ from step 4 along this direction, denoting the result $h_2$. Voxels with a high value after this blurring are likely not on a boundary.

6. Our boundary indicator function is found by subtracting the results of step 5 from the results of step 4, getting $h(x) = h_1(x) - h_2(x)$, and normalizing to $[0, 1]$.
Figure 4.10: An example showing true boundaries (green boxes) and false boundaries (red) (d), demonstrating the difficulty in distinguishing them. The appearances of the true and false boundaries overlap heavily, thus a powerful multi-modal classifier is necessary for accurate boundary detection.

Steps 4, 5, and 6 can be combined into a single directional filter applied to each voxel, which we refer to as a directional boundary filter (DBF):

$$DBF(f, x, v^1) = \sum_{y \in \mathcal{N}(x)} f(y) \exp \left( -\frac{\|x - y\|^2}{2\sigma^2} \right) \left( \frac{1}{2} - \frac{\langle u(x) - u(y), v^1 \rangle}{\|u(x) - u(y)\|} \right)$$  \hspace{1cm} (4.64)

where $\mathcal{N}(x)$ is a neighborhood are voxel $x$, $u(x)$ is the location vector of $x$, and $\sigma$ controls the width of the filter. If $(u(x) - u(y))$ is parallel to (orthogonal to) $v^1$, the contribution of $y$ to the filtered value at $x$ will be negative (positive). The DBF is useful in general to remove false boundaries whenever we have a boundary indicator function and a corresponding vector field normal to the detected boundary, as false boundaries are less likely to exhibit this sheet-like structure.

**Gabor Filters:** While the curvature filters are effective, they have several drawbacks

- The Gaussian has a fixed width;
- Curvature is not sufficient for detecting all boundaries;
- Strong enough false boundaries (intramuscular fat) may still be detected.
We address the first two drawbacks with an intermediate approach. Instead of only looking at the curvature of the smoothed image, we use a set of 20 3D directional Gabor filters, designed to capture a range of boundary widths and appearances, oriented in 18 different directions, for a total of $20 \times 18 = 360$ filters. Since the out-of-plane resolution is much greater than the in-plane resolution (see Sec. 4.1.1) and since the thigh muscle boundaries are usually oriented perpendicular to the image plane, the 18 directions are all chosen in-plane, aligned with the angles $\{10^\circ, 20^\circ, \ldots, 180^\circ\}$. A Gabor filter in the direction of the unit vector $\mathbf{v}'$ is given by

$$g(\mathbf{v}) = \exp\left(-\frac{\|\mathbf{v}\|^2}{2\sigma^2}\right) \cos\left(\frac{2\pi \langle \mathbf{v}', \mathbf{v} \rangle}{\gamma}\right)$$

or

$$g(\mathbf{v}) = \exp\left(-\frac{\|\mathbf{v}\|^2}{2\sigma^2}\right) \sin\left(\frac{2\pi \langle \mathbf{v}', \mathbf{v} \rangle}{\gamma}\right).$$

We use $\sigma^2 \in \{0.5, 1, 2, 4, 8\}$ and $\gamma \in \{0.25, 1\}$ for both sin and cos to get our 20 filters for a given direction.

We now apply each of the 20 filters for each of the 18 different directions and take the value of the filter with maximum response at each voxel. We then apply the DBF from (4.64) to these maximum responses. This approach detects more varied muscle boundaries than the initial approach, but the more filters that are used, the more likely one will give
(a) Thigh Slice  (b) Boundary Likelihood  (c) True Boundaries

Figure 4.12: An example of the boundary likelihoods found using our RF classifier. In (b), whiter shading corresponds to higher likelihoods. (c) shows the manual boundaries.

a false positive response. This could be mitigated by using, instead of the maximum response, differences between filter responses in different directions, similar to (4.63). Our full boundary detection approach (described next) takes this idea a step further, incorporating all the filter responses and using available training data to learn the appearance of muscle boundaries.

**Random Forest Boundary Detection:** Wang et al. account for the difficulty in locating muscle boundaries by adopting a random forest (RF) classifier [41], trained from existing segmentations, to attempt to locate landmark points. We expand on this technique and train an RF classifier to identify intermuscular boundaries. RFs are able to efficiently model complex unknown distributions, including distributions with many modes, and thus may be able to account for the large variety in intermuscular boundary appearances.

We use the 360 filter responses to construct a feature vector for each voxel, and then use these features to train an RF classifier on the training data. We use 100 trees in the RF, each with a depth of 7 and 20 features randomly chosen per tree and each node splitting on a single feature. The trade-off in these parameters is computation time versus accuracy; we found increasing these parameters did not significantly alter accuracy.

If we append the filter responses in a fixed order to create a length 360 feature vector, boundaries that are similar in appearance but oriented differently will have completely different feature vectors, complicating the training process. Instead, to construct the feature vectors for training, we shift the filter responses so the first 20 filters are oriented towards the nearest boundary (calculated using the training segmentations), with the remaining
filters in clockwise order, as illustrated in Fig. 4.11. For voxels on a boundary, this results in the first 20 feature components corresponding to responses for filters oriented across the boundary, adding rotational invariance to the classifier.

Since we are focused on detecting intermuscular boundaries (as mentioned, other boundaries are easy to detect), we only train on voxels that are labeled as muscle in the training segmentations. Intermuscular boundary voxels are assigned target value 1, voxels adjacent to these are also assigned target value 1 (since a near miss will still provide decent segmentation accuracy and this increases the amount of positive training data), and other muscle voxels are assigned target value 0.

Incorporating all of the filters into the RF (instead of just the ones oriented across a boundary) is useful for providing additional context to the classifier. For example, while filters in a certain direction may have a high response, if filters in orthogonal directions also have high responses, it may indicate a false boundary.

Given a novel image, we again extract the 360 filter responses at each voxel, but we are not able to shift these filters to align with the nearest boundaries, since these boundaries are not known. Instead, we apply the RF classifier to 18 different feature vectors, generated from all 18 possible shifts of the filter responses, which gives us a boundary likelihood for each voxel/direction pair. As a final step, for each of the directions, we apply a DBF to the likelihoods, and take the maximum resulting value at each voxel. This results in a boundary likelihood function $h: \Omega \rightarrow [0,1]$ and corresponding boundary normal vector field $d^1: \Omega \rightarrow S^2$. An example of the boundary likelihood function $h$ is shown in Fig. 4.12. We incorporate $h$ and $d^1$ into our segmentation algorithm using (4.52) with $\ell = 1$.

4.4 Segmentation Results

In this section, we first present synthetic results demonstrating the power of our general segmentation framework, focusing on the usefulness of the four important properties our framework incorporates (convexity, shape prior, multi-label, and probabilistic). We then present results specific to thigh muscle segmentation and demonstrate how our contributions, both general and specific to the problem, improve the accuracy and reliability with which the muscles are segmented.
CHAPTER 4. FULLY AUTOMATED SEGMENTATION

87

Figure 4.13: Examples of our segmentation algorithm applied to axial brain slice images. We see the shape prior greatly improves the segmentation. In most color figures of brain images seen here, red corresponds to CSF, green to gray matter, blue to white matter, and black to background. In gray scale images, lighter shades correspond to higher intensities/probabilities/entropies.

4.4.1 Synthetic Experiments

We compare the results achieved by our method, which satisfies the four aforementioned properties, to results achieved when one of the properties is removed. We show how turning off the shape prior greatly reduces the accuracy of the segmentation. We show the benefits of multi-label segmentation over a binary method applied to each region sequentially. We compare to an expectation-maximization (EM) segmentation framework [241] that does not guarantee a global optimum to demonstrate the benefits of having a convex energy while retaining the other 3 properties. We also demonstrate the benefits of a probabilistic segmentation.

To run our experiments, we took 20 magnetic resonance imaging (MRI) axial slices of size 60 × 160 of different brains from the Internet Brain Segmentation Repository (IBSR)
Figure 4.14: The segmentation of both an image and the same image partially obstructed, demonstrating the usefulness of our shape prior. The DSC of the segmentation without and with the obstruction is 0.8114 and 0.7995, respectively. Thus, despite the obstruction, the DSC only drops 1.5%. In 4.14d and 4.14h we note the increase in entropy under the obstruction (red arrow), corresponding to uncertainty in our probabilistic segmentation. Note that while such a glaring occlusion does not typically appear in MRI images, small artifacts and noise do. This example is designed to make the effect of the shape prior obvious.

We refer to the website (http://www.nitrc.org/projects/ibsr), along with crisp expert segmentations of the images into $K = 4$ regions: white matter, gray matter, cerebrospinal fluid (CSF), and background. These slices were affinely registered to create 20 aligned 2D images and their corresponding ground truth segmentations. We convert crisp segmentations to probabilistic ones by applying the inverse LogOdds transformation to their SDMs, using the technique from Sec. 3.2.1 and enforcing a minimum probability of $\frac{1}{100}$. The probabilistic segmentations are then mapped to ILR space. In the subsequent experiments, we use a leave-one-out testing scheme, removing one of the images and using the remaining images to construct shape models and regional priors. The $m = 15$ largest variance eigenvectors of the training samples were calculated using PCA and used to create a shape model.

For our energy function, we use (4.44) for our regional term, (4.51) for our shape term, and (4.52) with $\ell = 2$ for our boundary term. The weights were set empirically as $w_1 = 20$, $w_2 = 20$, and $w_4 = 1$. 
Figure 4.15: An obstructed image is segmented into 2, 3, and 4 regions, showing how multi-region interactions improve the segmentation of individual regions. The bottom row shows the ground truth for the CSF and how the CSF’s segmentation improves as more regions are included.

We first demonstrate the usefulness of shape priors in multi-label segmentation. The fourth column in Fig. 4.13 shows the resulting multi-label \((K = 4)\) segmentations of 3 of the 20 brain slices without shape prior using the RW segmentation algorithm [123], an algorithm that only satisfies the 3 other properties. For measuring accuracy, we use the Dice similarity coefficient (DSC) between a segmentation (thresholded to crisp, and averaged across regions) and its ground truth (GT) manual segmentation. The average DSC of all 20 of the segmentations without a shape prior (RW) was 0.5226 ± 0.1411. The DSC was greatly improved by the inclusion of the shape prior in our method to 0.8273 ± 0.0345. This is close to the DSC of the manual segmentation projected onto the shape space: 0.9029 ± 0.0136.

Handling occlusion is a natural way to demonstrate the power of a shape prior. Because of our shape prior, our segmentation algorithm is able to segment the ventricles in Fig. 4.14 despite the fact that they are about partially occluded. Our algorithm approximates the shape of the left ventricle based on the shape of the right ventricle due to the lateral symmetry encoded in the shape prior.

Extending this concept to multi-label segmentations, we show how the shape prior encodes regional interactions, which is not possible if a binary segmentation method was used.
CHAPTER 4. FULLY AUTOMATED SEGMENTATION

Figure 4.16: Analysis of segmentation uncertainty. A close up of the CSF region in the center of an image for the purposes of understanding the sources of the uncertainty in the segmentation, shown by its entropy in the bottom right figure. We can infer the origin of entropy of the segmentation (4.16h) from the intensity priors (4.16f) and the projected ground truth (4.16e). Areas of high intensity gradient (4.16c) create uncertainty in the intensity prior, and thus in the segmentation. Areas where the ground truth (4.16d) and the projected ground truth differ indicate features not representable by the shape prior, and again this affects the segmentation’s certainty.

to segment each region individually. To demonstrate how this improves the segmentation, we take the extreme example of completely obstructing the ventricles, and then segmenting the image into 2, 3, and 4 regions (Fig. 4.15). The segmentation of the obstructed ventricles becomes more accurate as more regions are considered because the shapes of the other regions correlate to the shape of the hidden region, improving the segmentation.

To demonstrate the usefulness of a probabilistic segmentation, we return to Fig. 4.14 and compare the segmentations obtained for the image in Fig. 4.14a and its obstructed counterpart, the image in Fig. 4.14e. We see that in Fig. 4.14h the obstructed area has a much higher entropy (red arrows) than in Fig. 4.14d, where no obstruction existed.

To further demonstrate the usefulness of a probabilistic segmentation over a crisp segmentation, we show how uncertainty in a probabilistic segmentation is related to uncertainty in the intensity and shape priors. In Fig. 4.16, we have a close up of the red-colored ventricles in the center of one of the brain slices. In Fig. 4.16h, we see that the pixels along
the bottom and inside boundaries of the ventricles have high entropy. This is caused by the large gradient in the image intensity there (Fig. 4.16c) creating uncertainty in the intensity prior. In Fig. 4.16h we see that the pixels corresponding to the green regions on the outside of either of the ventricles also have high entropy. As the segmentation in this area differs between the ground truth (Fig. 4.16d) and the projected ground truth (Fig. 4.16e), we deduce that this feature is not easily represented in the shape space, which creates greater uncertainty in the shape prior. Crisp segmentations produced by similar algorithms are not able to highlight these areas of uncertainty [96, 237, 297].

As our method is strictly convex, no initialization is required. To demonstrate the advantages of the convexity of our energy, we show how different initializations affect the results of a non-convex method that achieves our other goals, the EM-based algorithm employing the LogOdds representation from Pohl et al. [241]. In Fig. 4.17, we segment the image from the second row of Fig. 4.13 using a shape space constructed from the same training data used for our method, again with \( m = 15 \) eigenmodes.

This EM algorithm takes an element of the shape space and a set of intensity inhomogeneity parameters per pixel as initialization. In the top row of Fig. 4.17, we initialize with the mean shape and zeros for the inhomogeneity parameters, resulting in a poor segmentation with a DSC of 0.6766 with the GT. However, if we use the GT to find the optimal initialization, we see in the bottom row of Fig. 4.17 that the resulting segmentation is quite good, with a DSC of 0.8337 with the GT. Given certain initializations, the EM algorithm performs well here, but with other initializations, the method gets stuck in local minima. What initializations are “good” cannot necessarily be determined beforehand, and thus local minima can always cause problems for non-convex methods. Such problems are avoided with a strictly convex method such as ours, whose results are independent of initialization.

### 4.4.2 Thigh Muscle Segmentation

We test our automated thigh muscle segmentation technique by randomly taking 20 images for training and use the remaining 20 for testing. We choose 10 of the training images from the COPD patients and 10 from the healthy patients, ensuring our trained models incorporate both healthy and pathological examples.

Our parameters \( w_1 \), \( w_2 \), and \( w_3 \) were set in the training stage using only the training data. For \( w_1 \) and \( w_2 \), we found the best combination of weights, in terms of average DSC, out of \( 20 \times 20 = 400 \) possibilities when running our segmentation algorithm on each of the
Figure 4.17: The results of segmenting a brain slice using a non-convex the EM method [241] with different initializations. The initialization consists of an initial probabilistic segmentation in the shape space and an initial intensity inhomogeneity parameter for each pixel. When given the optimal initialization (bottom row), a good segmentation is found, having a DSC of 0.8337. However, when given an initialization of zeros (top row), the results are much worse, with the resulting segmentation having a DSC of 0.6766. The minimization gets stuck in a local minimum, as can often happen with non-convex energies. See the second row of Fig. 4.13 for our method’s performance on this image (requiring no initialization).

training images in a leave-one-out approach, giving \( w_1 = 5 \) and \( w_2 = 50 \). \( w_3 \) was set by finding the value that provided the best match between the presegmented muscles and the manually segmented muscles in the training data, giving \( w_3 = 10 \). By setting the parameters using only the training data, we avoid biasing the weights. \( w_4 \) was set to 1.

We perform the 7 steps in Algorithm 2. Our main contributions related to thigh muscle segmentation correspond to steps 1 and 2 (presegmentation and alignment), step 3 (RF boundary classifier), and step 5 (the GLR segmentation representation). We will evaluate the importance of each of these contributions by removing them one at a time from the segmentation algorithm and comparing the change in segmentation accuracy, quantified using the DSC with the manual segmentation.

We remove steps 1 and 2 by using the GMM probabilities instead of the presegmentation to construct \( E_{\text{reg}} \) in (4.47) and by performing a rigid alignment of the foreground voxels
Algorithm 2 An Overview of the Thigh Muscle Segmentation Framework

1: Construct a presegmentation of all the images (training and testing) (see Sec. 4.3.1, Fig. 4.4).
2: Use the presegmentation to align all the images (see Sec. 4.3.2, Fig. 4.6).
3: Train an RF boundary classifier (see Sec. 4.3.3, Fig. 4.12).
4: Convert the training segmentations to probabilistic segmentations by applying LogOdds to the SDMs of the labels (see Sec. 3.2.1).
5: Use boundary overlaps of the labels in the training segmentations to choose an appropriate GLR transformation, and apply it to the probabilistic training segmentations (see Sec. 4.2.2).
6: Construct a PCA shape model (see Sec. 4.3.2, Fig. 4.7).
7: For each testing image, construct and minimize the appropriate energy functional (4.42) to find the optimal probabilistic segmentation.

Figure 4.18: A summary of our method and the various techniques we compare against (see Algorithm 2). Colored blocks correspond to steps from Algorithm 2, and gray blocks correspond to one of our contributions being replaced by another technique.

Instead of the alignment described in Sec. 4.3.2, denoting the resulting algorithm “GLR-No Preseg”. We remove step 3 by using the existing curvature-based boundary function described at the beginning of Sec. 4.3.3 [8], denoting the resulting algorithm “GLR-No RF”. We remove the use of the GLR transformation in step 5 by instead using the LogOdds and ILR transformations, denoting the resulting algorithms “LogOdds-Full” and “ILR-Full” respectively. We denote our full algorithm “GLR-Full”. Fig. 4.18 provides a summary. We note that the RF boundary technique detected $22 \pm 7\%$ as many false positives compared to a Canny edge detector, while detecting $130 \pm 18\%$ true positives.

We evaluate our method against the recent thigh segmentation techniques of Baudin et al. [24]. The technique of Baudin et al. achieves an excellent mean DSC of $0.86 \pm 0.07$, though their data differs significantly from ours, as the image acquisition was designed to capture
CHAPTER 4. FULLY AUTOMATED SEGMENTATION

Figure 4.19: A comparison of manual and automated segmentation results.

Figure 4.19: A comparison of manual and automated segmentation results.

thigh muscle boundaries [26]. Their images exhibit stronger intermuscular boundaries and less false boundaries from intramuscular fat (Fig. 4.23). These complications with evaluation are discussed in more detail later in this section. We compare to their technique by running an implementation of it on our data. Their technique incorporates a PCA shape model into a random walker segmentation formulation (RW), so we use the same training and testing sets to build their shape model. We set the free parameters in their algorithm similarly to how we set our $w_1$, $w_2$, and $w_3$, using only the training data. Baudin et al. also introduce a technique for automatically finding seed voxels in the centers of muscles, based on detecting image gradients [23]. We found the false boundaries from intramuscular fat in our images made placing seeds correctly difficult, but to ensure a fair comparison, we use the manual segmentations of the testing images to provide the RW algorithm with 5 to 15 seeds for each muscle (depending on the muscle volume) spread along their 3D medial axes (note these seeds are not used when running our method).

The comparative segmentation results of the methods from Fig. 4.18 are summarized in Fig. 4.20, where we see that GLR-Full in general outperforms the other methods in terms of mean DSC. More detailed results for individual muscles are seen in Fig. 4.21. The muscle labels are ordered by mean volume across the images, with label 1 corresponding to the smallest muscle (Fig. 4.1). We note that in Fig. 4.21 the GLR-Full algorithm typically
Figure 4.20: A comparison of GLR-Full to each of the other 5 methods in Fig. 4.18, summarized across each label and each image.

provides higher DSC values compared to the other algorithms that do not include all of our contributions, with the exceptions being RW-Shape Prior and ILR-Full. RW-Shape performs better on muscles with strong intermuscular boundaries and minimal intramuscular fat (e.g. label 1), but performs significantly worse on muscles with weak intermuscular boundaries (e.g. label 8). ILR-Full performs better on some of the larger muscles, as is expected, because the GLR transformation was designed to ensure the smaller muscles were captured better by the segmentation. Both small and large muscles are important for COPD treatment planning [132], so properly segmenting smaller muscles is a key feature.

Another the recent thigh segmentation technique is that of Gilles and Magnenat-Thalmann [118]. Gilles and Magnenat-Thalmann also focus on modeling other anatomy, such as ligaments and cartilages. They report a mean distance between manual and automatic segmentations (without user input) of $1.58 \pm 1.92$ mm, whereas the mean distance found from our automatic segmentation after thresholding is slightly less, at $1.54 \pm 0.67$ mm. We note that our method focuses specifically on segmenting smaller muscles more accurately, while focusing more closely on larger muscles may provide a smaller mean distance to the manual segmentation due to a larger surface area.

**Segmentation Uncertainty**

Our probabilistic segmentation technique actively attempts to maintain meaningful probabilities at each voxel. As such, we should be able to extract useful segmentation uncertainty information from these probabilities. We follow Saad et al. [275] and calculate uncertainty
CHAPTER 4. FULLY AUTOMATED SEGMENTATION

Figure 4.21: A comparison of the various algorithms, where positive values indicate GLR-Full performs better. See text for discussion. The results are summarized in Fig. 4.20.
Figure 4.22: An analysis of uncertainty information and error. (a) shows the voxels of top 5% most uncertainty (green), the erroneous voxels (red), and their intersection (yellow). (b) and (c) show the “error precision” (yellow/(yellow+green)) and “error recall” (yellow/(yellow+red)) for percentiles of uncertainty across all images. A general strategy for identifying errors in a crisp segmentation might be to simply take boundary voxels as being likely erroneous, so we present this as a baseline in (b) and (c). We see that the top 5% of voxels have about 20% higher precision but the same recall as the boundary voxels, indicating uncertainty is useful for error detection.

as 1 minus the difference between the highest and second highest label probabilities, a measure of how close the segmentation was to assigning a different label to a voxel. These uncertainty values should correlate well with segmentation error; more specifically, the set of voxels with the highest uncertainty should have both high “error precision” (i.e. most of them should be erroneous) and high “error recall” (i.e. most of the error should be contained in this set). We evaluate these assertions in Fig. 4.22.

COPD Classification

A unique component of our data is the inclusion of COPD patients. While inclusion of COPD data complicates the segmentation process, through increased intramuscular fat and shape variability, it also provides an opportunity to evaluate our shape model by attempting to distinguish COPD and healthy images. We build two new shape models, one from the 10 COPD training images and one from the 10 healthy training images, and then construct the energy term $E_{shp}$ from (4.51) for both. We evaluate these two shape terms at each of the
automatically generated segmentations from the testing set, the intuition being that the we should be able to determine if a patient has COPD symptoms based on which shape energy gives a lower value (indicating a better shape match). We found that, of the 10 COPD and 10 healthy images from our testing set, 6 of the COPD images and 9 of the healthy images were properly classified using this technique. This is not a statistically significant result (due to the small sample size), but indicates that potential differences in higher order shape moments are being captured by these shape models.

**Comparison to Other Works**

Due to the lack of publicly available thigh muscle data and differences in imaging modalities between related publications, comparison to competing methods proves difficult. The works of Baudin et al. [23–25] are perhaps closest to this work algorithmically, but the data they use exhibits significantly different properties (Fig. 4.23). Due to the stronger intermuscular boundaries and less false edges, their RW-based algorithm is well-suited to the task (and performs better than our method would on similar data). However, as demonstrated in this section, our method performs better on our data. Future work toward establishing standardized testing data for this challenging problem could ameliorate these difficulties and speed research progress.

![Comparison of MRI Data](image)

Figure 4.23: Our data (a) compared to data used by Baudin et al. (b).
4.5 Chapter Summary

In this chapter, we have introduced a novel fully automated segmentation framework that is the first to simultaneously incorporate four important properties: initialization independence, incorporating a statistical shape model, simultaneously segmenting multiple regions, and evaluating uncertainty through probabilistic labels. We achieve these goals by representing segmentations using our GLR transformations that facilitate the construction of multi-label probabilistic shape models, and defining convex energy terms over these representations that captures desirable features.

We demonstrated how a GLR transformation be constructed to enforce certain distances between pairs of labels, and that appropriate choice of GLR transformation can encode prior anatomical volume and adjacency information into a segmentation representation. The distances between labels represent implicit label transition penalties, similar to penalties encoded explicitly by other techniques [175]. Once the GLR transformation has been applied, the penalties are automatically enforced by standard gradient and distance terms. In the future, we will explore other techniques for incorporating prior information into the GLR transformation.

We applied this segmentation framework to the challenging problem of thigh muscle segmentation, introducing several additions designed specifically for this task. Our anatomically-based alignment scheme leverages a relatively easy to solve subproblem (pre-segmentation) to perform an automatic rigid alignment, facilitating the construction of a PCA shape model. Non-rigid alignments often introduce user-dependence or free parameters into the algorithm, and it is not clear what shape variability will remain to be captured by PCA shape models.

Our RF boundary detection technique learns the appearance of both true and false boundaries, with minimal prior information and without the free parameters often associated with determining boundary appearance. In the future, we will extend the RF to learn boundaries between specific pairs of labels, and include this information in our energy term.

Our preliminary results indicate that statistical shape models can capture variability in the higher order shape moments between healthy and COPD thigh muscles. For future work, we will investigate how these shape models are useful for subsequent analysis the effects of COPD on these muscles, leading to better treatment planning.
Chapter 5

Precomputation

The previous two chapters focused on fully automated segmentation schemes designed to be run offline, before any user interaction. While these schemes were designed to encode uncertainty information that could be used online, to guide subsequent user interaction, the form this interaction might take was not specifically addressed. In this chapter, we focus on algorithms designed explicitly with offline and online components, where the offline precomputation does not provide explicit results itself, but is designed to compute data that can be used to increase the speed or accuracy of the online algorithm.

We focus specifically on the random walker (RW) algorithm, a popular tool that has been applied to a variety of medical imaging tasks such as image segmentation (RWIS) and registration (RWIR). The RW algorithm’s popularity is due to several useful properties, including:

- It provides a single globally optimal solution;
- It seamlessly allows user interaction;
- It provides probabilistic results, useful for evaluating uncertainty.

The first part of this chapter focuses on precomputation in RWIS, and the second focuses on precomputation in RWIR.
5.1 Fast Random Walker Image Segmentation

Random Walker Image Segmentation

Let $J$ be an image over the voxel set $\Omega$, with $|\Omega| = N$. The RW algorithm assigns each voxel a $K$ length probability vector (i.e. a positive vector that sums to 1), which we will denote by the $N \times K$ row stochastic matrix $U = [u^1, \ldots, u^K]$. In RWIS, these probabilities correspond to $K$ different structures in an image. In RWIR, the probabilities correspond to $K$ different potential displacement vectors.

The RW algorithm consists of defining constraints on $U$ and then regularizing it subject to these constraints. Typical constraints are either soft, consisting of an $N \times K$ matrix $P = [p^1, \ldots, p^K]$ of prior probabilities that $U$ should be similar to, or hard, consisting of user specified “seeds” voxels that have one label fixed to probability 1. In RWIR, these seeds take the form of landmarks. In order to simplify the notation related to the seed constraints, we re-order the pixels so the rows corresponding to seeded pixels come first, and then divide $U$ into “s”eeded and “n”on-seeded components using block matrix notation:

$$U = \begin{bmatrix} U_s \\ U_n \end{bmatrix}, \quad u^k = \begin{bmatrix} u^k_s \\ u^k_n \end{bmatrix},$$ (5.1)

where $U_s$ is fixed and $U_n$ are the remaining unknowns. We denote $S \ll N$ as the number of seeds.

In RWIS, the prior probabilities can be calculated from statistics on the user input seeds, allowing spatially disjoint objects to be segmented without seeding each object individually [123]. In RWIR, the soft constraints are usually based on a local image similarity measure, and are the driving force behind the registration.

The RW regularization is done using an image graph, with vertices corresponding to voxels and weighted edges between neighboring voxels, where the weights control how similar the probabilistic labels of neighboring voxels should be. We define $W$ as the matrix of edge weights with components

$$w_{xy} = \begin{cases} \exp(-\beta|J(x) - J(y)|) & x \in \mathcal{N}(y) \\ 0 & \text{otherwise} \end{cases}$$ (5.2)

Here, $x, y \in \Omega$, $J(x)$ is the image’s intensity at $x$, $\mathcal{N}(y)$ is the set of pixels neighboring $y$, and $\beta \geq 0$ is a scalar parameter. Letting $D$ be the diagonal matrix with diagonal entries
given by the row sums of $W$ (or column sums, since $W$ is symmetric), the Laplacian matrix of the graph is defined as $L = D - W$.

The RW algorithm calculates the label probabilities by solving, for each $k \in \{1, \ldots, K\}$,

$$
\arg\min_{u_k^n} u_k^T L u_k + \Gamma \|u_k - p_k\|^2,
$$

(5.3)

where $\Gamma$ is an $N \times N$ diagonal matrix with spatially varying prior weights $\{\gamma_1, \ldots, \gamma_N\}$ on its diagonal. The matrix $\Gamma$ encodes spatially varying prior strength, which could be based on image information (e.g. relying more on the prior probabilities in homogeneous image regions with few defining features), on the uncertainty in the priors, on seed locations (e.g. rely less on prior probabilities near seeds), or other application dependent definitions.

The energy in (5.3) is convex, and the globally optimal solution can be found for each $k$. Writing $L$, $P$, and $\Gamma$ in block matrix notation similar to (5.1),

$$
L = \begin{bmatrix}
L_s & B \\
B^\top & L_n
\end{bmatrix}
$$

(5.4)

$$
P = \begin{bmatrix}
P_s \\
P_n
\end{bmatrix}, \quad P^k = \begin{bmatrix}
p^k_s \\
p^k_n
\end{bmatrix}
$$

(5.5)

$$
\Gamma = \begin{bmatrix}
\Gamma_s & 0 \\
0 & \Gamma_n
\end{bmatrix}
$$

(5.6)

the minimizing $u^k$, for each $k$, is found by solving the linear system of equations

$$
(L_n + \Gamma_n) u_n^k = \Gamma_n p_n^k - B^\top u_s^k.
$$

(5.7)

We can combine (5.7) into one equation for all $k$:

$$
(L_n + \Gamma_n) U_n = \Gamma_n P_n - B^\top U_s.
$$

(5.8)

This equation consists of $K$ linear systems, each of size $(N - S) \approx N$, and solving them can be very computationally expensive, which can disrupt the workflow when a user is waiting to provide additional input. However, $L$ is known offline, suggesting we could perform precomputation to quickly approximate the solution to (5.8) online.
5.1.1 Precomputation for Random Walker Image Segmentation

Precomputation without Prior Probabilities

Grady and Sinop [126] presented a scheme for precomputing data offline, before seeds are given, to speed up the solving of (5.8) online, which we review in this section. We first note that their precomputation scheme assumes the only constraints are seeds, so $\Gamma = 0$. This limitation will be addressed in the next section. Their precomputation scheme is based on an eigenvector decomposition of the Laplacian:

$$L = Q\Lambda Q^\top,$$

(5.9)

where the columns of $Q$ are the eigenvectors and $\Lambda$ is a diagonal matrix of the eigenvalues, ordered from smallest to largest. The Laplacian is known to be positive semi-definite, with one zero eigenvalue corresponding to the normalized constant vector $g = 1/\sqrt{N}$. We now show how to use these eigenvectors to quickly find an approximate solution to (5.8). $E$ is defined as the pseudo-inverse of $L$, given by

$$E = Q\Lambda^{-1}Q^\top.$$

(5.10)

Writing $E$ and $g$ in block notation,

$$E = \begin{bmatrix} E_s & R \\ R^\top & E_n \end{bmatrix}$$

(5.11)

$$g = \begin{bmatrix} g_s \\ g_n \end{bmatrix}.$$  

(5.13)

In practice it is infeasible to calculate all $N$ eigenvectors of $L$, as $Q$ would have $\sim N^2$ non-zero elements, which would be too large of a matrix from both a computational and memory perspective, particularly for volumetric images. Thus, we only calculate the $m \ll N$ eigenvectors, corresponding to the $m$ smallest eigenvalues, since the smallest eigenvalues give the best approximation to $E$. Since $L$ has $g$ as its one zero eigenvector,

$$EL = (I - gg^T),$$

(5.14)
where $I$ is the identity matrix of size $N \times N$. For the remainder of this chapter, we use subscripts to denote the size of identity matrices that are not $N \times N$. We now define the unknown $F = LU$:

\[
\begin{align*}
F &= \begin{bmatrix} F_s \\ F_n \end{bmatrix} \\
F_s &= L_s U_s + B U_n \\
F_n &= B^T U_s + L_n U_n \\
&= 0, \text{ from (5.8) with } \Gamma = 0.
\end{align*}
\] (5.15)

Left multiplying by $E$ gives

\[
\begin{align*}
EF &= ELU = (I - gg^T)U \\
R^T F_s &= U_n - g_n g_n^T U_n - g_n g_s U_s
\end{align*}
\] (5.19)

where the last step takes only the non-seeded rows. This equation has two unknowns, $U_n$ and $F_s$. The strategy is to now remove $U_n$ from (5.20) and solve for $F_s$. We first define $\alpha = U_n^T g_n \in \mathbb{R}^K$, then we right multiply (5.20) by $B$ and subtract it from (5.16), giving

\[
(I_s - B R^T) F_s = L_s U_s + B g_n (\alpha^T + g_s U_s),
\] (5.21)

We rewrite $F_s = \hat{F}_s + \bar{F}_s \alpha^T$, where $\hat{F}_s$ is an $N \times K$ matrix and $\bar{F}_s$ is an $N \times 1$ vector. This lets us divide (5.21) into two equations:

\[
\begin{align*}
(I_s - B R^T) \hat{F}_s &= L_s U_s + B g_n g_s U_s \\
(I_s - B R^T) \bar{F}_s &= B g_n
\end{align*}
\] (5.22, 5.23)

Both these systems of equations are of size $S \ll N$, and can be solved efficiently. We solve for $\alpha$ from (5.16) to get

\[
\begin{align*}
g^T F &= g^T LU \\
&= 0
\end{align*}
\] (5.24, 5.25)

\[
\begin{align*}
g_n^T \bar{F}_s \alpha^T &= -g_s^T \hat{F}_s \\
\alpha &= -\frac{g_s^T \hat{F}_s}{g_n^T \bar{F}_s}
\end{align*}
\] (5.26, 5.27)
Finally, (5.20) gives

$$U_n = R^\top F_s + g_n(\alpha - g_s U_s).$$  \hspace{1cm} (5.28)

Thus, Grady and Sinop reduce solving (5.8) with $\Gamma = 0$, an equation of size $(N - S)$, to the much quicker task of solving two equations of size $S$, (5.22) and (5.23). However when priors are added to the RW formulation ($\Gamma \neq 0$), this precomputation scheme is no longer applicable, as is discussed in the next section, where we introduce a new precomputation scheme that allows the inclusion of priors.

**Precomputation with Prior Probabilities**

When priors probabilities are included in RWIS ($\Gamma \neq 0$), the definition of $F$ becomes:

$$F = (L + \Gamma)U$$  \hspace{1cm} (5.29)

$$F_s = L_s U_s + \Gamma_s U_s + B U_n$$  \hspace{1cm} (5.30)

$$F_n = B_n^\top U_s + L_n U_n + \Gamma_n U_n$$  \hspace{1cm} (5.31)

$$= \Gamma_n P_n , \text{ from (5.8)},$$  \hspace{1cm} (5.32)

and (5.20) becomes

$$EF = E(L + \Gamma)U = (I - gg^\top + E\Gamma)U$$  \hspace{1cm} (5.33)

$$R^\top F_s + E_n \Gamma_n P_n = (E_n \Gamma_n + I_n)U_n - g_n \alpha^\top.$$

(5.34)

In Sec. 5.1.1, we eliminated $U_n$ by multiplying (5.34) by $B$ and subtracting it from (5.30), but in (5.34), $U_n$ is already multiplied by $(E_n \Gamma_n + I_n)$, so this scheme will no longer work.

To address this issue, we derive a generalization of the precomputation scheme. To do this, we analyze what information is available offline when prior probabilities are included. We will not rely on the prior probabilities $P$ being available offline, as we want to allow $P$ to incorporate information from user input seeds. In order to combine precomputation and prior probabilities, we enforce the restriction on $\Gamma = \Gamma' + \gamma I$ that only the scalar $\gamma$ can be updated online. This allows spatially varying influence of the prior probabilities based on image information, and allows the overall trade-off between the seeds and the prior probabilities to be adjusted online. We update the precomputation scheme to perform the eigendecomposition:

$$L + \Gamma' = QAQ^\top.$$  \hspace{1cm} (5.35)
$E$ is updated to the pseudo-inverse of $(L + \Gamma' + \gamma I)$:

$$E = Q(\Lambda + \gamma I_m)^{-1}Q^\top. \quad (5.36)$$

The inverse in (5.36) is of a diagonal matrix of size $m$ and can be performed efficiently online. In block notation, $E$ is updated to

$$E = \begin{bmatrix} E_s & R \\ R^\top & E_n \end{bmatrix} \quad (5.37)$$

$$= \begin{bmatrix} Q_s(\Lambda + \gamma I_m)^{-1}Q_s^\top & Q_s(\Lambda + \gamma I_m)^{-1}Q_n^\top \\ Q_n(\Lambda + \gamma I_m)^{-1}Q_s^\top & Q_n(\Lambda + \gamma I_m)^{-1}Q_n^\top \end{bmatrix}. \quad (5.38)$$

The updated precomputation scheme has an added benefit: if $\Gamma'$ is not a zero matrix, $(L + \Gamma')$ is positive definite, and has no zero eigenvalues, so

$$E(L + \Gamma' + \gamma I) = I. \quad (5.39)$$

Multiplying $F$ from (5.29) by this updated $E$ gives

$$EF = E(L + \Gamma)U = U \quad (5.40)$$

$$R^\top F_s + E_n \Gamma_n P_n = U_n, \quad (5.41)$$

where the last step takes only the non-seeded rows. To remove $U_n$, we right multiply (5.41) by $B$ and subtract it from (5.30), giving

$$(I_s - BR^\top)F_s = (L_s + \Gamma_s)U_s + BE_n\Gamma_n P_n, \quad (5.42)$$

which has $F_s$ as the only unknown, and can thus be found by solving the systems of equations. These systems are only of size $S \ll N$, and can be solved efficiently. Finally, we recover the RW probabilities $U_n$ using (5.20).

**Extended Precomputation**

While RWIS with precomputation avoids solving the $K$ linear systems of equations of size $\sim N$ in (5.8), it will still require multiplications between matrices of size $\sim N$, since it calculates the $(N - S) \times K$ matrix $U_n$. In this section, we present a more detailed algorithm for RWIS with precomputation, designed to minimize the number of multiplications between large matrices.
Algorithm 3 RWIS with Precomputation:

**Offline:**
1: Calculate $Q$ and $\Lambda$ from $(L + \Gamma')$

**Online:**
2: Solve for $F_s$:
   \[(I_s - BR^\top)F_s = (L_s + \Gamma_s)U_s + BE_n\Gamma_nP_n\]
3: $U_n = R^\top F_s + E_n\Gamma_nP_n$

Algorithm 3 lists the offline and online steps in RWIS with precomputation. Steps 2 and 3 both require multiplications between matrices of size $(N - S)$, which are expensive online operations. To minimize these multiplications, we note that while matrices with subscript $n$ are not known offline, their “full” analogues (without the seeded rows/columns removed) are available offline. We can avoid performing certain matrix multiplications online by performing other matrix multiplications offline and then taking the non-seeded rows and columns of the resulting matrices online.

Referring to (5.37), we can rewrite
\[
BR^\top = BQ_n(\Lambda + \gamma I_m)^{-1}Q_s^\top \quad (5.43)
\]
\[
BE_n = BQ_n(\Lambda + \gamma I_m)^{-1}Q_n^\top \quad (5.44)
\]
\[
E_n\Gamma_n = Q_n(\Lambda + \gamma I_m)^{-1}Q_n^\top (\Gamma'_n + \gamma I_n) \quad (5.45)
\]
\[
R^\top F_s + E_n\Gamma_nP_n = Q_n(\Lambda + \gamma I_m)^{-1}(Q_s^\top F_s + Q_n^\top (\Gamma'_n + \gamma I_n)P_n) \quad (5.46)
\]

The expensive matrix multiplications are $BQ_n$ and $Q_n^\top (\Gamma'_n + \gamma I_n)$, but we can precompute matrices $\Pi^1$ and $\Pi^2$ to avoid performing the multiplications online:

\[
\Pi^1 = LQ = \begin{bmatrix} L_sQ_s + BQ_n \\ B^\top L_s + L_nQ_n \end{bmatrix} \quad (5.47)
\]
\[
\Pi^2 = \Gamma'Q = \begin{bmatrix} \Gamma'_sQ_s \\ \Gamma'_nQ_n \end{bmatrix} \quad (5.48)
\]
\[
BQ_n = \Pi^1_s - L_sQ_s \quad (5.49)
\]
\[
Q_n^\top (\Gamma'_n + \gamma I_n) = (\Pi^2_n + \gamma Q_n)^\top \quad (5.50)
\]

The full process is given in Algorithm 4. Step 5 requires a multiplication between matrices of size $m \times (N - S)$ and $(N - S) \times K$. Unfortunately this cannot be avoided if $P$ is not available offline. Step 7 requires a multiplication between matrices of size $(N - S) \times m$ and $m \times m$, which is the only other online operation requiring $O(N)$ time.
Algorithm 4 RWIS with Extended Precomputation:

**Offline:**
1. Calculate $Q$ and $\Lambda$ from $(L + \Gamma)$
2. $\Pi^1 = LQ$
3. $\Pi^2 = \Gamma^t Q$

**Online:**
4. $\Pi^3 = (\Pi^1_s - L_s Q_s)(\Lambda + \gamma I_m)^{-1}$, an $S \times m$ matrix
5. $\Pi^4 = (\Pi^2_n + \gamma Q_n)^t P_n$, an $m \times K$ matrix
6. Solve for $F_s$:
   
   $$(I_s - \Pi^3 Q_s^t) F_s = (L_s + \Gamma_s) U_s + \Pi^3 \Pi^4$$
7. $U_n = Q_n(\Lambda + \gamma I_m)^{-1}(Q_s^t F_s + \Pi^4)$

The RW algorithm ran using this precomputation technique will be referred to as “fastRWIS”, and without as “basic RWIS”. Unless otherwise stated, we use Algorithm 4 for fastRWIS.

**Results: fastRWIS with Priors**

In this section, we show that using our precomputed data makes the RWIS online phase fast enough for interactive segmentation without compromising accuracy. Therefore, we present results showing our high speed gains on real 2D and 3D data while maintaining negligible (and controlled) reduction in accuracy. In this section, we measure accuracy using the Dice similarity coefficient (DSC) between basic RWIS results and fastRWIS. Typically, segmentation accuracy is measured using the DSC between automatic segmentation results and manual results; we do this in subsequent sections, but here we focus on the segmentation differences resulting from the approximations made by fastRWIS. We note that the speed increase from fastRWIS allows much quicker seed editing and thus will translate to much improved accuracy per time spent by user.

The accuracy of the segmentations generated by our fastRWIS algorithm are evaluated by their similarity to the segmentations generated by basic RWIS with priors; the accuracy of basic RWIS is well justified in other works [123]. We note that the speed and accuracy of our algorithms depend on the image only through $m$. We leave analytical methods for finding optimal $m$ to the next section, here we use $m \in [40, 160]$, empirically found to be enough for 2D images, and we needed no more than $m = 350$ for larger 3D images. All experiments in this section were performed 100 times.

Tests were performed on the 2D blood cell image in Fig. 5.1 of size $N = 265 \times 272 \approx$
Figure 5.1: Comparison of results with and without precomputation for segmentation using priors on an image of size $N \approx 72,000$ pixels. For $m$, the number of eigenvectors used, we report the DSC between results from basic RWIS with priors and fast RWIS, $t_{on}$, the online time taken, and $t_{off}$, the offline time taken. We note that we are only concerned with $t_{on}$ and with $m = 80$ our method achieves excellent results in less than a fifth of the time taken when not using precomputation. Red and green correspond to different region boundaries.
72,000 pixels with $\beta = 50$, $\Gamma = (0.01) \cdot I$, and two regions, where one region was divided into multiple disconnected sections and seeds were only put in one of these sections. These segmentation times do not include calculating the priors, an efficient step which was performed online, and is similar in all cases. The priors were calculated by fitting a Gaussian to the intensity values of the seeds for each region and normalizing. Fig. 5.1 shows the DSC and the average runtimes in seconds for both the online and offline phases of fastRWIS for different values of $m$ and compares the results to basic RWIS with priors, showing excellent speedup and minimal accuracy lost. Fig. 5.2 compares the runtimes of the different methods for different resolutions of the blood cell image in Fig. 5.1, again showing our precomputation gives excellent speedup.

As seen in Fig. 5.1, even larger values of $m$ provide online runtimes much faster than can be achieved without precomputation, so choosing $m$ large enough to guarantee accuracy is our prime concern. While the next section contains experiments regarding image and seed dependent ways to choose $m$, in this section we present results exploring the relationship between the amount of noise in an image and how large $m$ must be to achieve a certain level of accuracy. The pixel intensities in our blood cell image range from 0 to 1 and various levels of Gaussian noise with standard deviations $\sigma \in [0, 1]$ were added to the 2D image in Fig. 5.1. From Fig. 5.3 we see that fastRWIS still provides accurate segmentations for small amounts of noise, up to $\sigma = 0.2$, (with DSC > 0.95) if a large enough $m$ is used. As the noise increases to $\sigma = 0.7$, DSC decreases. We can see the same trend in Fig. 5.4, where
Figure 5.3: Effect of $m$ and noise on segmentation accuracy. Compares the DSC between the segmentations found using basic RWIS and fastRWIS. Results are shown for two levels of noise and for multiple numbers of eigenvectors.

Figure 5.4: Effect of $m$ and noise on segmentation accuracy. Shows the DSC between the segmentations at varying levels of noise with $m = 200$ eigenvectors and 20 trials for each level of noise. We see that large enough $m$ lets us account for reasonable amounts of noise.
CHAPTER 5. PRECOMPUTATION

(a) Without Precomputation  (b) With Precomputation

Figure 5.5: The bones of a knee segmented with $S = 100$ seeds in one of the bones. (a) was found using basic RWIS in 40.5 seconds and (b) was found using fastRWIS with $m = 350$ in 1.56 seconds. DSC between the two is 0.975.

$m = 200$ eigenvectors are used in the precomputation and the noise ranges from $\sigma = 0$ to 1. We note that fastRWIS can tolerate reasonable amounts of noise and still provide DSC close to 1.

Tests were performed on a 3D CT image of the knee in Fig. 5.5 of size $N = 55 \times 55 \times 36 \approx 109,000$ voxels, a 26-connected image graph, and two regions, bone and non-bone. The bone region consists of 3 disconnected subregions (the femur, tibia, and patella). We tested the algorithms using priors by segmenting all the bones but placing seeds only in the tibia. We used basic RWIS and fastRWIS with $\beta = 100$ and $\gamma = 0.01$ and compared their average runtimes and the DSCs of their resulting segmentations. The average runtime of RWIS was about 40.5 seconds, and when $m = 350$ eigenvectors are used, the average runtime of fastRWIS was about 1.56 seconds. The DSC between RWIS's segmentation and fastRWIS's segmentation was 0.975. Thus our fastRWIS method achieved a speedup of 25 times over basic RWIS while maintaining excellent accuracy.

5.1.2 Adaptive Precomputation for Segmentation

While fastRWIS provides an excellent speed increase over basic RWIS with minimal loss in accuracy [10,126], as seen in the previous section’s results, some questions still remain:

1. How large should $m$ (the number of eigenvectors) be? A larger $m$ leads to greater accuracy, but greater run-time both offline and online, since the online phase involves
Figure 5.6: A slice of a thigh MRI and several corresponding eigenvectors. The eigenvectors correspond to relaxed normalized cuts of the image, and thresholding them would result in a segmentation of certain structures, albeit perhaps not a useful one, as no user guidance has yet been given.

multiplications between matrices of size $N \times m$.

2. What if the Laplacian is not known offline? Particularly, what if one wants a different value for $\beta$ in (5.2)?

So far, justification of the above precomputation technique is largely algebraic. In order to answer these questions, we provide a more intuitive interpretation of the eigenvectors $Q$.

**Relationship to Normalized Cuts**

Grady and Sinop [126] showed their precomputation technique could be thought of as incorporating user interaction into the *normalized cuts* segmentation technique of Shi and Malik [286]. Given a weighted image graph, a normalized cut seeks to find a partition of the vertices $\Omega$ into two sets, $\Omega_A$ and $(\Omega \setminus \Omega_A)$ that minimizes the normalized cut value

$$\text{Ncut}(\Omega_A) = \frac{\text{cut}(\Omega_A)}{\text{assoc}(\Omega_A)} + \frac{\text{cut}(\Omega_A)}{\text{assoc}(\Omega \setminus \Omega_A)}$$ (5.51)

$$\text{cut}(\Omega_A) = \sum_{x \in \Omega_A, y \in \Omega \setminus \Omega_A} w_{xy}$$ (5.52)

$$\text{assoc}(\Omega_A) = \sum_{x \in \Omega_A, y \in \Omega} w_{xy}.$$ (5.53)

While finding the minimizing normalized cut is intractable, the relaxed solution (assigning real numbers instead of binary indicators to each vertex) is found by calculating the eigenvectors with smallest eigenvalues in the generalized eigenvector system (recall $L = D - W$):

$$Lv = \lambda Dv.$$ (5.54)
Defining $\hat{L} = D^{-1/2}LD^{-1/2}$ as the normalized Laplacian, (5.54) can be rewritten as a standard eigenvector system:

$$\hat{L}\hat{v} = \lambda \hat{v},$$

(5.55)

where $\hat{v} = D^{1/2}v$. Note that since the eigenvectors are orthogonal, they tend to represent complimentary cuts, representing different structures in the image (Fig. 5.6).

To complete the connection, we note that the RW algorithm can be formulated using the normalized Laplacian $\hat{L}$ instead of $L$ by defining $\hat{U} = D^{1/2}U$ and $\hat{P} = D^{1/2}P$, substituting these new variables into the RW equation (5.8), solving for $\hat{U}$, and then recovering $U$. This results in the precomputation scheme from Sec. 5.1.1 calculating, in (5.9), the eigenvectors of $\hat{L}$ instead of $L$, which has been advocated because the normalized Laplacian tends to have a better behaved spectrum than the unnormalized Laplacian [71, 126]. In this formulation, the columns of $Q$ correspond to relaxed normalized cuts of the image. In the subsequent sections, we assume the normalized Laplacian is used, but maintain the use of $U$, $P$, and $L$ for consistency, explicitly noting any differences that would arise when using the normalized instead of the unnormalized Laplacian.

Shi and Malik explore several techniques for using the eigenvectors to segment an image [286], such as thresholding the eigenvectors. In the subsequent section, however, we only require the intuition that fastRWIS is approximating the original image with its most prominent structures as determined by normalized cuts, and, as seen in Algorithm 4, the segmentations are linear combinations of the eigenvectors:

$$U_n = Q_n \left( (\Lambda + \gamma I_m)^{-1} (Q_s^T F_s + \Pi^4) \right).$$

(5.56)

**Determining the Number of Eigenvectors: User Seeds**

Computing eigenvectors of the Laplacian is expensive, so there may be a limit to the number that can be computed offline, though this limit depends on factors such as the amount of computation power available and the throughput of images. More importantly, the online runtime scales linearly with $m$, so with large enough $m$ fastRWIS would be more computationally expensive than basic RWIS. Thus, we focus on choosing an appropriate number of eigenvectors online, and assume at least that many are computed offline.

The accuracy of fastRWIS is characterized by how close the solution it produces is to the exact RW solution (found from (5.8)), in terms of the DSC. However, the relationship
between accuracy and $m$ for any user input seeds is difficult to characterize. For example, in fastRWIS, if the user input seeds happen to correspond well to the image structures represented by the first few eigenvectors, a very small $m$ should give good accuracy. However, if a user is trying to delineate more complex structures, a large $m$ may be required before these structures are represented well by the eigenvectors. Below we develop a strategy for efficiently determining the appropriate number of eigenvectors to use online, after seeds are input.

From (5.56), each column of the segmentation $U_n = [u_{n1}, \ldots, u_{nK}]$ is a linear combination of the columns of $Q_n$. Recall $U_s = [u_{s1}, \ldots, u_{sK}]$ is an $S \times K$ matrix of 0's and 1's, with a single 1 in each row, representing the probabilities of the seeded voxels. Intuitively, since the unseeded voxels are derived from the seeds (see (5.8)), if no linear combination of the eigenvectors gives a segmentation for the seeded pixels similar to the labeling dictated by $U_s$, we conclude the segmentation specified by these seeds cannot be represented well by the current set of eigenvectors.

To make this rigorous, if we define $\sigma^k \in \mathbb{R}^K$ such that $u_{nk} = Q_n \sigma^k$, then we expect $u_{sk} \approx Q_s \sigma^k$. We note the constraint

$$\|u_k\| = \|Q \sigma^k\| = \|\sigma^k\| \leq \sqrt{N},$$

(5.57)
since the components of $\|u_k\|$ are at most 1. If the normalized Laplacian is being used, $\sqrt{N}$ should be replaced by $\sqrt{1^\top D 1}$. We now define the function $f(u^k_s)$ measuring how well the current eigenvectors can represent the seeds for label $k$:

$$f(u^k_s) = \min_{\|\sigma^k\| \leq \sqrt{N}} \|Q_s \sigma^k - u^k_s\|^2.$$  

(5.58)

The minimization in (5.58) is over a convex function with a convex constraint, and is of size $S \times m \ll N$, so can be solved efficiently. We calculate $f(u^k_s)$ for each $k$ to ensure each label is represented well by the eigenvectors, and if any are above a threshold $f_{\text{max}}$, we add more eigenvectors and repeat the process. $f_{\text{max}}$ controls the trade-off between speed and accuracy, and is thus application dependent, though since $f(u^k_s)/S$ is roughly a measure of the expected squared error at each voxel, $\sqrt{f_{\text{max}}/S}$ is roughly the expected absolute error. In our results, we use $f_{\text{max}} = S/100$. 

*CHAPTER 5. PRECOMPUTATION* 115
Updating the Regularization

The RW algorithm can be significantly affected by how the edge weights are calculated in (5.2), but the Laplacian must be fixed when precomputation is performed. Previously, if a user wanted to change the edge weights online, they had to abandon using fastRWIS. In this section, we propose a technique to update the precomputed data when the edge weights are changed. Specifically, we focus on changes in the key parameter $\beta$.

One potential technique is to precompute eigenvectors for multiple Laplacians with different edge weights, though there are limitations to this technique: there’s no guarantee the desired Laplacian (i.e. for a specific $\beta$) will be precomputed offline, and the offline computational burden could limit the number of Laplacians that can be used. Instead, we make the observation that when $\beta$ is increased or decreased, the relative ordering of the edge weights does not change, so small relaxed normalized cuts for one value of $\beta$ should also be small for other values of $\beta$. Thus, while the eigenvectors of the Laplacian are different for different values of $\beta$, they should correspond roughly to the same prominent image structures. This suggests we may be able to compute the eigenvectors for a certain $\beta'$ and then reuse them for other $\beta$’s. The key is updating the eigenvalues.

In normalized cuts, the eigenvector system (5.55) that is used to solve for the relaxed cuts is derived from the Rayleigh quotient minimization problem:

$$
\min_v \text{Ncut}(v) = \min_v \frac{v^\top \hat{L} v}{v^\top v}.
$$

(5.59)

$\text{Ncut}(v)$ is the value of the relaxed normalized cut $v$, and if $v$ is an eigenvector of $\hat{L}$ with corresponding eigenvalue $\lambda$, $\text{Ncut}(v) = \lambda$. Thus, the values on the diagonal of $\Lambda$ are not only the eigenvalues corresponding to the eigenvectors $Q$, but also their normalized cut values.

When $\beta$ is updated online, we construct the new normalized Laplacian, plug each column of $Q$ into the new $\text{Ncut}(\cdot)$ function, and replace the values on the diagonal of $\Lambda$ with the new normalized cut values. That is, for $Q = [q^1, \ldots, q^m]$, we define $\hat{\Lambda}$ as a $m \times m$ diagonal matrix with diagonal elements $\{\hat{\lambda}_1, \ldots, \hat{\lambda}_m\}$ given by

$$
\hat{\lambda}_i = q_i^\top \hat{L} q_i.
$$

(5.60)

$\hat{L}$ is then approximated by

$$
\hat{L} \approx Q \hat{\Lambda} Q^\top.
$$

(5.61)
This weights each column of $Q$ by how well it “cuts” the updated image graph, so columns that no longer correspond to good cuts will be largely ignored. While $Q$ and $\hat{\Lambda}$ can no longer be considered eigenvectors and eigenvalues, their use in fastRWIS does not otherwise change. While this approximation is expected to get worse for $\beta$ further from $\beta'$, we mitigate this by precomputing eigenvectors for several different $\beta'$s and using (5.59) to “interpolate” between them.

**Results: Adaptive fastRWIS**

In this section, we test the effectiveness of the techniques presented in Sec. 5.1.2. Our evaluations in this section focus on measuring the loss in accuracy when comparing both basic RWIS and fastRWIS to manual segmentations using the DSC. Ideally, only a small fraction of the overall accuracy will be lost when using fastRWIS.

We use two 2D images and a data set of 40 volumetric images. The 2D images consist of a $256 \times 256$ CT cardiac image slice, manually segmented into 4 regions, and the $265 \times 272$ blood cell image, manually segmented into 2 regions (blood cell and background). 2D data allows a large number of tests with different seed locations. The cardiac image is used because it provides a multi-label segmentation and the blood cell image is used because it requires prior probabilities calculated from the user seeds to segment all of the cells (Fig. 5.7).

The volumetric data set consists of the 40 ($250 \times 250 \times 40$) MRI images of thighs from Chapter 4, manually segmented into 16 regions. We combine some regions to simplify the segmentation problem, resulting in a 5-label segmentation (muscle, fat, cortical bone, bone marrow, and background, see Fig. 5.7, Fig. 5.13).

We use seeds generated automatically using the manual segmentations. For the cardiac image slice, we randomly place 10 seeds in each region, and with the blood cell image, we randomly place 10 seeds inside a single blood cell and 10 seeds nearby in the background. We generate 100 different sets of seeds for each image, with the single blood cell chosen randomly each time. For the thigh images, we randomly place 40 seeds each in the muscle, fat, and background regions, and 20 seeds each in the two bone regions. We generate 5 sets of seeds each for the 40 images. All of the segmentation results in this section are averaged over each set of seeds. Unless otherwise stated, we use $\beta = 50$, $\Gamma = (0.001) \cdot I$ for cardiac and thigh images, and $\Gamma = (0.01) \cdot I$ blood cell images. The prior probabilities $P$ are calculated by fitting a Gaussian to the intensity values of the seeds for each region and normalizing.
Figure 5.7: An example of the segmentations achieved by basic RWIS and fastRWIS with eigenvectors adapted to the seeds. Regions are outlined in color. fastRWIS takes only about 20% of time as basic RWIS (Fig. 5.8).
CHAPTER 5. PRECOMPUTATION

Figure 5.8: A comparison of run-time (as a fraction of basic RWIS) vs. accuracy for basic RWIS, fastRWIS with fixed numbers of eigenvectors, and fastRWIS with eigenvectors adapted automatically to the seeds. Our adaptive approach achieves excellent accuracy and run-time while automatically choosing the number of eigenvectors to use. See Table 5.1 for more details.

Determining the Number of Eigenvectors: First, we evaluate our technique from Sec. 5.1.2 for choosing the number of eigenvectors $m$ based on the seed locations, with $f_{\text{max}} = S/100$ (see (5.58)). Our goal is to show we can adaptively choose $m$ and achieve results comparable to the best fixed $m$. For the 2D images, we precompute 160 eigenvectors, and run RWIS and fastRWIS using $m \in \{60, 80, \ldots, 160\}$. For the 3D thigh images, we precompute 800 eigenvectors, and run RWIS and fastRWIS using $m \in \{300, 400, \ldots, 800\}$. We then run fastRWIS with $m$ chosen using our adaptive technique for each image. The accuracy and run-time of each segmentation algorithm is seen in Fig. 5.8, with example segmentations shown in Fig. 5.7.

We see that our technique for choosing $m$ online provides accuracy comparable to using all the eigenvectors while running, on average, significantly faster. We emphasize that while each image seems to have an “optimal” number of eigenvectors which gives similar results to our technique in terms of run-time and accuracy, this number is not known ahead of time and is different for each image class ($\sim 120$ for cardiac, $\sim 100$ for blood cells, and $\sim 700$ for thigh). Our technique requires no prior knowledge to achieve this accuracy, and further, provides slightly lower run-time than any fixed number of eigenvectors, due to cases where fewer eigenvectors are used.
Table 5.1: A table of the results from Sec. 5.8. Adaptive fastRWIS chooses the number of eigenvectors to use online based on the user seeds. The results reported for fastRWIS are for a fixed number of eigenvectors $m$, chosen separately for each image to give the run-time and accuracy closest to its adaptive counterpart. Accuracy is reported using DSC.

<table>
<thead>
<tr>
<th>Data</th>
<th>Algorithm</th>
<th>Accuracy (DSC)</th>
<th>Run-Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardiac</td>
<td>RWIS</td>
<td>0.943 ± 0.027</td>
<td>0.983 ± 0.012</td>
</tr>
<tr>
<td></td>
<td>fastRWIS</td>
<td>0.942 ± 0.031</td>
<td>0.203 ± 0.003</td>
</tr>
<tr>
<td></td>
<td>Adaptive fastRWIS</td>
<td>0.942 ± 0.034</td>
<td>0.178 ± 0.006</td>
</tr>
<tr>
<td>Blood Cell</td>
<td>RWIS</td>
<td>0.986 ± 0.012</td>
<td>1.07 ± 0.014</td>
</tr>
<tr>
<td></td>
<td>fastRWIS</td>
<td>0.984 ± 0.015</td>
<td>0.24 ± 0.004</td>
</tr>
<tr>
<td></td>
<td>Adaptive fastRWIS</td>
<td>0.983 ± 0.016</td>
<td>0.19 ± 0.007</td>
</tr>
<tr>
<td>Thigh</td>
<td>RWIS</td>
<td>0.927 ± 0.034</td>
<td>289.2 ± 10.3</td>
</tr>
<tr>
<td></td>
<td>fastRWIS</td>
<td>0.917 ± 0.041</td>
<td>42.0 ± 2.3</td>
</tr>
<tr>
<td></td>
<td>Adaptive fastRWIS</td>
<td>0.919 ± 0.040</td>
<td>43.6 ± 3.4</td>
</tr>
</tbody>
</table>

Figure 5.9: A comparison of fastRWIS accuracy when $\beta$ is known offline versus when $\beta$ is changed online and $\Lambda$ is updated using our technique. The eigenvectors from the Laplacian with $\beta' = 50$ are used as the base eigenvectors when $\beta$ is not known. The difference in DSC between the results of the two methods is reported. As expected, the DSC is higher when the Laplacian is known offline, though only slightly higher for a large range of values, particularly for the blood cell image. This indicates our technique allows $\beta$ to be adjusted online by at least a factor of 2 while sacrificing only minimal accuracy.
CHAPTER 5. PRECOMPUTATION

Updating the Laplacian: We evaluate our technique from Sec. 5.1.2 (updating the pre-computed values when the Laplacian edge weights are changed) to show the precomputed data can still be used to speed RWIS when $\beta$ is changed online, with minimal loss in accuracy. We do this by precomputing eigenvectors/values for an offline Laplacian constructed with $\beta' = 50$, then running fastRWIS using an online Laplacian with $\beta \in [25, 100]$ and a $\Lambda$ updated using (5.60). As a baseline, we compare the results to fastRWIS run using eigenvector/values calculated from the online Laplacian directly. We precompute 320 eigenvectors for the 2D images and 1600 for the thigh images. The accuracy of each technique is shown in Fig. 5.9. We see that when $\beta$ is not known offline, our technique for updating $\Lambda$ achieves very similar accuracy compared to when $\beta$ is known offline.

5.2 Fast Random Walker Image Registration

The RW algorithm has also been applied to image registration (RWIR). In RWIR, the general RW formulation remains largely the same, except each of the labels correspond to a displacement vector from a discrete pre-defined set. RWIR provides a globally optimal probabilistic deformation, from which uncertainty information can be calculated. Seeds can be provided to RWIR in the form of landmark points, but RWIR often relies more on prior probabilities for the displacements derived from image similarity measures, due to the difficulties in providing numerous landmarks in volumetric images.

One drawback of RWIR (and other discrete registration techniques) is that the predefined set of displacement vectors needs to be quite large, particularly in 3D tasks, in order to achieve adequate spatial resolution, leading to high computational costs. The requirement for a large label set has been mitigated by iterative approaches designed to incrementally update an initially sparse label set [72,140,317], though such iterative approaches often sacrifice global optimality. Alternatively, a sparser, image-adaptive discretization of the image domain can be used [246,315] instead of the standard uniform voxel-based discretization, so fewer spatial locations need to be solved for.

In this section, we extend the precomputation technique from Sec. 5.1.1 to RWIR, providing a technique to increase RWIR efficiency complementary to the techniques mentioned above. In registration, one of the images is usually available before the others (e.g. when registering images taken of the same patient months apart or when registering a new image to an existing atlas). We define “offline” as the time when only one of the images is
available, and perform precomputation on this image. Even in registration tasks without repeated user interaction, registration algorithms can be very time consuming, so increasing the speed of RWIR is beneficial. Further, while some registration parameters must be chosen before precomputation, several important parameters (e.g. image similarity, regularization strength) can be set afterwards, allowing multiple registrations with different parameter settings to be evaluated quickly.

**Random Walker Image Registration**

To register two images, \( J_1, J_2 : \Omega \rightarrow \mathbb{R} \), where \( \Omega \subset \mathbb{R}^d \) and \( d \) is the dimensionality of the images, we find a spatial transformation \( T : \Omega \rightarrow \Omega \) such that \( J_1 \circ T \) (that is, \( J_1 \) composed with \( T \)) is aligned to \( J_2 \). Not all transformations are equally likely, however, so some regularization is imposed on \( T \) to ensure a feasible transformation.

In RWIR, we work in the discrete image domain, so \( \Omega \) consists of \( N \) pixels, and we represent transformations using a discrete set of \( K \) displacements, \( \{v^1, \ldots, v^K\} \), where \( v^k \in \mathbb{R}^D \). \( U \), the \( N \times K \) matrix of RW probabilities, provides a distribution over these displacements for each pixel. As with RWIS, an image graph is constructed over the moving image with edge weights given by (5.2) and the corresponding graph Laplacian is used to regularize the registration. Assuming no landmarks, the RWIR probabilities are given by

\[
(L + \Gamma)U = \Gamma P .
\] (5.62)

In RWIR, the prior probabilities \( P = [p^1, \ldots, p^K]^\top \) are based on an image similarity function:

\[
p^k = \frac{1}{Z_x} \mathcal{D}(x, v_k, J_1, J_2) .
\] (5.63)

Here, \( Z_x \) is a normalization function ensuring the prior labels sum to one at each pixel. \( \mathcal{D} \) can include various image features, such as patch intensity differences, image gradients [130], or mutual information [348], and is largely application dependent. In this thesis, \( \mathcal{D} \) is a function of local patch differences between images. We note that when there are no user input seeds, \( \Gamma \) weights the trade-off between the priors and the regularization (through \( L \)), so we will often refer to \( \Gamma \) as controlling the regularization. Variable prior influence through \( \Gamma \) is useful for increasing regularization strength in homogeneous regions, where the prior labels \( p^k \) are not very informative.
Once the probabilistic labels have been calculated, they are used to construct a dense displacement field over the image; e.g. by taking the displacement corresponding to the maximum probability or the expected displacement [72]. This results in a globally optimal dense displacement field that can represent a wide range of transformations.

5.2.1 Precomputation for Registration

A major drawback of RWIR is that it is computationally expensive. The main bottleneck is solving the system of equations in (5.62), which is more time consuming than for RWIS since it must be solved for each of the $K$ displacement labels. To achieve adequate resolution, particularly for 3D registration, $K$ must be very large. As detailed above, several techniques have been proposed to increase the efficiency of RWIR by iteratively running the algorithm, starting with a sparse displacement label set and updating the label set after each iteration. These approaches have proven effective, but the iterative approach sacrifices global optimality, and there is no guarantee that the optimal displacements will be added to the label set.

In this section, we introduce the alternative, complementary approach of using precomputation to improve RWIR efficiency. The resulting algorithm will be referred to as “fastRWIR”, and RWIR without precomputation will be referred to as “basic RWIR”.

As in Sec. 5.1.1, we write $\Gamma = \Gamma' + \gamma I$, where $\Gamma'$ is spatially varying (e.g. based on local image variability) and known offline and $\gamma$ is spatially uniform, but can be updated online. We perform an eigendecomposition:

$$L + \Gamma' \approx Q\Lambda Q^\top,$$

except only calculating the $m$ smallest eigenvalues and corresponding eigenvectors. We can then approximate $U$ from (5.62) quickly using

$$U \approx Q(\Lambda + \gamma I_m)^{-1}Q^\top \Gamma P.$$  

$m$ controls the trade-off between accuracy and speed; for small enough $m$ (5.65) is much more efficient to compute than (5.62). We note this formulation is notationally simpler than fastRWIS due to the lack of seeds; if landmarks were included, the formulation would be the same as for fastRWIS.
CHAPTER 5. PRECOMPUTATION

Figure 5.10: The average distance between the displacements generated by RWIR and fastRWIR for different values of $k$ and the average times taken for both RWIR and fastRWIR. For reference, the average distance between the RWIR and the ground truth displacements was $2.24 \pm 0.05$ mm, while for $k = 1000$ the average distance between the RWIR and the fastRWIR displacements is much smaller at $0.07 \pm 0.06$ mm, indicating that the approximation used in fastRWIR will have minimal effect on the registration error. RWIR runs in $2390 \pm 1700$ seconds (shown in red as a baseline), whereas fastRWIR with $k = 1000$ runs in $795 \pm 253$ seconds.

Parameter Tuning

Properly chosen parameters are key to the success of many registration algorithms. The speedup provided by fastRWIR allows more registrations to be performed with different parameter settings in a given amount of time, increasing the robustness of the results to parameter tuning. However, fastRWIR requires certain parameter values be set at precomputation time, so it is important to note what parameters can be updated without affecting the precomputed matrices.

Clearly, the similarity term $P$ can be updated online, allowing for multiple similarity functions to be tested quickly. Another important parameter that can be set after precomputation is the strength of the priors, $\gamma$. fastRWIR can be performed quickly with various prior strengths, making the final deformation robust to under-regularization (e.g. if the transformation is irregular) or over-regularization (e.g. if the images are too dissimilar).
CHAPTER 5. PRECOMPUTATION

Results: fastRWIR

In this section, we provide results comparing our method, fastRWIR, to basic RWIR, demonstrating the considerable speed up it provides with minimal loss in accuracy. Our experiments are performed on 40 T1-MR volumetric brain images from the LONI dataset [161]. We focus on comparing basic RWIR deformations to fastRWIR deformations, and show their differences are small compared to the differences with the ground truth deformation, indicating fastRWIR exhibits a minimal loss in accuracy. We use a displacement label set of size $K \approx 5000 \ (17 \times 17 \times 17)$.

Synthetic Misalignment: In our first experiment, we applied known warps to the images and then attempted to recover these warps using RWIR and fastRWIR. We generated 10 warps for each image by randomly displacing B-spline control points, spaced 18 mm apart, where the displacements were sampled uniformly from vectors up to 6 mm in magnitude. We registered each image and its warped version using basic RWIR and fastRWIR with $m \in \{100, 200, \ldots, 1000\}$ precomputed eigenvectors. In Fig. 5.10, we evaluated the accuracy of fastRWIR by calculating the mean distance between the displacements generated at each voxel by basic RWIR and by fastRWIR. We compared these distances to the mean distance between the basic RWIR displacements and the ground truth displacements (i.e. the registration error) of $2.24 \pm 0.05 \ mm$. For $m = 1000$, fastRWIR generated similar results to basic RWIR, with an average distance between displacements of $0.07 \pm 0.06 \ mm$, only 3% of the registration error. For $m = 300$, the average distance between fastRWIR and basic RWIR is still only $0.46 \pm 0.10$, about one fifth of the error.

Fig. 5.10 also compares the time taken by RWIR and fastRWIR for different values of $m$. For $m = 1000$, fastRWIR achieved a roughly 3 times speedup, and for $m = 300$ it achieved a roughly 10 times speedup. Choosing an appropriate value for $m$ is addressed in Sec. 5.2.2.

Real Data: In our next experiment, we performed registration between each pair in the 40 images (randomly choosing which one to use as the moving image) using both RWIR and fastRWIR with $m = 1000$. Each image has 56 anatomical brain regions segmented, and we evaluated the accuracy of the registrations by comparing the Dice similarity coefficients (DSCs) between each of these regions in the pair of images before and after the registration is performed.

Fig. 5.11 shows a comparison of the DSCs for each of the brain regions before registration and after registration using RWIR and fastRWIR. Both registration methods typically
increase the DSCs, indicating they improve the correspondence between anatomical regions significantly. As expected, fastRWIR performs slightly worse than RWIR on average, though the improvement in DSC when using RWIR instead of fastRWIR is small (0.013 ± 0.005) compared to the improvement in DSC before and after registration (0.074 ± 0.041). Thus, the approximations introduced by fastRWIR reduce the increase in DSC from registration by less than a fifth. Despite only a small drop in accuracy, fastRWIR ran more than 3 times faster than RWIR (830 ± 478 seconds for fastRWIR vs. 2716 ± 1783 seconds for RWIR).

Fig. 5.12 provides a more detailed comparison of the algorithms, showing the increase in DSC when using RWIR instead of fastRWIR for each of the image pairs. For most regions, the difference in DSC is consistently close to 0, though a few regions appear to give fastRWIR trouble. This is likely due to the eigenvector approximation of the Laplacian of the image graph being poor in the neighborhood of these regions.

5.2.2 Adaptive Precomputation for Registration

fastRWIR provides an excellent speed increase with only a minimal loss in accuracy, as seen in the previous section, but some questions still remain:
Figure 5.12: The increase in DSC of each of the 56 segmented brain regions when registering with RWIR instead of fastRWIR with $m = 1000$. Some outliers above and below the current bounds have been excluded for readability.

1. How large should $m$ be? This question was addressed for fastRWIS in Sec. 5.1.2 based on user input seeds, but these are often not available or prominent in RWIR, so we require a technique based on the prior probabilities.

2. What if the graph connectivity changes? In order to make RWIR more efficient, techniques have been developed that use sparser graph structures when solving (5.8), and this structure may be updated online [315].

To answer these questions, we build on the normalized cut interpretation of the eigenvectors from Sec. 5.1.2

**Determining the Number of Eigenvectors: Prior Probabilities**

In fastRWIS, we determined how many eigenvectors to use by evaluating how well the seeds could be represented by the eigenvectors; we take a similar approach with the prior probabilities.

The prior probabilities $P = [p^1, \ldots, p^K]$ correspond to an unregularized probabilistic deformation, so determining how well they can be represented by the eigenvectors is fairly straightforward. Since the columns of $Q$ are orthonormal, we can project each of the columns
of $P$ onto their span and calculate the magnitude of the residual:

$$g(p_k) = \|p_k - QQ^\top p^k\|^2. \quad (5.66)$$

Since each label corresponds to a displacement vector, labels corresponding to similar displacement vectors often have similar probabilities, thus \((5.66)\) need not be evaluated for every $k$. To increase efficiency, we evaluate $g(p_k)$ for one out of every $r^d$ displacement labels, uniformly spaced, where $d$ is the image dimension. If any of them are above a threshold $g_{\text{max}}$, we add more eigenvectors. In this work, we use $r = 4$, so \((5.66)\) is evaluated for $(K/64)$ labels in volumetric registration. Similar to $f_{\text{max}}$, $g_{\text{max}}$ controls the trade-off between speed and accuracy, and $\sqrt{g_{\text{max}}/N}$ is roughly proportional to the voxel-wise error in the probabilities. In our results, we use $g_{\text{max}} = N/100$. Note that we save the residual $(p^k - QQ^\top p^k)$, so we are able to update \((5.66)\) efficiently when more eigenvectors are added by projecting the residual onto the new eigenvectors.

### Updating the Graph Topology

In RWIR, as with other discrete registration techniques, there has been recent work targeted at reducing the number of vertices in the image graph, and thus the computational cost of RWIR, by aggregating vertices into “super-vertices”. While this aggregation has often been done in an image agnostic way, using grids of predefined resolutions, recent image dependent online techniques have proven successful [222, 246, 315]. In general, vertex aggregation is performed by defining a new set of vertices $\bar{\Omega}$, $|\bar{\Omega}| = \bar{N} < N$ and a “projection function” $\eta(x, y)$, where $x \in \Omega$ and $y \in \bar{\Omega}$, encoding the influence of each super-vertex on vertex $x$ when propagating values from the sparser graph to the denser graph. The corresponding “aggregation function” is defined as $\bar{\eta}(x, y)$, encoding the influence of each vertex on super-vertex $y$ when propagating values from the denser graph to the sparser graph. These functions are normalized:

$$\sum_{y \in \bar{\Omega}} \eta(x, y) = 1, \quad \sum_{x \in \Omega} \bar{\eta}(x, y) = 1. \quad (5.67)$$

Once RW is solved on the aggregated graph, the solution is propagated back to the original vertices. Denoting probabilities assigned to $y \in \bar{\Omega}$ as $\bar{U}_y$, the corresponding probabilities for vertex $x$ are given by

$$U_x = \sum_{y \in \bar{\Omega}} \eta(x, y)\bar{U}_y. \quad (5.68)$$
Note that for a given \( x \), often \( \eta(x, y) = 0 \) for most \( y \), with vertices only dependent on a few (or one) nearby super-vertices.

Vertex aggregation performed online invalidates the eigenvectors precomputed from the original graph. We cannot apply the technique for updating \( \beta \) from Sec. 5.1.2 directly, since the number of vertices have changed, so the normalized cut values cannot be computed for the columns of \( Q \). Thus, we develop a technique for aggregating the columns of \( Q \) to the super-vertices.

Let \( \mathbf{q} = [q_1, \ldots, q_N]^\top \) be one of the columns of \( Q \), and let \( \bar{\mathbf{q}} = [ar{q}_1, \ldots, \bar{q}_N]^\top \) be the (undetermined) aggregation of \( \mathbf{q} \) to the super-vertices. The most straightforward technique for calculating \( \bar{\mathbf{q}} \) would be to use \( \bar{\eta}(\cdot, \cdot) \) directly:

\[
\bar{q}_y = \sum_{x \in \Omega} \bar{\eta}(x, y) q_x .
\] (5.69)

However, using (5.69) may not respect the property that \( \bar{\mathbf{q}} \) should have a small normalized cut value on the aggregate graph. A super-vertex \( y \) may “consume” some edges, if the vertices on both ends of the edge are assigned exclusively to \( y \). The influence of consumed edges should not be considered when calculating \( \bar{\mathbf{q}} \), as they have no effect on the normalized cut of the aggregate graph. Consider a vertex \( x \) with all of its edges consumed; \( q_x \) was based exclusively on those edges, and thus should not be considered when performing aggregation.

To account for this, instead of weighting components of \( \mathbf{q} \) just by \( \bar{\eta}(\cdot, \cdot) \), we also weight them by the local change in \( \eta(\cdot, \cdot) \), since the edges between vertices assigned to different super-vertices will still affect the normalized cut value:

\[
\Delta(x) = \sum_{x' \in N(x)} \sum_{y \in \Omega} |\eta(x, y) - \eta(x', y)| \]

\[
\bar{q}_y = \sum_{x \in \Omega} \Delta(x) \bar{\eta}(x, y) q_x .
\] (5.71)

We note that the vectors generated by (5.71) will not form an orthonormal set, but are made so using the Gram-Schmidt process. Combining the orthonormal vectors into a matrix \( \bar{Q} \) and then calculating their corresponding normalized cut values allows \( \text{fastRWIR} \) to be run on the aggregate graph with minimal overhead and loss in accuracy.

Results: Adaptive \( \text{fastRWIR} \)

We now test the effectiveness of the techniques presented in this section. Since \( \text{fastRWIR} \) is an approximation to the basic RWIR method, our evaluations focus on measuring the loss
in accuracy compared to the decrease in run-time when using fastRWIR with our adaptive techniques. Ideally, only a small fraction of the overall accuracy will be lost when using fastRW. Note that “accuracy” will be defined using mean overlap (MO) [161]. For two images segmented into $K$ foreground regions given by $\Omega^1_k$ and $\Omega^2_k$, $k \in \{1, \ldots, K\}$,

$$\text{MO} = 2 \frac{\sum_{k=1}^{K} |\Omega^1_k \cap \Omega^2_k|}{\sum_{k=1}^{K} |\Omega^1_k| + |\Omega^2_k|}. \quad (5.72)$$

**Setup:** We use 2 volumetric data sets, each segmented into many different regions, so the MO provides a useful measure of accuracy. The first data set consists of 40 ($250 \times 250 \times 40$) MR images of thighs (see Sec. 4.1.1), manually segmented into 16 regions. The second data set consists of 40 ($181 \times 181 \times 217$) T1-MR brain images from the LONI dataset [161] (used in the previous section), manually segmented into 56 regions, skull stripped, and with their intensity histograms normalized (Fig. 5.13).

We register each pair of images together. The local similarity between two voxels is
Figure 5.14: A comparison of run-time (as a fraction of basic RWIR) vs. accuracy for basic RWIR, fastRWIR with fixed numbers of eigenvectors, and fastRWIR with eigenvectors automatically adapted to the prior probabilities. Our adaptive approach achieves excellent accuracy and run-times while automatically choosing the number of eigenvectors to use. See Table 5.2 for more details.

evaluated using the sum of absolute intensity differences in a patch of size $5 \times 5 \times 5$. The prior probabilities $P$ for displacement vectors are calculated as the negative exponential of the squared local patch difference, normalized to sum to 1 at each voxel. We use $K \approx 4000 (21 \times 21 \times 9)$, $\beta = 50$, and $\Gamma = I$.

**Determining the Number of Eigenvectors:** We evaluate our technique for choosing the number of eigenvectors $m$ based on the prior probabilities (see (5.66)). Our goal is to show we can adaptively choose $m$ and achieve results comparable to the best fixed $m$. For each pair of images, we choose one randomly as the moving image and precompute 1500 eigenvectors. We run RWIR, fastRWIR with $m \in \{250, 500, \ldots, 1500\}$, and fastRWIR with $m$ chosen using our adaptive technique. The accuracy and the run-time of each registration algorithm is seen in Fig. 5.14, with example registrations show in Fig. 5.13. Similar to segmentation, our technique allows fastRWIR to achieve an average accuracy and run-time comparable to the best fixed number of eigenvectors, but without any prior knowledge.

**Aggregated Image Graphs:** In this section, we evaluate our technique for aggregating the eigenvectors into super-vertices, given by (5.71), showing the precomputed data can still be used to speed RWIR when the graph topology is changed online, with minimal loss in accuracy. We use $m = 2500$ eigenvectors for fastRWIR. We aggregate vertices based on their prior probabilities $P$ and spatial proximity [315] (Fig. 5.15), and thus the aggregation must be performed online, after the precomputation. For each pair of thigh and brain images,
we run 4 registration schemes: basic RWIR, fastRWIR with the eigenvectors aggregated using the naïve technique in (5.69), fastRWIR with the eigenvectors aggregated using our proposed technique in (5.71), and using eigenvectors calculated from the Laplacian of the aggregated graph directly (used only as a baseline, since the aggregated graph is not known offline). The accuracy and run-time of each technique is seen in Fig. 5.16. Our proposed technique performs significantly better than the naïve technique, and only slightly worse than the eigenvectors calculated directly from the aggregate graph.

5.2.3 Ensuring Anatomical Feasibility

To encourage anatomical feasibility, registration algorithms impose regularization on the deformations to encourage smoothness and topology preservation (TP). Often, imposing a spatially constant regularization sufficiently strong to ensure TP everywhere can result in
Figure 5.16: A comparison of RWIR techniques used on graphs with aggregated super-vertices. Accuracy and run-time are given in proportion to RWIR run on the full graph (Fig. 5.14 for absolute values). When run on the aggregated graph, RWIR achieves a significant speed-up with a minimal loss in accuracy, with further speed-up achieved by fastRWIR. The run times and MO values are given as a fraction of those achieved by the basic RWIS algorithm (Fig. 5.14 for absolute values). We note that using the eigenvectors from the aggregated graph (“□”) is not actually possible, since the aggregated graph is not known off-line, and is only included as a baseline.

over-smoothing and loss of accuracy due to inhomogeneities in the image. As mentioned previously, a useful feature of RWIR is that it seamlessly allows for spatially adaptive regularization (by changing the prior weights Γ), which have been shown to improve registration accuracy compared to a constant regularization weight [318]. However, even with spatially adaptive regularization, it is not clear a priori how much regularization is required in RWIR to ensure TP. One option to deal with this is time-consuming trial and error. Alternatively, post-processing techniques such as scaling and squaring [141] can convert a transformation to one that preserves topology, but such methods discard the uncertainty information provided by RWIR and the resulting transformation is no longer the global minimum of an energy function.

In this section, we introduce a precomputation technique that determines if a set of regularization weights for RWIR will result in a topology preserving transformation much faster than the time needed to run the full RWIR algorithm. We use our technique to
Table 5.2: A table of the results from Fig. 5.14. Adaptive fastRWIR chooses the number of eigenvectors to use online based on the prior probabilities. The results reported for fastRWIR are for a fixed number of eigenvectors \( m \), chosen separately for each image to give the run-time and accuracy closest to its adaptive counterpart. Accuracy is reported using MO.

iteratively increase regularization (i.e. reduce prior influence) in regions where topology is not preserved, avoiding over-smoothing in other regions.

Topology Preservation for Probabilistic Transformations

A main requirement for a spatial transformation \( T \) to be anatomically feasible is topology preservation, which primarily requires that the Jacobian (for brevity, we refer to the Jacobian determinant as simply the Jacobian) of \( T \), \( J(T) \), is positive everywhere. We adopt the convention that a probabilistic transformation is topology preserving if its expected transformation, with the expected displacement taken at each pixel, has a positive Jacobian everywhere. Recalling that \( U \) is an \( N \times K \) matrix of the RW displacement label probabilities, \( V \) is an \( d \times K \) matrix of the displacement vectors (where \( d \) is the image dimension), and defining \( X \) as an \( N \times d \) matrix of the pixel locations, then the expected transformation \( T \) is given by:

\[
T = X + UV^T. \tag{5.73}
\]

\( T \) is a discrete transformation, whose \( x^{th} \) row corresponds to the location pixel \( x \) is mapped to. \( J(T) \) is defined in the continuous case as a function of the derivatives of the components of \( T \). So, to discretize the Jacobian, we must choose a discrete approximation of the derivative operators. It has been shown that if all combinations of forward and backward difference operators along the \( d \) axis directions, \( \Delta_i^F \) and \( \Delta_i^B \), \( i \in \{1, \ldots, d\} \), are used to construct \( 2^d \) approximate Jacobians, and if they are all positive everywhere, then the continuous bi- or tri-linear interpolated version of \( T \) will also have positive Jacobian
everywhere [157]. We thus define the discrete Jacobian at pixel $x$, $J_x(T)$, to be the minimum of the $2^d$ approximate Jacobians.

We note that while topology preservation is often a necessary condition for anatomical feasibility, it is not always sufficient: transformations exhibiting excessive stretching may be anatomically infeasible. These conditions can be identified by pixels with very large or small (but still positive) Jacobians.

Our goal in the following sections is to choose spatially adaptive prior weights $\gamma_1, \ldots, \gamma_N$ in such a way that $U$, calculated from (5.62), corresponds to an anatomically feasible expected transformation $T$ with respect to its Jacobian values.

**Efficient Jacobian Precomputation**

In this section, we introduce a technique for precomputing the Jacobian of the expected transformation $J(T)$ without needing to calculate the probabilistic transformation $U$. This technique is orders of magnitude faster than the calculation of $U$ via (5.62), allowing regularization to be increased where needed, prior to running the RWIR algorithm.

In order to calculate $J(T)$, where $T = [t^1, \ldots, t^d]$, we must calculate $\Delta_i t^j$ for $i, j \in \{1, \ldots, d\}$, where $\Delta_i \in \{\Delta_i^F, \Delta_i^B\}$. Combining (5.62) and (5.73) gives

$$\Delta_i t^j = \Delta_i \left(X e^j + (L + \Gamma)^{-1} \Gamma PV^\top e^j\right),$$

(5.74)

where $e^j$ is the $j^{th}$ standard basis vector for $\mathbb{R}^d$. We define $b^j = \Gamma PV^\top e^j$ and $s^j = (L + \Gamma)^{-1} b^j$, and note that $s^j$ can be calculated without performing an expensive matrix inversion by solving the system of equations

$$(L + \Gamma) s^j = b^j.$$  

(5.75)

We note the similarity between (5.62) (used to solve for $U$) and (5.75). The difference is that in (5.75), the right hand side has 1 column, whereas in (5.62) the right hand side has $K$ columns, which can be on the order of hundreds or even thousands in 3D registration.

Solving for $s^j$ allows us to rewrite (5.74) as:

$$\Delta_i t^j = \Delta_i X e^j + \Delta_i s^j.$$  

(5.76)

Once (5.76) is solved for each $(i, j)$ pair, we can calculate $J(T)$. By repeating this process for each combination of forward and backward operators, we will determine if and where the discrete Jacobian of the expected transformation is negative.
The calculations in this section consist mainly of efficient multiplications between sparse matrices, with the exception being the need to solve (5.75) for each \( j \in \{1, \ldots, d\} \); this is the computational bottleneck. Since \( d \) is usually only 2 or 3, this bottleneck is still orders of magnitude faster than the full RWIR algorithm. This leads us to the strategy of repeatedly calculating \( J(T) \) with different settings for \( \Gamma \) until we determine how much regularization is needed at each pixel in order to ensure an anatomically feasible Jacobian, as we detail in the next section.

### Determining Regularization Weights

When selecting the prior weights \( \gamma_1, \ldots, \gamma_N \), we wish to only regularize (by reducing prior weights) as much as necessary to ensure TP. To do this, we could identify pixels with negative Jacobians using the technique from the previous section, decrease their prior weight slightly, and repeat. However, this technique may require a large number of iterations. In order to speed this process, we both decrease prior weights by larger amounts each iteration and decrease the prior influence on pixels in the neighborhood of a pixel with negative Jacobian. We decrease prior influence in a neighborhood because if neighboring regions in an image have conflicting prior probabilities, it might be the case that only pixels near the interface between these regions have negative Jacobians. If the interface pixels are highly regularized, the negative Jacobians may be “pushed out” to nearby pixels. Thus, increasing regularization for pixels near a pixel with negative Jacobian may greatly reduce the number of iterations, particularly for very negative Jacobians. An illustration of this concept in 1D is shown in Fig. 5.17.

To make this concept concrete, we define a function \( \phi_1 : \Omega \rightarrow \mathbb{R}^+ \) evaluating how much...
additional regularization each pixel \( x \) needs:

\[
\phi_1(x) = \tau_1 \max_{y \in \Omega} \left( \max(0, -J_y(T)) - \tau_2 \|x - y\|_2 \right),
\]

where \( \tau_1 \) and \( \tau_2 \) are positive scalar parameters. \( \tau_1 \) controls the amount that the regularization strengths are updated each iteration, and \( \tau_2 \) controls how large of an area is affected by a violation. Increasing \( \tau_1 \) and \( \tau_2 \) will increase the speed of convergence to a TP solution, but may also result in over-regularization if they are too large, so should be set based on the amount of time that can acceptably be spent precomputing the Jacobians. We simultaneously update the prior weights of each pixel \( x \) based on \( \phi_1 \):

\[
\gamma_x \leftarrow \gamma_x \cdot (1 + \phi_1(x))^{-1}.
\]

The update values for each pixel can be efficiently computed using dynamic programming (i.e. shortest path from multiple sources, the pixels with negative Jacobian). With the updated regularization weights, \( J(T) \) can be recalculated, and the update (5.78) performed again until all Jacobian values are positive. A topology preserving probabilistic transformation \( U \) can then be calculated using the updated prior weighting matrix \( \Gamma \).

While this method will ensure topology preservation, ground truth anatomical transformations will usually be fairly regular, having Jacobian values close 1 [263, 271]. The above algorithm can easily be generalized if a user wishes to put stricter constraints on the Jacobian. We define a second, more general function:

\[
\phi_2(x) = \tau_1 \max_{y \in \Omega} \left( \max \left( 0, \frac{\tau_3 - J_y(T)}{J \text{ Too Small}}, \frac{J_y(T) - \tau_4}{J \text{ Too Large}} \right) - \tau_2 \|x - y\|_2 \right),
\]

where \( \tau_3 \) and \( \tau_4 \) are the minimum and maximum Jacobians that we deem anatomically feasible. By replacing \( \phi_1 \) with \( \phi_2 \) in (5.78), we not only regularize away negative Jacobian values, but also Jacobian values that we deem anatomically infeasible. This targeted approach also avoids over-regularizing regions that already have feasible transformations under the current prior weights, allowing those regions to stay loyal to their prior probabilities. Of course, if no such prior knowledge of the Jacobian values is confidently known, \( \tau_3 = 0 \) and \( \tau_4 = \infty \) will still ensure TP.
CHAPTER 5. PRECOMPUTATION

<table>
<thead>
<tr>
<th>Priors</th>
<th>TRE (mm)</th>
<th>% of Pixels with $J \leq 0$</th>
<th>Run Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>2.83 ± 0.18</td>
<td>16.6 ± 1.0</td>
<td>–</td>
</tr>
<tr>
<td>RWIR-C</td>
<td>2.56 ± 0.24</td>
<td>1.67 ± 0.54</td>
<td>4957 ± 533</td>
</tr>
<tr>
<td>RWIR-U</td>
<td>1.70 ± 0.19</td>
<td>0.36 ± 0.11</td>
<td>5056 ± 561</td>
</tr>
<tr>
<td>RWIR-TP</td>
<td>1.64 ± 0.19</td>
<td>0.0 ± 0.0</td>
<td>5657 ± 577</td>
</tr>
</tbody>
</table>

Table 5.3: The results of running RWIR with different regularization matrices. RWIR-TP achieves less error than the other registrations while also ensuring topology preservation, while only requiring marginally longer computation time.

Results: Anatomically Feasible RWIR

In this section, we perform experiments to demonstrate the benefits of our spatially adaptive regularization scheme. We compare registration results from RWIR with and without our technique for choosing regularization weights. We also establish the correlation between uncertainty and error in probabilistic transformations, indicating that uncertainty information should not be discarded by post-processing techniques designed for non-probabilistic transformations.

Our experiments are performed on 40 MR volumetric thigh images of dimension 250 × 250 × 40, each segmented into 16 regions, including 11 different muscles (Fig. 4.1, Sec. 4.1.1). This data set was used because it contains regions with rich details leading to accurate priors (e.g. around the bone) and other regions that are largely devoid of detail (e.g. the homogeneous thigh muscles), so we expect well chosen spatially adaptive regularization weights to be important for accurate registration.

Synthetic Warpings: We applied known warps to each image and attempted to recover these warps using RWIR. We generated 5 warps for each image by randomly displacing B-spline control points, spaced 30 mm apart, where the displacements were sampled uniformly from vectors up to 8 mm in magnitude.

For each of the 200 image/warp pairs, we compare the results of RWIR run using 3 different prior weight matrices $\Gamma$. First, we use a spatially constant $\Gamma = \gamma I$, where $I$ is the identity matrix and $\gamma$ is a scalar. We refer to the results of this registration as RWIR-C. Second, we use uncertainty information from the prior probabilities $P$ to construct $\Gamma$ (based
Figure 5.18: A histogram of the Jacobian values for the ground truth, RWIR-U, and RWIR-TP transformations, averaged over each of the tests, and with a log-scale on the $y$-axis for readability. While the ground truth and RWIR-TP Jacobians are mostly contained in the range $[0.5, 1.5]$, the RWIR-U have Jacobian values far from 1 and even less than 0.

on an idea presented in previous works [182]):

$$\gamma_a = c \cdot \exp \left( \frac{-H(p_a)}{H_{\text{max}}} \right)$$  \hspace{1cm} (5.80)

$$H(p_a) = \sum_{k=1}^{K} \sum_{\ell=1}^{K} p_{a_k}^k p_{a_\ell}^\ell \|v^k - v^\ell\|^2,$$  \hspace{1cm} (5.81)

where $H$ is a measure of uncertainty found to correlate well with registration error [182], and $H_{\text{max}}$ is the maximum possible value for $H$. Using (5.81) results in stronger regularization for pixels with uncertain prior probabilities. We refer to the results of this registration as RWIR-U. Third, we use (5.78) with $\phi_2$ to iteratively update $\Gamma$, initialized using (5.81) and iterating 10 times, which was found to be sufficient to ensure topology preservation. We refer to the results of this registration as RWIR-TP. In both RWIR-C and RWIR-U, $\gamma$ was empirically set to minimize target registration error (TRE), the distance between the RW and GT displacements at each pixel. The parameters $\tau_1$, $\tau_2$, $\tau_3$, and $\tau_4$ were set to 0.3, 5, 0.5, and 1.5, respectively. These parameter were chosen empirically through tests performed on one of the 40 images, with $\tau_3$ and $\tau_4$ based on the synthetic warping parameters. We use $K \approx 12,500$ ($31 \times 31 \times 13$).
Figure 5.19: Mean TRE for different uncertainty percentiles. The mean TRE of the top $i\%$ most uncertain pixels was calculated, for $i = \{1, \ldots, 99\}$. The mean TRE consistently increases across percentiles, indicating uncertainty could be used to identify errors in a registration. Particularly, we see a sharp increase in TRE for the top 10% most uncertain pixels.

A comparison of the results achieved by the 3 registrations is shown in Table 5.3. The regularization used in RWIR-TP ensures a positive Jacobian, while only requiring about 10% longer to run. Further, RWIR-TP achieves less error than RWIR with other regularization matrices. Fig. 5.18, shows histograms of the Jacobian values for RWIR-U, RWIR-TP, and the ground truth displacements, showing the range of Jacobian values for RWIR-TP is much closer to the ground truth. Penalizing excessively small and large Jacobian values with RWIR-TP provides a more anatomically feasible transformation, resulting in decreased TRE.

To demonstrate the benefits of a probabilistic registration, for each test we calculated uncertainty values for each pixel by applying $H$ from (5.81) to the displacement probabilities generated by RWIR-U and RWIR-TP. We found a Pearson correlation coefficient between the uncertainty and the TRE of 0.45 for RWIR-U and 0.50. Fig. 5.19 demonstrates the relationship between uncertainty and TRE by taking the top $k\%$ most uncertain pixels, for $k = \{1, \ldots, 99\}$, and calculating their mean error. We see that error steadily increases for more uncertain pixels, and that the top 10% most uncertain pixels have significantly more error than an average pixel. Using post processing techniques to ensure topological invertibility would discard these uncertainty values.

**Real Data:** In this section, we demonstrate the usefulness of our method in a real medical imaging application by taking each pair of thigh images and registering them to each other, using both RWIR-U and RWIR-TP. We evaluate registration results using the mean overlap (MO) between the fixed image’s segmentation and the warped segmentation of the moving image (see (5.72)). We evaluate anatomical feasibility by looking at the number of pixels
Figure 5.20: An example on real data demonstrating how our technique increases regularization locally where necessary without over-regularizing other areas and affecting accuracy.

with negative Jacobian.

The results of this experiment are summarized in Table 5.4. We see that while the transformations resulting from RWIR-U and RWIR-TP have very similar MOs, the RWIR-U transformations have $3.57 \pm 0.78\%$ of pixels with negative Jacobian on average, and thus are not anatomically feasible transformations, whereas none of the RWIR-TP transformations have negative Jacobians, and thus represent anatomically feasible alignments (again, with only about 10\% increased run time). Fig. 5.20 demonstrates how locally updating the prior weights using RWIR-TP leads to a more accurate registration.

5.3 Chapter Summary

In this chapter, we explored the use of offline precomputation of data designed to increase the speed and accuracy of segmentation and registration algorithms run online. Specifically, in Sec. 5.1.1 and Sec. 5.2.1 we derived precomputation techniques for the popular RWIS and RWIR algorithms, and we presented results on real medical data demonstrating these
Table 5.4: MOs and % of pixels with negative Jacobian from registering pairs of images, averaged across all regions and tests. RWIR-TP and RWIR-U achieve similar MOs, but RWIR-TP provides topology preserving transformations.

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Priors P</th>
<th>RWIR-U</th>
<th>RWIR-TP</th>
</tr>
</thead>
<tbody>
<tr>
<td>MO</td>
<td>0.71 ± 0.12</td>
<td>0.81 ± 0.13</td>
<td>0.87 ± 0.06</td>
<td>0.88 ± 0.06</td>
</tr>
<tr>
<td>% ( J \leq 0 )</td>
<td>–</td>
<td>39.2 ± 7.34</td>
<td>3.57 ± 0.78</td>
<td>0.0 ± 0.0</td>
</tr>
</tbody>
</table>

*Fast* RW techniques provide a significant increase in speed while only incurring a minimal loss in accuracy. For both RWIS and RWIR, in Sec. 5.1.2 and Sec. 5.2.2 we developed techniques for increasing the adaptability of the *fast* RW algorithms, removing the dependence of the precomputation schemes on the number of eigenvectors and allowing the image graph weights and topology to be updated online. These techniques increase flexibility of the precomputed data, allowing *fast* RW paradigms to be integrated with other RW extensions. We also presented results on real medical data demonstrating the effectiveness of these techniques.

While the results shown here demonstrate the potential effectiveness of our techniques, the popularity of the RW algorithm has lead other interesting and potentially powerful extensions that are not discussed here, such as the inclusion of learnt shape priors and automatic seeding for RWIS [23, 25] and adaptive displacement labels for RWIR [317]. Further, the RW algorithm was shown by Couprie et al. to be part of a family of graph-based algorithms [83], including other popular algorithms such as Graph Cuts [38]. Random walks have also been applied to other imaging tasks, such as stereo matching [285] and shape correspondence [215], and extending our precomputation techniques to these applications may increase the impact of our contributions. Our future work will focus on extending our precomputation techniques to the extensions of the RW algorithm, similar graph-based algorithms, and other imaging applications mentioned above.

In Sec. 5.2.3 we introduced a precomputation scheme that is run online for RWIR, after both images being registered are available, to find weights for the prior probabilities that ensure a topology preserving transformation. This scheme runs quickly enough that it does not significantly affect the overall run-time of the full RWIR algorithm. Properly selecting regularization parameters is often key to accurate registration, yet in general can be very difficult without expensive trial and error approaches, involving a registration being performed multiple times and the results examined. Our method efficiently adjusts spatially adaptive regularization weights in order to provide explicit guarantees regarding topology.
preservation without over-regularizing regions with accurate prior probabilities. Ensuring topology preservation indirectly increases accuracy by removing anatomically infeasible regions from a transformation, as we demonstrate on real medical data. For future work, we will more rigorously examine how the values of the $\tau$ parameters affect the speed and accuracy of the registration for different applications.
Chapter 6

Conclusion and Future Work

6.1 Thesis Summary

In this thesis, we presented novel techniques for performing computation offline before any user interaction. These contributions spanned the spectrum of user interaction in medical image analysis tasks, from techniques focused on learning anatomical shape information in order to circumvent the lack of user input, to techniques focused on facilitating repeated user interaction.

In Chapter 3, we compare techniques for mapping probabilistic labels to a vector space in order to facilitate algebraic manipulation and statistical analysis. A well known method for performing this mapping is the LogOdds (ALR) transform, which has several useful properties, and has been applied to medical image analysis previously. However, LogOdds is asymmetric when there are multiple foreground labels, and changes in the background probability cause a greater change in the ALR transformed data than equivalent changes in the foreground probabilities. Making use of established methods for compositional data analysis, we proposed using the symmetric ILR transform to map probabilistic label. We demonstrated that the asymmetry of the ALR transform magnifies the importance of changes in the background label in statistical analysis techniques such as PCA, at the cost of other label boundaries, leading to shape models that capture foreground-background boundaries more accurately than foreground-foreground boundaries. Thus, the ILR transform is a more appropriate mapping in applications containing many foreground-foreground boundaries.

In Chapter 4, we utilize the above theory in describing a unique framework for fully automatic image segmentation that, to our knowledge, is the first to simultaneously achieve
four important properties: convexity, including a shape prior, multi-label, and probabilistic. While this segmentation framework has some drawbacks, particularly compared to the power provided by some non-convex formulations, it fills an important niche in the MIS literature.

We introduce several additions to our segmentation framework to improve its effectiveness for the difficult task of thigh muscle segmentation. Our anatomically-based alignment scheme leverages a relatively easy to solve subproblem (presegmentation) to perform an automatic rigid alignment. Our RF boundary detection technique learns the appearance of both true and false boundaries, with minimal prior information and without the free parameters often associated with determining boundary appearance. Our GLR transformation provides a segmentation representation that implicitly encodes label transition penalties. Each of these represent novel contributions that have potential to be extended to other applications.

In Chapter 5, we introduce extensions to the popular RW algorithm for interactive image segmentation and registration. Our extensions focus on using precomputation techniques to increase the speed and accuracy (by ensuring topology preservation) of the RW algorithm, reducing the amount of time spent by a user waiting while a RW algorithm ran.

6.2 Future Work

Deeper connections between these chapters could be established by noting the probabilistic segmentations generated offline by the fully automatic segmentation framework described in Chapter 4 could be used as prior probabilities in RWIS. Further, the RW algorithm uses Euclidean distances between probabilistic labels to define its energy function, which we argued against in Chapter 3. In our future work, we will explore possibilities for unifying the algorithms presented in this thesis into a single GLR-based framework for image segmentation and registration, using precomputation both to generate prior results and to speed user interaction, and incorporating statistically meaningful uncertainty information.

Throughout this thesis, we have identified other direct extensions to our work that we will pursue in the future:

- Drawing further connections between the Aitchison geometry and information geometry;
• Exploring other techniques for incorporating prior information in the GLR transformation;

• Extending the RF boundary classifier and corresponding energy terms to detect boundaries between specific labels;

• Investigate using thigh muscle shape models to improve the understanding of how these muscles are affected by COPD, leading to more informed treatment options;

• Develop precomputation techniques for extensions to the RW algorithms and other related discrete algorithms;

• Fine-tune our technique for ensuring TP registration results, and further explore the effects of the $\tau$ parameters.

Other future work will involve applying successful techniques from other works to increase the speed and robustness of our algorithms, such as GPU implementations and automatic learning of parameters from training data [26, 205].

The MIA techniques presented in this paper encode uncertainty by assigning discrete distributions over a set of labels to each pixel, which can be interpreted as distributions with the influence of all other pixels marginalized out. While this representation of a probabilistic segmentation or deformation is compact, it does not explicitly encode joint distributions between neighboring pixels. Thus, we are forced to approximate the statistics of some important quantities (e.g. we must use a fuzzy DSC instead of the expected DSC, the Jacobian of the expected deformation instead of the expected Jacobian). Our future work will involve developing techniques to infer joint distributions between neighboring pixels that are loyal to the original marginals. In the case of probabilistic results derived from MRF-based energies, we will explore using these energies directly to calculate joint distributions. These joint distributions would facilitate extending techniques developed for crisp segmentations that involve comparing two or more nearby pixels, thus improving the utility of probabilistic results.
Bibliography


BIBLIOGRAPHY


[165] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In ICCV, pages 1–8, 2007. 31


