

Image Synthesis in Magnetic Resonance Neuroimaging

by

Amod Jog

A dissertation submitted to The Johns Hopkins University in conformity with the
requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

February, 2016

© Amod Jog 2016

All rights reserved

Abstract

Automatic processing of magnetic resonance images (MRI) is a vital part of neuroscience research. Yet even the best and most widely used medical image processing methods will not produce consistent results when their input images are acquired with different pulse sequences. The lack of consistency is a result of multiple sources of variation in the acquired MRI data. MRI, unlike computed tomography (CT), does not produce images where the magnitude of the intensity is standardized across scanners. In a typical scanning session, different MRI pulse sequences are acquired at different resolutions for various reasons. Certain pulse sequences are prone to artifacts that cause corrupted data. Medical image analysts have developed preprocessing algorithms such as intensity standardization and image synthesis methods to address this problem. However, their performance remains dependent on knowledge and consistency of the pulse sequences used to acquire the images. In this thesis three different approaches—REPLICA, Ψ -CLONE, and SynthCRAFT—to perform image synthesis in MRI are presented. REPLICA is a multi-resolution framework that performs random forest regression with carefully designed features that capture anatomical variability in MRI.

ABSTRACT

Ψ -CLONE takes into account the physics of MRI acquisition process and estimates the pulse sequence parameters used to acquire the given subject image. These are then used to create new training images that are, by design, standardized to the subject image. This step allows for improved training of the random forest regression, which generates the final synthetic image. SynthCRAFT, in contrast to REPLICA and Ψ -CLONE, is a probabilistic framework for image synthesis. The conditional probability of the unknown, desired, synthetic image given the subject input images is modeled as a Gaussian conditional random field (CRF). Inference on this CRF, which models inter-voxel dependencies, results in the output synthetic image. All approaches were validated using simulated and real brain MRI data by direct image comparison and were shown to outperform state-of-the-art image synthesis algorithms. The ability to synthesize T_2 -weighted (T_2w) and FLuid Attenuated Inversion Recovery (FLAIR) images has been showcased for all three algorithms. Subsequent lesion segmentations of synthetic FLAIR images were shown to be similar to those obtained from real FLAIR images, thus demonstrating the utility of synthesis. In addition, REPLICA was shown to be capable of synthesizing full-head images (not skull-stripped), which is a challenging synthesis task. Intensity standardization using synthesis between two different T_1 -weighted pulse sequences was demonstrated using REPLICA and SynthCRAFT. Ψ -CLONE was used to standardize the intensities of a large dataset, leading to more consistent segmentation results within that dataset. All three algorithms were used to perform super-resolution of low resolution T_2w and FLAIR images.

ABSTRACT

The resulting super-resolution FLAIR images showed improved lesion segmentation. All three methods were demonstrated to be effective preprocessing algorithms that mitigated the variation in MRI data and improved the consistency of subsequent image processing.

Primary Reader: Dr. Jerry L. Prince

Secondary Readers: Dr. Russell Taylor and Dr. Junghoon Lee

Acknowledgments

I have put a lot of thought into writing this section of my thesis. Not that the following ones are any less thoughtful, but because this is that section of any thesis that can arguably claim maximum readership. This thesis is the culmination of my past four years at IACL, and past six years at Hopkins. It has been an enjoyable journey at its best and character-building at its worst. The credit for guiding me successfully through the labyrinth that is dissertation research has to go to my advisor, Dr. Jerry Prince. Needless to say, I have enjoyed working with him. He has always pointed me in the right direction that I needed to take to solve our problem. He has also trusted me enough to stumble around on my own and find my own paths in that direction. For that, I will always be grateful. I have appreciated his counsel the most when I believed I had reached a dead-end while solving a problem with no solution on the horizon. All our meetings that started with this premise ended on a positive note, with me somehow believing that not all was lost. This support has been vital for me to maintain my motivation, and I thank him for it. I have learned a lot from Dr. Prince, from formulating research problems to presenting their solutions effectively,

ACKNOWLEDGMENTS

and I hope to continue doing so in the future!

I could not have successfully meandered along in the direction Dr. Prince pointed me in, if it were not for Aaron Carass—lab-mate and friend. Aaron not only provided some of the best advice and ideas, but was also with me, right in the trenches fighting the ill-natured servers that sometimes refused to run my precious code. Many publications (including this one) would not have been possible on time if it were not for Aaron pushing me to do it and his expertise in editing and \LaTeX ’ing. He has helped me in every single aspect of my research since I joined IACL, for which I am thoroughly indebted.

I reserve a special thanks to Dr. Snehashis Roy, with whom I worked in my first two years at IACL, and continue to collaborate with. It was Roy who first started working on image synthesis and I jumped on the bandwagon two years later. I am especially grateful to him for being the trailblazer who had to cut his way through the hostile jungle of publishing very innovative work. It was because of his pioneering work, that our subsequent papers had a smooth passage later. Roy helped me become familiar with all the tools he had used and developed. He had also developed a lot of neat tricks to process our brain imaging data that he freely shared and lightened my load considerably.

I also want to thank Dr. Dzung Pham who has been a long-time collaborator and co-author on many of my publications. He has provided many a useful advice regarding research problems, writing advice, accessible data, and funding as well.

ACKNOWLEDGMENTS

I would not have learned half of what I have learned in the last six years, had it not been for my inspiring lab-mates. At IACL, hanging around Roy, John, Nathanael, Min, Andrew, Murat, Zhen, Chuyang, and Fangxu, pushed me to be better than what I was when I first joined the lab. In my previous lab at LCSR, this role was played successfully by Kel, Yixin, and Anand.

I would like to thank Dr. Rajesh Kumar, who was my advisor in LCSR for my initial training in graduate research. I am also grateful to Dr. Greg Hager, for his goodwill and assistance when he advised me in the interim period before I joined IACL.

I want to thank Drs. Bruno Jedynak, S. Rao Kosaraju, Greg Hager, and Ben Tsui for taking the time to serve on my GBO committee.

I am also thankful to my committee members, Dr. Russell Taylor and Dr. Junghoon Lee, for taking the time out of their busy schedules to read this thesis and serve on the committee.

As a student in the CS department, I was lucky to have kind and patient people on the administrative side such as Cathy, Debbie, Laura, Emily, and more recently Zack. They have taken care of all the administrative difficulties I have ever landed in and made sure that I never had to worry about them. For someone who somehow always lands on the wrong side of such things, I am extremely grateful that they had my back. At IACL, I am thankful to Laura Granite for the same. Thanks are also in order to Jamie Meehan and Alison Morrow at LCSR.

ACKNOWLEDGMENTS

“Apartment, department. Advisor, Budweiser”, was how a senior graduate student, a living fossil at that point, pithily summed up grad school when I asked him about it in my first year. After six years, I am glad to report that this is not true. My grad school experience has been enlivened by the friends that I have made here, and for which I thank them. Pavan and Kunal, my first roommates, were excellent friends and even better cooks. Raghavendra, squash and table-tennis partner for long and also provider of epic South Indian feasts. Sumedh, fellow language connoisseur and philosopher friend. Rahul Agarwal, a fellow failed cricketer, with whom were spent many a discussion over dinners. Hari Menon and Purnima Rajan, fellow CS grad students and eager participants at many coffee-fueled discussions on Sunday evenings at the CS lounge. Apurv Nakade, a fellow yoga fan, who has introduced me to his varied interests by taking me along. Pooja, who has been a friend that I could always count upon. Anand and Princy, who have been like family to me and are now almost too close to thank. They along with Siddharth (M) have been excellent companions for dinners, movies, and outings. In addition, M has been a great sounding board for talking research and is a most reliable squash partner. Vishwa and Akanksha, complete my close-knit group that has formed over the last four-five years and has, in many respects, satisfied my need of a home in this friendly but faraway country.

Napoleon Bonaparte nailed it when he said that an army marches on its stomach; so does a grad student. It would be ungrateful of me to not mention the Chipotle on St. Paul street, from where I derived my nourishment pretty much every day for the

ACKNOWLEDGMENTS

past six years. I also want to thank the erstwhile Cafe Q and the present day Daily Grind for brewing the best coffee in the campus, and keeping me caffeinated enough to help me conquer the vicissitudes of grad life.

Finally, I would like to thank my family—my parents, Drs. Kalpana and Suhas Jog, and my brother and best friend, Varun— for their love and support, that is so essential when one is far away from home and engaged in, what can often be, a solitary enterprise. To them I dedicate this thesis.

Dedication

Dedicated to Aai, Baba, and Varun.

Contents

Abstract	ii
Acknowledgments	v
List of Tables	xvi
List of Figures	xviii
1 Introduction	1
1.1 Variation in MRI Acquisitions	4
1.2 Image Synthesis: Motivation and Previous Work	6
1.3 Challenges and Contributions	12
2 REPLICA: Multi-resolution Random Forest-based Regression Synthesis	21
2.1 Introduction	21
2.2 Method	23

CONTENTS

2.2.1	Features	24
2.2.1.1	Multi-resolution Features	25
2.2.1.2	High Resolution Context Descriptor	27
2.2.2	Training a Random Forest	29
2.2.3	Predicting a New Image	33
2.3	Results	35
2.3.1	Synthesis of T_2w Images	36
2.3.2	Synthesis of Whole Head Images	38
2.3.3	Synthesis of FLAIR Images	43
2.3.4	Intensity Standardization	51
2.3.5	Super-resolution using REPLICA	55
2.3.5.1	T_2w Super-resolution	56
2.3.5.2	FLAIR Super-resolution	59
2.4	Summary and Discussion	62
Appendices		67
2.A	Parameter Selection for REPLICA	67
2.A.1	Number of Trees	68
2.A.2	Number of samples in leaf nodes	69
2.A.3	Number of training samples	70
2.A.4	Additional Atlases	72
2.A.5	Using Different Atlas Sets	73

CONTENTS

2.A.6	Size of Local Patch	74
3	Ψ-CLONE: Pulse Sequence Information Driven MR Image Synthesis	77
3.1	Introduction	77
3.2	Method	79
3.2.1	Estimation of Subject Pulse Sequence Parameters	81
3.2.2	Synthesizing a New Atlas Image with Subject Pulse Sequence Estimates	86
3.2.3	Learning and Applying Nonlinear Regression on Image Patches	87
3.3	Computational Phantom Experiments	90
3.3.1	Brainweb SPGR: Estimating a_{b_i}	90
3.3.2	Brainweb SPGR Intensity Standardization	94
3.3.3	Brainweb T_2w Synthesis	97
3.4	Real Data Experiments	99
3.4.1	Human Stability Data	99
3.4.2	MR Intensity Scale Standardization for MS Patients	102
3.4.3	T_2w Synthesis from Real MPRAGE Data	106
3.5	Further Synthesis Applications	111
3.5.1	Synthesizing Higher Resolution T_2w Data	111
3.5.2	FLAIR Synthesis	113
3.6	Summary and Discussion	119

CONTENTS

4 SynthCRAFT: Tree-encoded Conditional Random Fields for Image

Synthesis	125
4.1 Background	127
4.1.1 Conditional Random Fields	132
4.1.2 Gaussian Conditional Random Fields	136
4.2 Method	141
4.2.1 Problem Setup	141
4.2.2 Unary Regression Tree Construction	142
4.2.3 Construction of Pairwise Regression Subtrees	144
4.2.4 Parametrization	146
4.2.5 Parameter Learning	148
4.2.6 Inference	151
4.3 Results	154
4.3.1 Synthesis of T_2 w Images from T_1 w Images	154
4.3.2 Synthesis for FLAIR Images	155
4.3.3 Intensity Standardization	158
4.3.4 Super-resolution of FLAIR	161
4.4 Summary and Discussion	162

5 Discussion, Conclusions, and Future Work

5.1 Summary	166
5.2 REPLICA	167

CONTENTS

5.2.1	Key Points and Results	167
5.2.2	Future Work	168
5.3	Ψ -CLONE	170
5.3.1	Key Points and Results	170
5.3.2	Future Work	171
5.4	SynthCRAFT	172
5.4.1	Key Points and Results	172
5.4.2	Future Work	173
5.5	Concluding Thoughts	174
Bibliography		176
Vita		193

List of Tables

2.1	Mean and standard deviation (Std. Dev.) of the PSNR, UQI, and SSIM values for synthesis of T_2w images from 32 MPRAGE scans. . .	38
2.2	Mean and standard deviation (Std. Dev.) of the PSNR, UQI, and SSIM values for synthesis of T_2w images from 32 MPRAGE scans. . .	42
2.3	Mean (Std. Dev.) of PSNR (in decibels), UQI, and SSIM values over 125 FLAIR images synthesized by REPLICA.	46
2.4	Mean (Std. Dev.) of Dice coefficients based on LesionTOADS segmentation of the real FLAIR and synthetic FLAIR over 125 images. . . .	47
2.5	Mean PSNR (dB) and standard deviation for each of three methods across 40 scans. The three methods are nearest neighbor (NN) and tri-linear (TL) interpolation, b-spline (BSP) and our method—REPLICA. The state-of-the-art self similarity based super-resolution (SSS) [1] failed to run on any of the real data. We made several attempts to transform the real data into a usable form for SSS, which did not help.) * denotes statistically significantly better than either of the other two methods (α level of 0.01) using a right-tailed test.	57
2.A.1	Time and memory requirements of training REPLICA as function of number of training samples	71
3.1	Average T_1 , T_2 and relative P_D values of CSF, GM and WM at 1.5 T and 3 T	84
3.1	Flip angles used in Brainweb SPGR simulation vs estimated Flip angles after fitting.	93
3.2	RMSE between a_{b_1} and b_1 (as a % w.r.t max intensity) and PSNR (dB) values for Brainweb T_1w SPGR atlas synthesis for subject pulse sequences with varying flip angles.	94
3.3	PSNR (dB) values between a_1 and \hat{b}_1 , for standardization of Brainweb phantoms with varying flip angles ($^\circ$) and noise levels are shown for UPL and Ψ -CLONE. In the noise free case, UPL is better, however the introduction of noise causes UPL results to deteriorate.	97

LIST OF TABLES

3.4	PSNR (dB) and UQI values for Brainweb T_2w synthesis with varying noise levels (%) and flip angles ($^\circ$) are shown for MIMECS and our algorithm (Ψ -CLONE).	99
3.1	Coefficient of Variation (CV) of the relative tissue volumes ($\times 10^{-3}$) over the nine weeks on the original data and after intensity standardization with Ψ -CLONE for each of white matter (WM), cortical gray matter (Cort. GM), subcortical gray matter (Sub. GM), cerebrospinal fluid (CSF), and the ventricles (Vent.). Table 3.2 shows further breakdown of Sub. GM structures.	101
3.2	Coefficient of Variation (CV) of the relative tissue volumes ($\times 10^{-3}$) over the nine weeks on the original data and after intensity standardization with Ψ -CLONE for each of caudate, putamen, and thalamus.	102
3.3	The mean intensity value for the atlas used in the MS standardization experiment are shown for ten structures. We also show the mean ($\times 10^4$) and std ($\times 10^4$) (over 57 images) of the average intensity value for each structure, based on the original unnormalized data (Original) and after standardization with both UPL [2] and our method (Ψ - CLONE).	104
3.4	Contrast values between neighboring structures for original, synthetic, and atlas images. * indicates that the contrast in synthetic images higher than the original images (statistically significant using Student's one-tailed T test with $p < 0.05$).	105
3.5	Mean and standard deviation (Std. Dev.) of the PSNR and UQI values for synthesis of T_2w images from 40 MPRAGE scans.	109
4.1	Mean and standard deviation (Std. Dev.) of the PSNR, UQI, and SSIM values for synthesis of T_2w images from 32 MPRAGE scans. . .	155

List of Figures

1.1	Shown here are the intrinsic NMR parameter maps of a brain. Particularly (a) P_D map, (b) T_1 map, (c) T_2 map, and finally (d) a T_1 -weighted image produced by the MPRAGE pulse sequence.	2
1.2	Shown here are different pulse sequence acquisitions: (a) MPRAGE, (b) the P_D w output obtained from the first echo of the DSE sequence, (c) the T_2 w output obtained from the second echo of the DSE sequence, and (d) FLAIR.	4
1.3	An example image synthesis scenario with training atlas images and a given subject image.	8
1.4	(a) A full head T_1 w image, (b) T_2 w image, and (c) a joint histogram for corresponding T_1 w and T_2 w images. The x -axis represents the intensity in the T_1 w image and the y -axis represents the T_2 w intensity.	13
2.1	A graphical description of the REPLICA algorithm. The task depicted here involves predicting T_2 w images from T_1 w images. The left portion shows the training for all scales. The trained random forests (RF) at each level are then applied to the scaled versions of the input subject T_1 w image, starting from the coarsest scale $s = 3$ to the finest scale $s = 1$. The feature extraction step extracts different features at each level. Refer to the text for the notation.	26
2.2	High resolution context descriptor. The voxel \mathbf{x} at which this descriptor is calculated is at the center of this figure. The center of the slice \mathbf{o}_z is shown on the right. The unit vector \mathbf{u} is directed from \mathbf{x} to \mathbf{o}_z and is shown in red. It is rotated in increments of $\pi/4$ to identify the rest of the eight directions. At each radius $\in \{r_1, r_2, r_3, r_4\}$, along these eight different directions, we evaluate the mean of image intensities within a 3D cubic region (depicted here as colored 2D squares). The cubic widths $\{w_1, w_2, w_3, w_4\}$ are also shown for a set of regions.	28

LIST OF FIGURES

2.3	Shown are (a) the input MPAGE image, (b) the true T_2w image, and the synthesis results of (c) FUSION [3], (d) MIMECS [4], and (e) REPLICA (our method). The lesion (green arrow) and the cortex (orange arrow) in the true image are correctly synthesized by MIMECS and REPLICA, but not by FUSION. The dark boundary just outside the cerebral tissue (yellow arrow) is incorrectly synthesized as bright by MIMECS, but not by FUSION and REPLICA.	36
2.4	(a) Original input MPAGE, (b) REPLICA T_2w synthesis using $3 \times 3 \times 3$ patch as feature vector, followed by (c) with additional high resolution context descriptor feature, and (d) using the full, multi-resolution REPLICA framework. (e) shows the ground truth T_2w image. In the next row we have the corresponding difference images with respect to the real T_2w image in (f) , (g) and (h) respectively. It is clear that using multi-resolution REPLICA produces a higher quality synthesis for the challenging task of synthesis of full-head images. (i) - (p) Show the same images for a more superior slice. The maximum intensity in these images is 255. The difference images have a maximum intensity around 60.	39
2.5	PSNR, UQI, and SSIM as functions of the features and multi-resolution framework of REPLICA.	41
2.6	(a) The input MPAGE image, (b) the real T_2w image, (c) FUSION result, (d) MIMECS result, (e) REPLICA result. Note the synthesis errors in the cortex for FUSION and in the ventricles for MIMECS.	42
2.7	Subject input images along with the synthetic and true FLAIR images.	43
2.8	Visual comparison of FUSION, MIMECS and REPLICA for the FLAIR synthesis task. The two rows show images from different subjects. (a) and (e) real FLAIRs, (b) and (f) FUSION results, (c) and (g) MIMECS results, (d) and (h) REPLICA results.	46
2.9	(a) Input MPAGE for LesionTOADS segmentation, (b) original FLAIR, (c) synthetic FLAIR generated from T_1w , T_2w and P_Dw images, (d) LesionTOADS segmentation using real FLAIR + MPAGE, (e) LesionTOADS segmentation using synthetic FLAIR + MPAGE.	47
2.10	A Bland-Altman plot of lesion volumes for synthetic FLAIRs vs lesion volumes of real FLAIRs. Since zero lies within the $\pm 1.96\sigma$ range, these two measurements of the same quantity are interchangeable.	49
2.11	The red Bland-Altman plot is for lesion volumes as measured by a manual rater versus LesionTOADS lesion volumes on synthetic FLAIRs. The blue Bland-Altman plot is for lesion volumes as measured by a manual rater versus LesionTOADS lesion volumes on real FLAIRs. This plot shows that the lesion volumes obtained using the synthetic FLAIRs are marginally closer to those found by the manual rater as the red mean line is closer to zero.	51

LIST OF FIGURES

2.12	(a) SPGR, (b) MPRAGE, (c) Synthetic MPRAGE generated from the SPGR. (d–f) their respective AtlasEM segmentations. The SPGR segmentation is quite different from the MPRAGE one. The synthetic MPRAGE tries to bridge this difference.	54
2.13	(a) CSF AtlasEM volume scatter plot for SPGR vs MPRAGE and synthetic MPRAGE vs MPRAGE, (b) Ventricles AtlasEM volumes, (c) GM AtlasEM volumes, and (d) WM AtlasEM volumes. The blue scatter plots are of volumes observed in SPGR vs those in MPRAGE. The red scatter plots are of volumes observed in synthetic MPRAGE vs those in MPRAGE. The black line indicates the identity transform $x = y$	55
2.14	Shown here are (a) HR T_1 w image with 1.5 mm slice thickness, (b) HR T_2 w image with 1.5 mm slice thickness, (c) LR T_2 w image with 4 mm slice thickness, upsampled using cubic b-splines, and (d) SR T_2 w image using REPLICA on LR T_2 w of slice thickness 4 mm.	58
2.15	Coronal slices of: (a) HR MPRAGE, (b) LR FLAIR, (c) HR FLAIR, (d) SR FLAIR, followed by coronal slices of LesionTOADS lesion segmentations of (e) HR MPRAGE + LR FLAIR, (f) HR MPRAGE + HR FLAIR, and (g) HR MPRAGE + SR FLAIR. Sagittal slices of: (h) HR MPRAGE, (i) LR FLAIR, (j) HR FLAIR, (k) SR FLAIR, followed by sagittal slices of LesionTOADS lesion segmentations of (l) HR MPRAGE + LR FLAIR, (m) HR MPRAGE + HR FLAIR, and (n) HR MPRAGE + SR FLAIR.	61
2.16	The x -axis axis denotes subject IDs. The y -axis denotes the lesion volumes provided by LesionTOADS in mm^3 . The green plot shows lesion volumes acquired by LesionTOADS on HR FLAIR+HR MPRAGE. The blue plot shows LesionTOADS LR FLAIR lesion volumes, and the red plot shows LesionTOADS SR FLAIR lesion volumes. Notice that the green plot is closer to the red plot than the blue plot for most of the subjects.	62
2.A.1	Box plots for PSNR, UQI, and SSIM values obtained by comparing the synthetic REPLICA-generated images with the ground truth.	69
2.A.2	Synthesis performance as a function of t_c	70
2.A.3	As number of training samples increases, all metrics improve but plateau later. UQI and SSIM improve statistically significantly from $1e3$ to $1e5$, but not PSNR.	72
2.A.4	Box plots for all the metrics when an increasing number of atlases are used. PSNR for number of atlases=1 is significantly different than the rest of the atlases ($p < 0.05$) using the two-sample t-test. However other metrics and further increase in number of atlases do not show any significant differences.	73

LIST OF FIGURES

2.A.5	Box plots for all the metrics when four different atlases are used. PSNR for Atlas 1 is significantly different than the rest of the atlases ($p < 0.05$) using the two-sample t-test. However other metrics and other atlases show no significant differences.	74
2.A.6	Increase in patch size from $1 \times 1 \times 1$ to $3 \times 3 \times 3$ and further to $5 \times 5 \times 5$ are statistically significant ($p < 0.05$, two-tailed t-test).	76
3.1	A flow chart of the Ψ -CLONE algorithm.	80
3.1	Shown are (a) an SPGR image ($TR = 18$ ms, $TE = 10$ ms, $\alpha = 45^\circ$ with 0% additive noise) which we use as our subject image b_1 , the maximum intensity value is 1080, the (b) new atlas image, a_{b_1} , with pulse sequence parameters estimated from b_1 , and (c) the difference image $ a_{b_1} - b_1 $, the maximum value is 5.	91
3.2	Shown are (a) an SPGR image ($TR = 18$ ms, $TE = 10$ ms, $\alpha = 45^\circ$ with 3% additive noise) which we use as our subject image b_1 , the (b) reconstruction, \hat{b}_1 , of the subject image with the same pulse sequence as used to image (c) the atlas target image a_1 ($TR = 18$ ms, $\alpha = 30^\circ$, $TE = 10$ ms with 0% noise).	95
3.3	Histogram of a typical noiseless Brainweb phantom	96
3.4	(a) An example input subject SPGR from which we synthesize a T_2w image. (b) The true T_2w image and the outputs of synthesis produced by (c) MIMECS and (d) Ψ -CLONE.	98
3.1	The brown plot illustrates relative WM volumes (with respect to the ICV) over nine weeks before any standardization while the green plot illustrates the same values after intensity standardization using Ψ -CLONE.	100
3.2	Shown are (a) the true T_2w image, and the synthesis results from the MPRAGE for each of (b) FUSION, (c) MIMECS, and (d) Ψ -CLONE (our method). The lesion (in the green circle) and the ventricles (in the blue circle) in the true image are synthesized by MIMECS and Ψ -CLONE, but not by FUSION.	108
3.1	The MPRAGE has a through-plane resolution of 1.1 mm, while the original T_2w has through-plane resolution of 2.2 mm. This is evident as the true interpolated T_2w image shows blurring while the Ψ -CLONE synthesized image is crisp.	113
3.2	Subject input images along with the synthetic and true FLAIR images.	115
3.3	(a) Real T_1w image, (b) synthetic FLAIR, (c) LesionTOADS segmentation of real T_1w + synth. FLAIR, (d) real FLAIR, (e) LesionTOADS segmentation of real T_1w image + real FLAIR.	116
3.4	(a) Real T_1w image, (b) real T_2w image, (c) real P_Dw image, (d) synthetic FLAIR, (e) real FLAIR. The real FLAIR shows motion artifacts in (e), which are not present in the synthetic FLAIR (d).	118

LIST OF FIGURES

3.5	(a) Real T_1w image, (b) real T_2w image, (c) real P_Dw image, (d) synthetic FLAIR, (e) real FLAIR. The real FLAIR shows motion artifacts in (e), which are not present in the synthetic FLAIR (d). The real FLAIR in (e) shows blurring because of interpolation in order to match the high resolution T_1w	119
4.1	A typical hidden MRF. Shown here is the voxel location i with two neighbors, $i - 1$ and $i + 1$. The observed data at these voxels is given by \mathbf{x} . The underlying unknown y_i form the MRF.	128
4.1	A typical synthesis problem setup. In the first row, we show the atlas images \mathbf{x}_a and \mathbf{y}_a , used for training. The second row shows a given test subject image \mathbf{x}_s . The synthesis task involves estimating the unknown image, $\hat{\mathbf{y}}_s$	141
4.2	Construction and encoding of CRF parameters within a regression tree.	143
4.1	Shown are (a) the input MPRAGE image, (b) the true T_2w image, and the synthesis results from the MPRAGE for each of (c) FUSION, (d) MIMECS, and (e) SynthCRAFT (our method). The lesion (green circle) and the cortex (yellow circle) in the true image are synthesized by MIMECS and SythCRAFT, but not by FUSION.	154
4.2	(a) Subject T_1w , (b) T_2w , (c) P_Dw images, with (d) SynthCRAFT FLAIR, and (f) Subject Real FLAIR.	156
4.3	(a) Subject T_1w , (b) SynthCRAFT FLAIR, (c) LesionTOADS segmentation of $T_1w + \text{SynthCRAFT FLAIR}$, (d) Subject real FLAIR, and (e) LesionTOADS segmentation of $T_1w + \text{real FLAIR}$	157
4.4	A Bland-Altman plot of lesion volumes for synthetic FLAIRs vs real FLAIRs.	158
4.5	(a) CSF AtlasEM volume scatter plot for SPGR vs MPRAGE and synthetic MPRAGE vs MPRAGE, (b) Ventricles AtlasEM volumes, (c) GM AtlasEM volumes, and (d) WM AtlasEM volumes. The blue scatter plots are of volumes observed in SPGR vs those in MPRAGE. The red scatter plots are of volumes observed in synthetic MPRAGE vs those in MPRAGE. The black line indicates the identity transform $x = y$.	160
4.6	Coronal slices of LR, HR, and SR FLAIRs along with their corresponding LesionTOADS segmentation are shown. It is evident that using a LR FLAIR affects the segmentation of the lesions and even the cortex.	161
4.7	Shown are the lesion volumes acquired by LesionTOADS on HR FLAIR+HR MPRAGE (black), LR FLAIR+HR MPRAGE (blue), and SR FLAIR+HR MPRAGE (red). Note that the black plot is closer to the red plot than the blue plot for all but one of the subjects. . . .	163

Glossary

P_D Proton density.

T_1 Longitudinal relaxation time.

T_2 Transverse relaxation time.

CRF Conditional random field.

CSF Cerebrospinal fluid.

CT Computed Tomography.

CV Coefficient of variation.

DSE Double Spin Echo.

FLAIR Fluid Attenuated Inversion Recovery.

FUSION Synthesis using multi-atlas registration and intensity fusion.

GCRF Gaussian conditional random field.

Glossary

GM Gray matter.

ICV Intra-cranial volume.

MAP Maximum a posteriori.

MIMECS Patch-based synthesis approach that uses sparse reconstruction.

MPRAGE Magnetization Prepared Gradient Echo.

MR Magnetic resonance.

MRF Markov random field.

MRI Magnetic Resonance Imaging/Images.

MS Multiple Sclerosis.

NMR Nuclear magnetic resonance.

Ψ -CLONE Pulse Sequence Information-based Contrast Learning on Neighborhood Ensembles.

PSNR Peak signal to noise ratio.

REPLICA Regression Ensembles with Patch Learning for Image Contrast Agreement.

RF Radio-frequency.

Glossary

RMSE Root mean squared error.

RTF Regression tree field.

SPGR Spoiled Gradient Recalled.

SSIM Structural similarity.

SyN Symmetric diffeomorphic image registration.

SynthCRAFT Synthesis with Conditional Random Field Tree.

TD Delay time.

TE Echo time.

TI Inversion time.

TR Repetition time.

UPL Piecewise linear standardization method by Udupa et al..

UQI Universal quality index.

WM White matter.

WML White matter lesions.

Chapter 1

Introduction

Magnetic resonance imaging (MRI) of the brain is the investigative imaging modality of choice for clinicians and neuroscience researchers. MRI is capable of delivering high quality, high resolution, three dimensional images of the brain, in a risk free fashion. An MRI scanner stimulates the tissues that produce a signal that is dependent on their nuclear magnetic resonance (NMR) properties. This signal is recorded as the acquired image. In comparison, computerized tomography (CT), a competing imaging modality, uses multiple x-ray projections of the anatomy to reconstruct a 3D image. Harmful ionizing radiation, such as x-rays, is absent in MRI.

In addition to being safer, the true power of MRI lies in its immense versatility. The intrinsic NMR properties such as proton density (P_D), longitudinal relaxation time (T_1), transverse relaxation time (T_2), T_2^* decay time, among others, can be manipulated in a variety of ways to tease out structural and functional differences

CHAPTER 1. INTRODUCTION

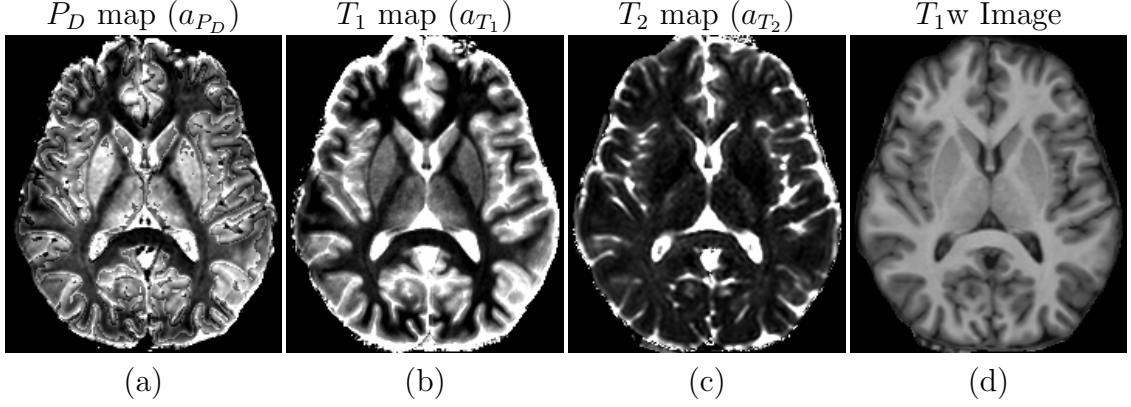


Figure 1.1: Shown here are the intrinsic NMR parameter maps of a brain. Particularly (a) P_D map, (b) T_1 map, (c) T_2 map, and finally (d) a T_1 -weighted image produced by the MPRAGE pulse sequence.

between the tissues being imaged. The procedure to perform such manipulations is called a *pulse sequence*. Pulse sequences are implemented using a strong, fixed magnetic field in the scanner that is subtly changed by applying gradients and radio-frequency (RF) stimulations. These stimulations cause the brain tissues to emanate RF signals with properties influenced by the tissue NMR parameters. Figures 1.1(a)–(c) show the intrinsic NMR properties, P_D , T_1 , and T_2 maps that influence MRI intensities in a typical MRI image (produced by the Magnetization Prepared Gradient Echo (MPRAGE) pulse sequence image in Fig. 1.1(d)). We notice that the white matter (WM), which forms the brightest regions in the MPRAGE image, has different NMR parameters than the gray matter (GM).

Many pulse sequences have been invented and continue to be invented for use in radiological diagnosis and medical science. During a single MRI scanning session, multiple pulse sequences can be acquired. Together, they provide us with complemen-

CHAPTER 1. INTRODUCTION

tary information about the underlying anatomy and function. Pulse sequences can be designed to obtain an image contrast that is primarily influenced by a particular intrinsic NMR parameter; for instance, pulse sequences like Magnetization Prepared Gradient Echo (MPRAGE) [5], or Spoiled Gradient Recalled (SPGR) [6] are designed to create a tissue contrast that is heavily T_1 -weighted (T_1w). For neuroimaging, this means that image intensities of tissues such as WM, GM, and cerebrospinal fluid (CSF) are heavily influenced by their T_1 values. These pulse sequences provide a very good WM-GM contrast and therefore, are useful to radiologists for visualizing the cortex and sub-cortical structures like the thalamus, caudate, putamen, etc. T_1w pulse sequences are also preferred as inputs for automated tissue segmentation [7–9] and cortical reconstruction algorithms [10–12]. Other common neuroimaging pulse sequences include the dual spin echo (DSE) which generates two images, one of which is P_D -weighted (P_Dw) while the other is T_2 -weighted (T_2w) (see Figs. 1.2(b) and 1.2(c)). Another popular T_2w pulse sequence is the FLAIR (FLuid Attenuated Inversion Recovery) sequence (see Fig. 1.2(d)). It is used extensively for imaging multiple sclerosis (MS) patients because it can provide a large image contrast between white matter lesions (WML) and normal WM. The lesions appear hyperintense (brighter) as compared to the normal WM, thus making it easier to delineate them using manual as well as automated lesion segmentation algorithms. It is vital to discover and quantify lesions because they are used as a criterion for disease diagnosis and can be tracked longitudinally to follow the disease state [13–15]. Figures 1.2(a)–(d) show the images

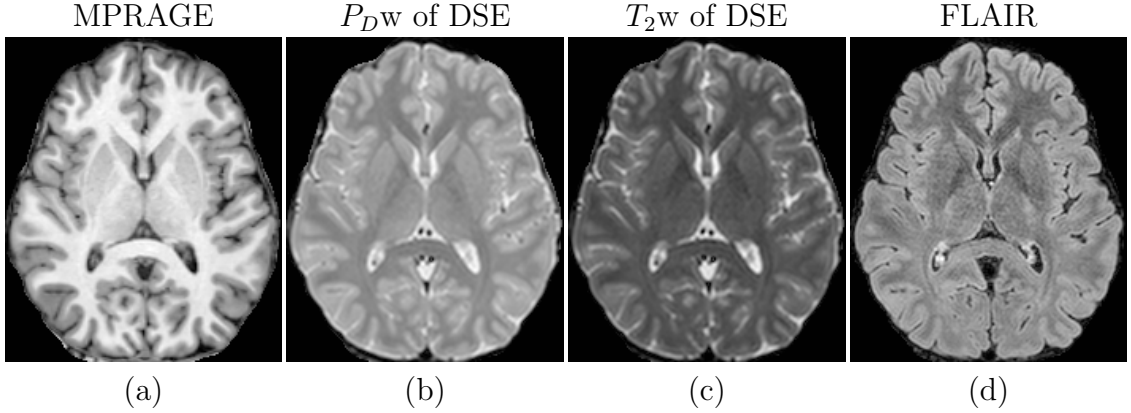


Figure 1.2: Shown here are different pulse sequence acquisitions: (a) MPRAGE, (b) the P_{Dw} output obtained from the first echo of the DSE sequence, (c) the T_{2w} output obtained from the second echo of the DSE sequence, and (d) FLAIR.

obtained by these sequences for a particular subject.

1.1 Variation in MRI Acquisitions

The versatility of MRI is a boon to diagnosticians. However, the variability in acquisitions introduced by the versatility poses a significant hurdle for image processing.

Some of the observed variations in MRI data are as follows:

- **Inconsistent acquisition protocols:**

Patient scanner time is finite and thus decisions are made about which subset of the many available pulse sequences to use. Another factor in this decision is the expense of scanning and the long scan time associated with various pulse sequences (DSE and FLAIR, for example).

CHAPTER 1. INTRODUCTION

- **Inconsistent quality within a scanning session:**

A confounding factor in acquired MRI data is its inhomogeneous quality arising from the different imaging requirements of each pulse sequence. For example, the DSE pulse sequence is generally acquired at a much lower resolution than a T_1 w MPRAGE pulse sequence because of a longer repetition time (TR). Even for the same pulse sequence, it is not easy to ensure that pulse sequence parameters, such as the flip angle for instance, are well-calibrated. This can result in the same subject having slightly different image data when scanned in the same scanner after a short interval.

- **Presence of artifacts and/or missing images:**

Pulse sequences like FLAIR suffer from artifacts that are not present in other pulse sequences [16], contributing to problems with their use in multimodal analysis. In particular, the imaging data can be corrupted due to patient motion or inappropriate parameter settings.

- **Multi-site and multi-scanner data:**

The problem is compounded when we consider data acquired from multiple sites that may house scanners from different manufacturers. Furthermore, scanner upgrades often change the image quality observed for the same pulse sequence. Both of these circumstances can affect the consistency of image processing algorithms [17].

None of these issues is difficult for trained radiologists to deal with for the purpose of diagnosis. However, most image analysis algorithms are not robust to these variations and give rise to inconsistent results [2, 18–23]. Results of image analyses obtained from such data are not ideal for drawing scientific conclusions. Our research in this thesis focuses on improving the utility of multimodal data through image synthesis, either by restoring corrupt data or by standardizing the intensity of existing data.

1.2 Image Synthesis: Motivation and Previous Work

By *image synthesis* we mean applying an intensity transformation to available MR images in order to produce new images that appear to have been acquired by a specific pulse sequence or an imaging modality. We also define intensity standardization, which is a common preprocessing task in MRI image processing pipelines. *Intensity standardization* transforms the intensities of a given subject image to a reference image, typically of the same (or similar) pulse sequence. It is a special case of image synthesis, in which the synthesized image is of the same (or similar) pulse sequence. Synthesized images are not meant to be used for diagnostic purposes or to replace scanning subjects. Rather, they are intended to facilitate image analysis for the extraction of clinical or scientific information.

We propose to use image synthesis as a solution to mitigate the variation in MRI

CHAPTER 1. INTRODUCTION

data. Some scenarios in which we can envision using image synthesis are as follows:

- If certain pulse sequence images were not acquired in a dataset, and were retrospectively discovered to be useful for image processing, image synthesis can be used to generate these.
- If certain pulse sequences images were acquired at a lower resolution, image synthesis can be used to create super-resolution images from the available ones. Improved resolution leads to better image processing such as improved registration and segmentation.
- If in an imaging session, one of the images is corrupted due to artifacts or any other reason, image synthesis can be used to generate an artificial replacement using the rest of the available images.
- If we have datasets from two different sites or scanner manufacturers that we need to reconcile, we can use image synthesis to transform image intensities of one of the datasets to match with the other to produce an intensity standardized dataset that will have a similar response to image analysis algorithms.

Image synthesis has gained significant attention in the medical imaging community in the last seven years [4, 24–38]. A typical image synthesis task is shown in Fig. 1.3, where the goal is to predict a synthetic T_2 w image \hat{b}_2 given an input T_1 w image b_1 . The target T_2 w contrast in \hat{b}_2 needs to be the same as the T_2 w contrast observed in a training image a_2 . We also have the corresponding T_1 w image a_1 , which may or

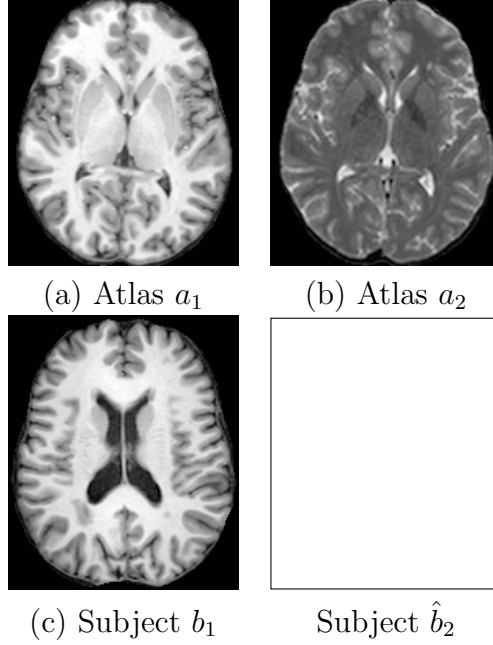


Figure 1.3: An example image synthesis scenario with training atlas images and a given subject image.

may not have the same acquisition parameters as b_1 . a_1 and a_2 preferably should be of the same resolution. We refer to the training images as *atlas images*. We refer to the test image b_1 and the unknown synthetic image \hat{b}_2 as *subject images*. By *pulse sequence* we mean particular image acquisition strategy, for example, MPRAGE. By *contrast* we mean the tissue contrast produced by a pulse sequence, say a T_1 w contrast or a T_2 w contrast. Since these contrasts are visually very different from each other, they can be treated as different *modalities* as well. For instance, the MPRAGE pulse sequence (or modality) produces a T_1 w contrast image. Since in most cases our data consists of specific pulse sequences producing specific contrasts, these terms are used interchangeably.

CHAPTER 1. INTRODUCTION

Image synthesis approaches in the literature can be broadly classified into two types:

- Registration-based
- Intensity transformation-based.

The main idea behind registration-based image synthesis was first described by Miller et al. [39]. They presented image synthesis as a single atlas registration and transformation problem. Given a subject image b_1 with contrast \mathcal{C}_1 and a pair of co-registered atlas images a_1 and a_2 of contrasts \mathcal{C}_1 and \mathcal{C}_2 , respectively, a_1 is registered to b_1 using a deformable registration algorithm, and the obtained transformation is then applied to a_2 to produce the synthetic image \hat{b}_2 with contrast \mathcal{C}_2 . They demonstrated synthesis of positron emission tomography images (PET) from MR images using this approach. An extension of this approach was described recently by Burgos et al. [3, 33] to synthesize CT images from MR images. They used multiple, aligned pairs of MR and CT images as atlases. Given a subject MR image, they registered all the MR atlases to it using deformable registration. The same deformations were applied to corresponding CT atlas images and intensity fusion was performed at each voxel to synthesize a subject CT image. The synthetic CT images were then used to learn the attenuation coefficients needed for PET reconstruction. This work was further extended by Cardoso et al. [40], wherein they framed the multi-atlas registration and intensity fusion as a generative process to perform image synthesis and inlier-outlier

CHAPTER 1. INTRODUCTION

classification.

Registration-based approaches are heavily dependent on the quality of the initial deformable registrations. Registration is not very accurate in the cortex of the brain, which is a highly convoluted region with many fine details. Consequently the synthesis in such regions is inaccurate. Another important issue is the quality of synthesis when the subject image presents abnormal anatomy, for example white matter lesions in multiple sclerosis patients. The location of these lesions in the brain is variable from subject-to-subject. Registration-based approaches are severely limited by the fact that the lesions may not occur in the exact same anatomical region of the atlas images as that of the subject image. If the atlas regions do not show lesions in those specific regions, then these will not be synthesized in the synthetic subject image. All current registration-based approaches [3, 39, 40] suffer from this problem. Cardoso et al. [40] model the intensities not present in the atlas as an outlier class and detect such regions as outliers in the synthesis process. However, the synthetic images themselves cannot be used as substitutes for real images for further image processing as they do not have the same appearance.

Intensity transformation-based image synthesis approaches are usually supervised prediction approaches. Supervised approaches require training data in the form of aligned atlas pairs of input and target contrast images. The learning task involves predicting the desired contrast intensities given information derived from the input images. To do this, one needs to create training data from the atlas images. For a

CHAPTER 1. INTRODUCTION

voxel location i in the atlas image a_1 , we can extract a feature vector \mathbf{f}_i , which could be a small 3D patch of voxels around i , for instance. It is paired with a corresponding voxel intensity v_i from a_2 . These are sampled from all over the atlas images to create the training data. Since the goal is to predict an image voxel intensity, which is a continuous value, image synthesis is a regression problem. It is possible to learn this regression using one of the many regression algorithms in the machine learning literature [41, 42]. Once the regression is learned, we can extract the same features from the subject image b_1 , apply the regression and generate a voxel intensity in the synthetic subject image \hat{b}_2 . On the face of it, this appears to be a reasonable approach to tackle image synthesis and researchers have approached it in a variety of ways.

One of the first intensity transformation-based synthesis methods, image analogies [43], came out of the computer graphics literature. Given the atlas image a_1 and a_2 of contrasts \mathcal{C}_1 and \mathcal{C}_2 respectively, a_2 is assumed to be a filtered version of a_1 , i.e. $a_2 = h * a_1$, where h is an unknown filter and $*$ denotes the convolution operator. Given a subject image b_1 , the task is to generate an image \hat{b}_2 , such that $\hat{b}_2 = h * b_1$. Each patch extracted from b_1 was matched to k nearest neighbor patches extracted from a_1 . The corresponding patches from a_2 were combined to form a patch of \hat{b}_2 , in a multi-scale framework. Image analogies has seen use in MR image synthesis with the goal of producing synthetic images for registration purposes [35]. Image analogies is handicapped by the fact that the nearest neighbor search for a high-dimensional patch, among a million patches is computationally very expensive. Roy et al. [4, 28] framed

CHAPTER 1. INTRODUCTION

the image synthesis problem as a sparse representation and reconstruction problem for small patches. Patches from a_1 are extracted to form a dictionary. Corresponding patches from a_2 are also extracted to create the target contrast dictionary. Next, each patch from b_1 is expressed as a sparse, linear combination of the a_1 dictionary patches. The corresponding patch of \hat{b}_2 is generated as the same linear combination of the corresponding patches from a_2 dictionary. They showed a variety of applications of image synthesis in neuroimaging [4]. This approach, which is called MIMECS [4, 28], is computationally expensive. It is not easy to add new features and increase the dimensionality of the patches, as this slows down the reconstruction process significantly; yet without additional features, some of the more challenging synthesis tasks (like full-head image synthesis) become impossible. Moreover, MIMECS was shown to be unable to synthesize pulse sequences like FLAIR [4] for subjects showing white matter lesions. FLAIR images are vital for imaging MS patients with white matter lesions because these lesions appear hyperintense with respect to the normal white matter and can be easily visualized and delineated.

1.3 Challenges and Contributions

Limitations in current, state-of-the-art intensity transformation-based algorithms motivate a number of challenges that we propose to tackle.

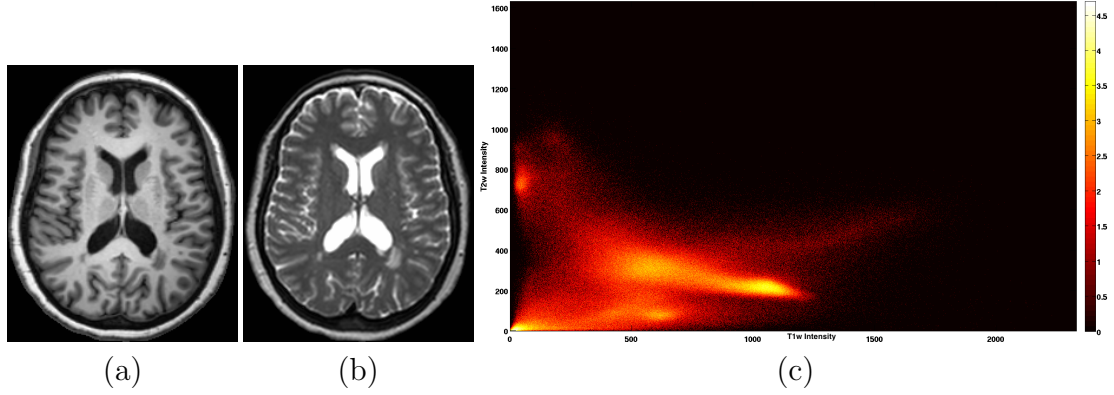


Figure 1.4: (a) A full head T_1 w image, (b) T_2 w image, and (c) a joint histogram for corresponding T_1 w and T_2 w images. The x -axis represents the intensity in the T_1 w image and the y -axis represents the T_2 w intensity.

- **Feature design:**

Treating image synthesis as a regression problem forces us to design appropriate features that will produce an accurate synthetic image. The human head consists of a variety of tissues such as skin, fat, bone, WM, GM, CSF, and lesions among others. The location and intensities of these tissues in MRI varies considerably from subject to subject. In order to synthesize the appropriate intensities for these tissues, we need features that can disambiguate between confounding intensity mappings.

- **Regression:**

Once a feature vector design is fixed, we must learn the regression that is going to generate the synthetic image, voxel by voxel. Looking at the joint histogram of a T_1 w and T_2 w image pair in Fig. 1.4, it is clear that this regression must be nonlinear. Nonlinear regression can be learned in a variety of ways. We must

CHAPTER 1. INTRODUCTION

choose an algorithm that is accurate, robust, easy to parametrize, interpretable, and quick to train and predict.

- **Intensity standardization:**

A regression algorithm will only be effective in prediction on test data if the samples in the training data are representative of the samples in the test data. The training feature vectors are extracted from a_1 and the test feature vectors from b_1 . All feature vectors that can be designed and extracted are going to be a function of the intensity images a_1 and b_1 . However, a_1 and b_1 do not necessarily have the same MRI acquisition parameters. This means that they will not have similar intensity values for similar tissues. The relation between the common tissue intensities of a_1 and b_1 may also not be a simple transform, for example a simple linear scaling. Thus, it implies that feature vectors extracted from a_1 will not be representative of the feature vectors extracted from b_1 , resulting in poor prediction performance and therefore, poor quality synthesis.

- **Probabilistic interpretation:**

If a_1 and b_1 are assumed to be intensity standardized, a typical regression algorithm (and all the previous synthesis algorithms) will predict the synthetic image \hat{b}_2 , voxel by voxel, in a somewhat ad-hoc manner. Voxels in medical images do not have intensities that are independent of each other. Neighboring voxels tend to belong to the same anatomy and tend to present similar intensities.

CHAPTER 1. INTRODUCTION

None of the previous image synthesis algorithms attempts to model these inter-voxel dependencies. In addition, the voxel-by-voxel generated images are not optimal according to any well-defined objective. It is therefore not possible to determine if a synthetic image is better or worse than any other according to any well-defined criterion. These issues can be addressed if the problem is modeled in a probabilistic framework, which has not been attempted by previous synthesis algorithms.

Our contributions listed below, address these challenges.

1. **REPLICA:**

Our first contribution jointly addresses the challenges of feature design and regression. Appropriate feature design is crucial for MRI synthesis, particularly for full-head images that are not skull-stripped (see Fig. 1.4). If the synthesis task entails synthesizing the T_2 w image in Fig. 1.4(b), from the T_1 w image in Fig. 1.4(a), we can make the following observations. In Figure 1.4(a) the CSF and bone from the skull in the T_1 w image appear dark. However, the CSF appears very bright in the corresponding DSE T_2 w image while the skull still appears dark (Fig. 1.4(b)). If we let the feature vector be just a single T_1 w voxel intensity from a_1 , it is paired with the corresponding single T_2 w voxel intensity from a_2 when the training data is created. In the training data, we will have dark voxels in T_1 w images being mapped to dark voxels in T_2 w images (skull voxels) and also being mapped to bright voxels in T_2 w images (CSF voxels). This

CHAPTER 1. INTRODUCTION

ambiguity is visible in the joint histogram of T_1w and T_2w images in Fig. 1.4(c), where for low T_1w intensities on the x -axis, we see two clusters, one with a low T_2w intensity (skull) and one with a high T_2w intensity (CSF), on the y -axis.

A regression learned on such training data is bound to have training and hence, prediction errors. Thus, it is imperative to design features that will result in an unambiguous mapping. An ideal feature vector should encode global and local information that remains comparable across different subjects. For example, a small intensity patch surrounding voxel i provides information about the local anatomical context. By itself however, it is inadequate to disambiguate between one-to-many intensity mappings. Therefore, we augment it with a *context descriptor*, described in Chapter 2, that provides some global spatial context of the location of the voxel in the brain anatomy.

The more features we augment, the higher is the dimensionality of the feature vector. At very high dimensions, it is difficult to find nearest neighbor matches for a given test feature vector based on Euclidean distance. It is also computationally intensive. We observed that a small patch provides a much larger anatomical context at a lower resolution. Based on this observation, we have designed a framework that decomposes the images into a multi-resolution Gaussian pyramid and synthesizes each level of the pyramid, from lower to high resolution, step-by-step. We call this framework REPLICA. In REPLICA, a small $3 \times 3 \times 3$ -sized patch at two levels lower resolution in the Gaussian pyramid

CHAPTER 1. INTRODUCTION

covers a large part of the anatomy and provides global context that is provided by a $12 \times 12 \times 12$ -sized patch at the given resolution (i.e., the highest available resolution). Using such a patch immediately reduces the dimensionality of the feature vector from 1728 to 27.

To learn the nonlinear regression, we choose the random forests [42] method. A random forest learns the nonlinear regression by approximating it in a piecewise constant manner and learning all the pieces. It is very fast to train, and easy to interpret. Prediction is also very fast, which is ideal since we want to use our algorithms for quick pre-processing.

REPLICA can be used to perform intensity standardization and alternate tissue contrast synthesis. REPLICA-generated synthetic images have been shown to provide more consistent segmentation. We have also used REPLICA to generate substitute synthetic images as target images for registration tasks and shown improved registration [44]. More importantly, REPLICA is designed to be robust enough to handle complex synthesis scenarios such as synthesizing full-head MRI images, unlike skull-stripped brain images. Full-head image synthesis is a hard problem due to large variations in anatomy and intensity values and cannot be achieved by other synthesis algorithms as easily. REPLICA is described in detail in Chapter 2. A journal paper manuscript describing REPLICA has been submitted and is under review.

CHAPTER 1. INTRODUCTION

2. Ψ -CLONE:

Our second contribution addresses the problem of intensity standardization in synthesis by standardizing the subject image b_1 to the atlas image a_1 using a novel technique that takes into account the MRI acquisition physics of the the pulse sequences used to acquire a_1 and b_1 . In this approach, given just the intensity image b_1 , we can approximately estimate the pulse sequence parameters used to acquire it. We can apply this pulse sequence imaging equation to the intrinsic NMR parameter maps of the atlas images such as a_{PD} , a_{T_1} and a_{T_2} (see Fig. 1.1) to create a new, additional atlas image a_{b_1} , which is a synthetic atlas image that, by design, has the same pulse sequence parameters as that of b_1 . In essence, they are intensity standardized, and hence, feature vectors extracted from a_{b_1} are now representative of those extracted from b_1 . Now we can extract training data and learn a regression that can then be applied to feature vectors from b_1 to produce a more accurate synthetic image \hat{b}_2 . In our approach, we use small patches as feature vectors and use random forests [42] to learn the nonlinear regression that predicts intensities of \hat{b}_2 .

We refer to this approach as Ψ -CLONE [45]. We have used Ψ -CLONE for synthesizing T_2 w images from T_1 w images, synthesizing FLAIR images from input T_1 w, P_D w, and T_2 w images, and intensity standardization tasks. We have also shown improved image analysis results such as consistent segmentation, as a

CHAPTER 1. INTRODUCTION

result of our synthesis. Ψ -CLONE is computationally fast and can be easily used as a preprocessing method for image analysis pipelines. Ψ -CLONE is described in depth in Chapter 3 and has also been published in the Medical Image Analysis journal [45].

3. SynthCRAFT:

In our third contribution, we have proposed a probabilistic framework within which we can frame any image synthesis problem. REPLICA in its present form does not produce an image that is considered optimal according any well-defined objective. We model the synthesis problem as an inference problem on a Gaussian conditional random field (GCRF). We employ what we have learned from designing REPLICA and Ψ -CLONE to build a framework within which we can specify probability distributions of entire unknown, to-be-synthesized images, given available images. We call this framework SynthCRAFT [46]. Using SynthCRAFT, given a subject T_1 w image b_1 , we are able to construct a probability distribution $p(b_2|b_1)$, of the corresponding T_2 w image b_2 , given b_1 . Once we know this distribution, predicting the synthetic image \hat{b}_2 is simply a matter of finding the maximum a posteriori (MAP) estimate of $p(b_2|b_1)$.

Unlike REPLICA, which estimates the synthetic image voxel-by-voxel, each independent of the rest, SynthCRAFT estimates an entire image keeping note of the fact that neighboring voxel intensities are correlated to each other and are not completely independent. We show that SynthCRAFT is capable of

CHAPTER 1. INTRODUCTION

most of the tasks that REPLICA is and provides comparable results. REPLICA and SynthCRAFT attempt to solve the same problem. REPLICA is easy to interpret and hence is very well-tuned for image synthesis tasks, as is evident based on empirical testing. However, we believe that SynthCRAFT is a more general, elegant, and a principled way to define and solve the image synthesis problem and with some engineering effort, will replace REPLICA in the future. We describe SynthCRAFT in Chapter 4.

In Chapter 5, we discuss the conclusions and implications of our image synthesis techniques and contemplate future directions for our research.

Chapter 2

REPLICA: Multi-resolution

Random Forest-based Regression

Synthesis

2.1 Introduction

In this chapter, we describe our intensity transformation-based MR image synthesis framework. The synthesis problem is illustrated in Fig. 1.3 in Chapter 1. We assume that the atlas image set, $\mathcal{A} = \{a_1, a_2, \dots, a_n, \dots, a_r\}$ and the subject image set $\mathcal{B} = \{b_1, b_2, \dots, b_n\}$, are such that each a_i and corresponding b_i are already intensity standardized or can be standardized by a simple intensity scaling. The goal is to synthesize the subject image \hat{b}_r , which has the same contrast as the atlas image a_r .

CHAPTER 2. REPLICA

All previous and current image synthesis approaches [3, 4, 40, 43] also frame the image synthesis problem similarly. Here we describe our solution, which shares some common elements with some of these approaches.

We model the intensity transformation in our method as a nonlinear regression in a feature space that predicts the intensity in the target modality. The choice of feature space is crucial for nonlinear regression to work well. We desire a feature space that provides an unambiguous mapping from itself to the target contrast intensities. This means that two feature vectors that are close together in the feature space should not ideally map to two different intensities in the target contrast. The T_1 w- T_2 w joint histogram in Fig. 1.4 illustrates the necessity of an unambiguous mapping for typical image synthesis tasks. To that end, we use local patches and additional context features to form our feature space. We also wrap the nonlinear regression in a multi-resolution framework similar to the ideas described by Hertzmann et al. [43]. The nonlinear regression is learned using a regression ensemble based on random forests [47].

We call our approach Regression Ensembles with Patch Learning for Image Contrast Agreement or REPLICA. Our approach is computationally much faster than existing approaches and delivers comparable, if not better synthesis results. REPLICA can be used to to synthesize FLAIR images, which had not been done previously by other methods. REPLICA is also capable of synthesizing full-head images, as opposed to skull-stripped images. Full-head images possess a lot of variability across subjects and have been difficult synthesis targets. Additionally, we have demonstrated use

CHAPTER 2. REPLICA

of REPLICA in super-resolution and intensity standardization scenarios. Elements of this work have appeared in previous conference publications [30, 48]. A previous version of REPLICA was also used in Ψ -CLONE, described in Chapter 3 and in [45].

In this chapter, we provide a complete exposition of our multi-resolution approach, which is the culmination of several previous publications [30, 45, 48]. We include more in-depth experiments showcasing the efficacy of synthesis not only by performing image comparison with real images but also by quantifying the results of further image analysis on synthetic images and showing comparable results with real images. We describe our method in Section 2.2 and results in Section 2.3. In the results section, we show synthesis of T_2w and FLAIR images, intensity standardization of SPGR and MPAGE modalities, and super-resolution of T_2w and FLAIR images. Discussion, conclusions, and future work is in Section 2.4. Parameter selection experiments are described in Appendix 2.A.

2.2 Method

Let $\mathcal{B} = \{b_1, b_2, \dots, b_m\}$ be a subject image set, imaged with pulse sequences Φ_1, \dots, Φ_m . This set contains m images from m different pulse sequences such as MPAGE, FLAIR, DSE, etc. Let $\mathcal{A} = \{a_1, a_2, \dots, a_m, a_r\}$ be the atlas collection generated by the pulse sequences Φ_1, \dots, Φ_m (which are the same as those of the subject images) and Φ_r , where Φ_r is the target pulse sequence that we want to

CHAPTER 2. REPLICA

synthesize from the subject set. The atlas collection can contain imaging data from more than one individual but for simplicity in this description, we assume that there is only one individual in the atlas.

Our goal is to synthesize a subject image \hat{b}_r that has the same tissue contrast as the atlas image a_r . We frame this problem as a nonlinear regression where we want to predict the intensities of \hat{b}_r , voxel-by-voxel. This nonlinear regression is learned using random forest regression [47] as described in detail below. Training data is generated using the atlas image set by extracting feature vectors $\mathbf{f}(\mathbf{x})$ at voxel locations \mathbf{x} in the atlas image domain Ω . These features are paired with the voxel intensity $a_r(\mathbf{x})$ of the atlas target modality to learn the intensity transformation.

2.2.1 Features

Generating a synthetic image of modality Φ_r from modalities Φ_1, \dots, Φ_m involves calculating an intensity transformation that jointly considers features in these modalities to predict the intensities generated by Φ_r . If we consider an individual voxel intensity as the only feature then in most realistic image synthesis scenarios the intensity transformation is not a one-to-one function between the input and target modality voxel intensities. This is evident for the example of T_1 w and T_2 w images in Fig. 1.4 in Chapter 1. The regression learned from such ambiguous data is bound to be error-prone.

One way to reduce such errors is to add additional features associated with each

CHAPTER 2. REPLICA

voxel $\mathbf{x} \in \Omega$. Previous approaches like image analogies [43] and MIMECS [4] used small image patches (usually $3 \times 3 \times 3$ voxels centered on \mathbf{x}) as features. Small patches add some spatial context to a single voxel, but they are often unable to resolve the ambiguity of a one-to-many mapping from feature space to the target modality intensity. Registration-based approaches [3, 40] attempt to solve this issue by carrying over the spatial context from the atlas images to the subject image via registration under the assumption that after registration, the intensity transformation is simple enough to be learned by using small patches. We address this problem in two ways, first by using a multi-resolution framework and second by adding remote features which we call *context features*.

2.2.1.1 Multi-resolution Features

For each image a_i in the atlas set, we construct a Gaussian pyramid (using $\sigma = 1$ voxel) on scales $s \in \{1, \dots, S\}$. Let the atlas images at scale s , be $\mathcal{A}^s = \{a_1^s, \dots, a_m^s, a_r^s\}$. The first level ($s = 1$) corresponds to the original high resolution atlas images, and each successive level is created by Gaussian smoothing and downsampling the images by a factor of 2. These levels depicting the creation of training data are shown in Figs. 3.1(a), (b), and (c). At the coarsest resolution of the Gaussian pyramid ($S = 3$ is sufficient in most scenarios), the intensity transformation is simple and can be learned using small $p \times q \times r$ -sized 3D patches $\mathbf{p}_i^s(\mathbf{x})$ extracted from images a_i^s , $i \in \{1, \dots, m\}$ and concatenated together to create $\mathbf{p}^s(\mathbf{x}) = [\mathbf{p}_1^s(\mathbf{x}), \dots, \mathbf{p}_m^s(\mathbf{x})]$.

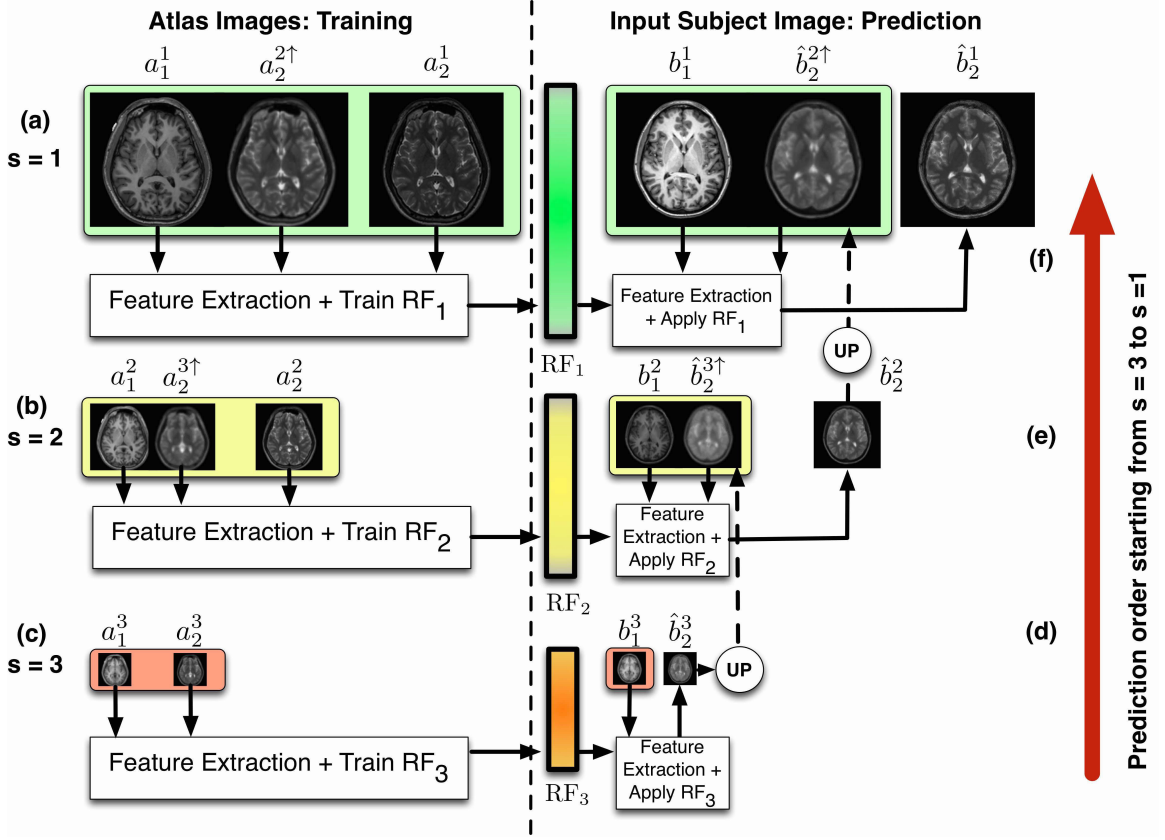


Figure 2.1: A graphical description of the REPLICA algorithm. The task depicted here involves predicting T_2w images from T_1w images. The left portion shows the training for all scales. The trained random forests (RF) at each level are then applied to the scaled versions of the input subject T_1w image, starting from the coarsest scale $s = 3$ to the finest scale $s = 1$. The feature extraction step extracts different features at each level. Refer to the text for the notation.

(Generally, we set $p = q = r = 3$.) This step is illustrated in Fig. 3.1(c); the details of training the random forest regression are described later in Section 2.2.2.

For all the scales $1 < s < S$ (which is just $s = 2$ in Fig. 3.1), the full feature vector consists of two distinct parts. The first part is a small patch at voxel location \mathbf{x} from the image set \mathcal{A}^s , which can be described by $\mathbf{p}^s(\mathbf{x}) = [\mathbf{p}_1^s(\mathbf{x}), \dots, \mathbf{p}_m^s(\mathbf{x})]$. The second part is a small patch at \mathbf{x} taken from $a_r^{s+1\uparrow}$, the upsampled *target image* from

CHAPTER 2. REPLICA

one level lower resolution in the Gaussian pyramid, where upsampling is done via trilinear interpolation. We denote this upsampled target patch by $\mathbf{q}^s(\mathbf{x})$. These steps correspond to stages (a) and (b) in Fig. 3.1. The patch $\mathbf{q}^s(\mathbf{x})$ helps to disambiguate regions of similar intensities by providing a low resolution estimate of the intensities in the target modality Φ_r . The feature vector at levels $1 < s < S$ is the concatenation of these two features: $\mathbf{f}^s(\mathbf{x}) = [\mathbf{p}^s(\mathbf{x}), \mathbf{q}^s(\mathbf{x})]$.

2.2.1.2 High Resolution Context Descriptor

Although multi-resolution features can help to disambiguate the most confounding intensity mappings, their use alone can yield overly smooth synthetic images due to the information arising from the lower resolutions in the multi-scale pyramid. To address this problem we introduce special context features that are used only at the finest level in the pyramid (stage (a) in Fig. 3.1).

We assume that the images have been registered to the MNI coordinate system [49] and are in the axial orientation, with the center of the brain approximately at the center of the image. Let the voxel \mathbf{x} be located on slice z , with the slice center at \mathbf{o}_z . Thus the unit vector $\mathbf{u} = \mathbf{o}_z - \mathbf{x} / \|\mathbf{o}_z - \mathbf{x}\|$ identifies the direction to the center of the slice from voxel \mathbf{x} .

We then define eight directions by rotating \mathbf{u} by angles $\{0, \frac{\pi}{4}, \dots, \frac{7\pi}{4}\}$ about the axis perpendicular to the axial slice (see Fig. 2.2). In each of these directions, at a radius $r_i \in \{r_1, r_2, r_3, r_4\}$ we calculate the average intensities in cubic regions with

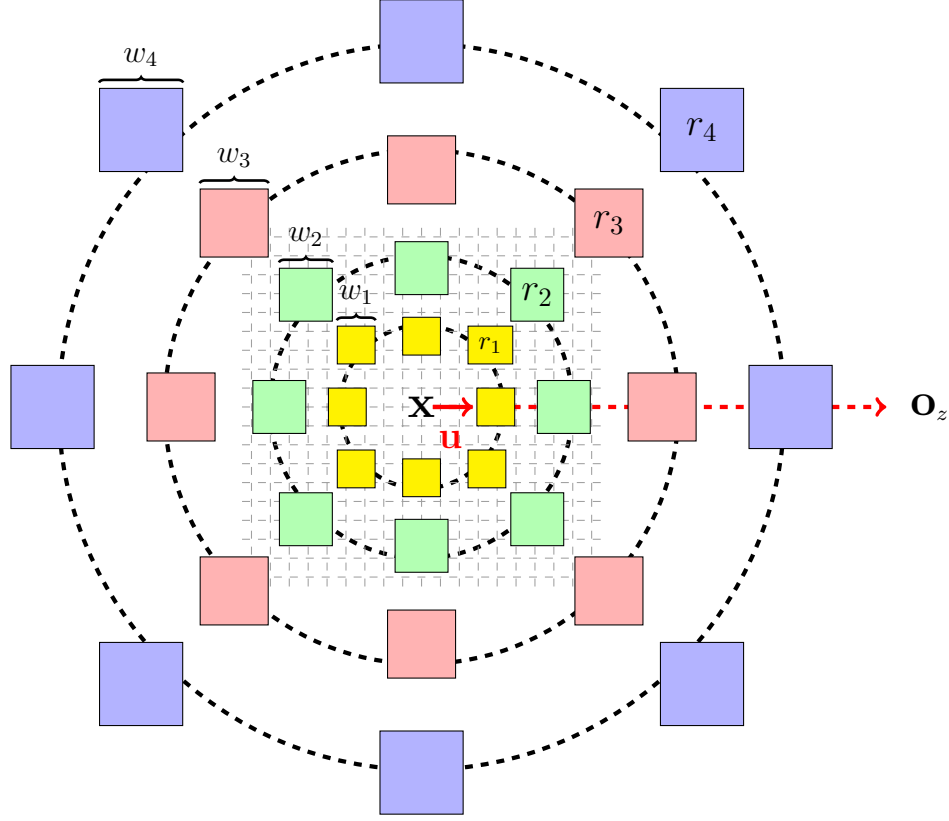


Figure 2.2: High resolution context descriptor. The voxel \mathbf{x} at which this descriptor is calculated is at the center of this figure. The center of the slice \mathbf{o}_z is shown on the right. The unit vector \mathbf{u} is directed from \mathbf{x} to \mathbf{o}_z and is shown in red. It is rotated in increments of $\pi/4$ to identify the rest of the eight directions. At each radius $\in \{r_1, r_2, r_3, r_4\}$, along these eight different directions, we evaluate the mean of image intensities within a 3D cubic region (depicted here as colored 2D squares). The cubic widths $\{w_1, w_2, w_3, w_4\}$ are also shown for a set of regions.

increasing cube widths $w_i \in \{w_1, w_2, w_3, w_4\}$, respectively. In Fig. 2.2, we have shown the voxel \mathbf{x} at the center. The unit vector \mathbf{u} pointing toward the slice origin \mathbf{o}_z is shown in red. We also show the cubic regions over which we average intensities as colored boxes at the eight orientations and four radii. Although they are shown as 2D rectangles in the illustration, these regions are actually 3D cubes of sizes $w_i \times w_i \times w_i$,

CHAPTER 2. REPLICA

where the center voxel of the region lies at the designated rotations and radii within the axial slice of the voxel \mathbf{x} .

This yields a 32-dimensional descriptor of the context surrounding voxel \mathbf{x} at the highest resolution. In our experiments, we have used the values $w_1 = 3, w_2 = 5, w_3 = 7, w_4 = 9$ and $r_1 = 4, r_2 = 8, r_3 = 16, r_4 = 32$ voxels, respectively. These values were determined empirically. Since the head region is roughly spherical, this feature can disambiguate patches from different locations of the brain slice based on their near and far neighborhoods. We denote this feature vector as $\mathbf{v}(\mathbf{x})$. The feature vector at the finest level in the pyramid ($s = 1$) is the concatenated feature $\mathbf{f}^1(\mathbf{x}) = [\mathbf{p}^1(\mathbf{x}), \mathbf{q}^1(\mathbf{x}), \mathbf{v}(\mathbf{x})]$, which is used to train the final random forest regression stage to estimate the center voxel value $a_r^1(\mathbf{x})$.

2.2.2 Training a Random Forest

We train a random forest regressor RF_s at each level s to predict the voxel intensities in the target modality at that level from the feature vectors at that level (the orange, yellow, and green blocks in Fig. 3.1). A random forest regressor consists of an ensemble of regression trees [42] with each regression tree partitioning the space of features into regions based on a split at each node in the tree. Let $\Theta_q = \{[\mathbf{f}_1; v_1], \dots, [\mathbf{f}_t; v_t]\}$ be the set of all training sample pairs at a node q in the tree. Here, $\mathbf{f}_i \in \mathbb{R}^J$ denotes the feature vector (which can be any one of those in the multi-resolution pyramid as described above) and v_i denotes a value in the modality to be predicted (which is the

CHAPTER 2. REPLICA

value of the image to be predicted at the same level in the multi-resolution pyramid).

Let the mean of the target intensities of the samples at node q be \bar{v}_q . The nodal splits are determined during training by randomly selecting one third of the features (i.e., one third of the indices $j = 1, \dots, J$ of the feature vector) and then finding the feature j in this subset together with a corresponding threshold τ_j that together minimize a least squares criterion as defined in the following paragraph.

The squared distance (SD) from the mean of the target intensities at node q is given by

$$\text{SD}_q = \sum_{i=1}^t (v_i - \bar{v}_q)^2 \quad (2.1)$$

where t is the number of training samples at node q . This quantity is seen as a measure of compactness of the target intensities in a node. If feature j and threshold τ_j are selected to determine the split then the training data are split into a “left” subset $\Theta_{qL}(j, \tau_j) = \{[\mathbf{f}_i; v_i] | \forall i, f_{ij} \leq \tau_j\}$ and a “right” subset $\Theta_{qR}(j, \tau_j) = \{[\mathbf{f}_i; v_i] | \forall i, f_{ij} > \tau_j\}$. These training samples are then used in new left and right child nodes in the tree. In particular, each of these child nodes has its own compactness given by

$$\text{SD}_{qL}(j, \tau_j) = \sum_{i=1}^{t_L} (v_i - \bar{v}_{qL})^2 \quad (2.2)$$

$$\text{SD}_{qR}(j, \tau_j) = \sum_{i=1}^{t_R} (v_i - \bar{v}_{qR})^2, \quad (2.3)$$

where t_L and t_R are the number of samples in the left and the right child nodes respectively and the target intensities v_i are taken from their corresponding training

CHAPTER 2. REPLICA

samples. It is desirable that the data in each child node is maximally compact, which can be achieved if j and τ_j are chosen as

$$(\hat{j}, \hat{\tau}_j) = \arg \min_{j, \tau_j} \{SD_{qL}(j, \tau_j) + SD_{qR}(j, \tau_j)\}. \quad (2.4)$$

This is the least squares criterion that determines node splits during training to create each random forest regressor.

The growth of each tree is controlled by three factors: t_p , t_c , and ϵ . If a node has fewer than t_p training samples, it will not be split into child nodes (and it therefore becomes a leaf node). If the optimum split of a given node leads to one of the child nodes having fewer than t_c training samples, then it will not be split (and it likewise becomes a leaf node). We set $t_p = 2t_c$ and $t_c = 5$ in our experiments. If $SD_q - (SD_{qL} + SD_{qR}) < \epsilon SD_q$ then the node is not split (and it likewise becomes a leaf node). We use $\epsilon = 10^{-6}$ in our experiments.

A master list of training data is created from the atlas data by sampling the features in all tissues making sure that abnormal tissues (like white matter lesions) are well-represented. Each RF regressor consists of sixty trees where each tree is learned from bootstrapped training data, where bootstrapping is carried out by randomly choosing $N = 1 \times 10^5$ training samples (with replacement) for each tree. Each trained tree contains the feature j and threshold τ_j at each non-leaf node and the average value of the target intensity at each leaf node. To use an RF regressor, the same

CHAPTER 2. REPLICA

feature vector is “fed” to each root node and each tree is traversed according to the stored feature indices and thresholds until a leaf node is reached whereupon the tree provides an intensity. The average intensity of all sixty trees is then formed as the output of the regressor.

We summarize the training stages in the Algorithm 1.

Algorithm 1 REPLICA: Training

```
1: Data: Co-registered pair of atlas images at scale  $s = 1$  (highest resolution),  $a_1^1$ 
   and  $a_2^1$ , rigidly aligned to the MNI space
2:
3: for scale  $s = 2 : S$  do
4:    $a_1^s = \text{GaussFilterAndDownsample2x}(a_1^{s-1})$ 
5:    $a_2^s = \text{GaussFilterAndDownsample2x}(a_2^{s-1})$ 
6: end for
7: for scale  $s = 1 : S$  do
8:   if scale  $s == S$  then
9:     Extract  $3 \times 3 \times 3$  patches from  $a_1^s$ 
10:    Extract corresponding voxel intensities from  $a_2^s$ , and create training data
11:    Train a random forest  $\text{RF}_s$  on the training data
12:  else if scale  $s > 1$  and  $s < S$  then
13:     $a_2^s \uparrow = \text{upsample2x}(a_2^{s+1})$ 
14:    Extract  $3 \times 3 \times 3$  patches from  $a_1^s$ 
15:    Extract corresponding  $3 \times 3 \times 3$  patches from  $a_2^s \uparrow$ 
16:    Concatenate corresponding patches to create a joint feature vector set
17:    Extract corresponding voxel intensities from  $a_2^s$  and create training data
18:    Train a random forest  $\text{RF}_s$  using the training data
19:  else if scale  $s == 1$  then
20:     $a_2^s \uparrow = \text{upsample2x}(a_2^{s+1})$ 
21:    Extract  $3 \times 3 \times 3$  patches from  $a_1^s$ 
22:    Extract high resolution context descriptors from  $a_1^s$ 
23:    Extract corresponding  $3 \times 3 \times 3$  patches from  $a_2^s \uparrow$ 
24:    Concatenate all to create a joint feature vector set
25:    Extract corresponding voxel intensities from  $a_2^s$ , and create training data
26:    Train a random forest  $\text{RF}_s$  using the training data
27:  end if
28: end for
```

2.2.3 Predicting a New Image

Given a subject image set, a Gaussian image pyramid is constructed and at each level s the subject image set is $\mathcal{B}_s = \{b_1^s, b_2^s, \dots, b_m^s\}$ and relevant features are extracted. Starting from the coarsest level ($s = 3$), RF_s is applied to synthesize the target modality at the lowest resolution \hat{b}_r^S . This step is depicted in stage (d) of Fig. 3.1. For levels $s < S, s \neq 1$, \hat{b}_r^S is upsampled to the next level to create $\hat{b}_r^{S\uparrow}$, which is a synthetic, up-sampled low resolution image. Feature vectors $\mathbf{f}^s(\mathbf{x}) = [\mathbf{p}^s(\mathbf{x}), \mathbf{q}^s(\mathbf{x})]$ are calculated at each voxel \mathbf{x} at this level and RF_s is applied to these to generate \hat{b}_r^{S-1} (see stage (e) in Fig. 3.1). This process continues until $s = 1$. At $s = 1$, the highest available resolution, feature vectors $\mathbf{f}^1(\mathbf{x}) = [\mathbf{p}^1(\mathbf{x}), \mathbf{q}^1(\mathbf{x}), \mathbf{v}(\mathbf{x})]$ which includes the high resolution context descriptor $\mathbf{v}(\mathbf{x})$ are calculated. The trained random forest RF_1 is applied to produce the final high resolution synthetic image \hat{b}_r^1 (stage (f) in Fig. 3.1). The prediction algorithm is summarized in Algorithm 2.

We have provided a very general description of the REPLICA image synthesis pipeline. Depending on the complexity of the application we might use a subset of this pipeline. For example, when synthesizing images for which the input images are already skull-stripped, the intensity mapping does not need the entire multi-resolution treatment as the high resolution features are sufficient. REPLICA has a number of free parameters that can be tuned to improve the resulting synthesis. We performed extensive parameter selection experiments, the results of which are available in Appendix 2.A. These parameters were set as follows, (a) number of trees ($= 60$),

Algorithm 2 REPLICA: Prediction

```

1: Data: Subject image  $b_1^1$  rigidly aligned to the MNI space and trained random
   forests  $\text{RF}_s$ , for  $s \in \{1, \dots, S\}$ 
2:
3: for scale  $s = 2 : S$  do
4:    $b_1^s = \text{GaussFilterAndDownsample2x}(a_1^{s-1})$ 
5: end for
6: for scale  $s = 1 : S$  do
7:   if scale  $s == S$  then
8:     Extract  $3 \times 3 \times 3$  patches from  $b_1^s$ 
9:     Predict  $\hat{b}_2^s$  by applying random forest  $\text{RF}_s$ 
10:  else if scale  $s < S$  and  $s > 1$  then
11:     $\hat{b}_2^s \uparrow = \text{upsample2x}(\hat{b}_2^{s+1})$ 
12:    Extract  $3 \times 3 \times 3$  patches from  $b_1^s$ 
13:    Extract corresponding  $3 \times 3 \times 3$  patches from  $\hat{b}_2^s \uparrow$ 
14:    Concatenate corresponding patches to create a joint feature vector set
15:    Predict  $\hat{b}_2^s$  by applying random forest  $\text{RF}_s$ 
16:  else if scale  $s == 1$  then
17:     $\hat{b}_2^s \uparrow = \text{upsample2x}(b_2^{s+1})$ 
18:    Extract  $3 \times 3 \times 3$  patches from  $b_1^s$ 
19:    Extract high resolution context descriptors from  $b_1^s$ 
20:    Extract corresponding  $3 \times 3 \times 3$  patches from  $\hat{b}_2^s \uparrow$ 
21:    Concatenate all to create a joint feature vector set
22:    Predict  $\hat{b}_2^s$  by applying random forest  $\text{RF}_s$ 
23:  end if
24: end for

```

(b) number of samples in a leaf node ($= 5$), (c) size of local 3D patch ($= 3 \times 3 \times 3$), (d) number of individuals in the atlas ($= 1$), (e) use of alternate atlas images (does not affect synthesis), and (f) use of context and multi-scale features, the results of which are shown in Section 2.3.2. For the remaining experiments, please refer to the Appendix 2.A.

2.3 Results

In this section, we describe a variety of REPLICA applications. In Section 2.3.1 we describe results of T_2w synthesis using REPLICA. In Section 2.3.2 we describe a comparatively more challenging task of synthesizing full head T_2w images (i.e., not skull-stripped) and show that REPLICA in its complete avatar (i.e., using the multi-resolution framework) is able to achieve it better than competing algorithms. In Section 2.3.3 we demonstrate the use of REPLICA to synthesize FLAIR images and show that image segmentation using synthetic FLAIRs is comparable to that achieved using real FLAIRs. In Section 2.3.4 we show intensity standardization between SPGR and MPRAGE using REPLICA. Finally in Section 2.3.5, we demonstrate example-based super-resolution using REPLICA.

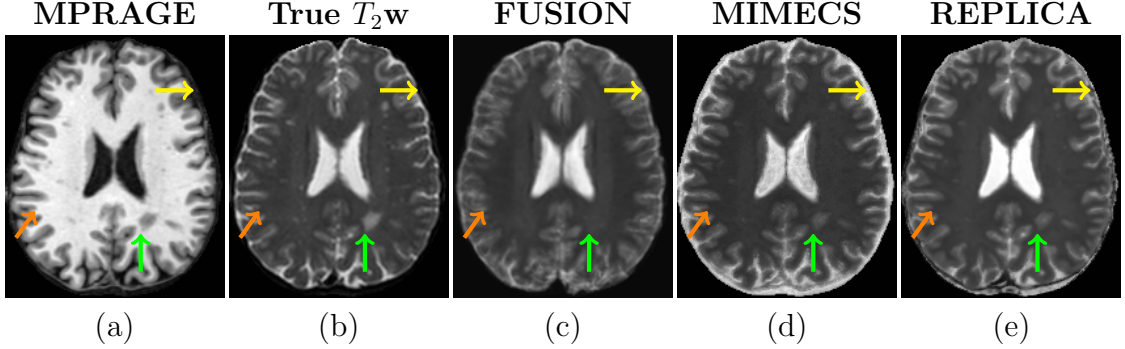


Figure 2.3: Shown are (a) the input MPRAGE image, (b) the true T_2w image, and the synthesis results of (c) FUSION [3], (d) MIMECS [4], and (e) REPLICA (our method). The lesion (green arrow) and the cortex (orange arrow) in the true image are correctly synthesized by MIMECS and REPLICA, but not by FUSION. The dark boundary just outside the cerebral tissue (yellow arrow) is incorrectly synthesized as bright by MIMECS, but not by FUSION and REPLICA.

2.3.1 Synthesis of T_2w Images

In this experiment, we synthesized T_2w images from skull-stripped T_1w MPRAGE images taken from the Multi-Modal MRI Reproducibility Resource (MMRR) data [50]. The MMRR data consists of 21 subjects, each with two imaging sessions acquired within an hour of each other. T_2w images can be used as registration targets while performing distortion correction on echo-planar images (EPI) images and also as input to lesion segmentation algorithms. Therefore, if T_2w images are absent, we can use REPLICA to synthesize them. We compared REPLICA to the MIMECS method [4] and a multi-atlas registration and intensity fusion method which we refer to as FUSION [3]. We used the NiftyReg affine registration followed by free-form deformation registration algorithm to drive the multi-atlas registration [51, 52] and implemented the intensity fusion as described in [3]. We used data from one randomly

CHAPTER 2. REPLICA

chosen subject as training for REPLICA and MIMECS and one image each from five subjects as the atlases for FUSION. We set the parameters for FUSION to $\beta = 0.5$ (weighting parameter) and $\kappa = 4$ (use the top 4 best patch matches to fuse); refer to [3] for more details. For the remaining $16 \times 2 = 32$ MPRAGE images, we synthesized T_2w images using all three methods. The input MPRAGE images were intensity standardized by scaling such that the white matter peak intensity in the histogram is 1. Synthesis of skull-stripped T_2w images can be achieved with an intensity transformation that is captured well using just the high resolution features. Therefore we do not use the entire multi-resolution framework for this application but only use the high resolution context descriptors and a local patch ($\mathbf{f}(\mathbf{x}) = [\mathbf{p}^1(\mathbf{x}), \mathbf{v}(\mathbf{x})]$).

For evaluation of synthesis quality, we used PSNR (peak signal to noise ratio), which is a mean squared error-based metric, UQI (universal quality index) [53], and SSIM (structural similarity) [54], with respect to the ground truth images. UQI and SSIM are more sensitive to perceptual differences in image structure than PSNR since they take into account properties of the human visual system. For both UQI and SSIM, a value of 1 indicates that the images are equal to each other; otherwise their values lie between 0 and 1.

Visual comparison to ground truth images is one of the primary indications of whether synthesis has been successful or not. Visually disparate synthetic images are likely to generate different outputs when fed into image processing algorithms. We can see from the results in Table 2.1 that REPLICA performs significantly better than the

Table 2.1: Mean and standard deviation (Std. Dev.) of the PSNR, UQI, and SSIM values for synthesis of T_2w images from 32 MPRAGE scans.

	PSNR	UQI	SSIM
	Mean (Std)	Mean (Std.)	Mean (Std)
FUSION	52.73 (2.78)*	0.78 (0.02)	0.82 (0.02)
MIMECS	36.13 (2.23)	0.78 (0.02)	0.77 (0.02)
REPLICA	50.73 (2.67)	0.89 (0.02)*	0.87 (0.02)*

* Statistically significantly better than either of the other two methods ($p < 0.01$) using a right-tailed test.

other methods for all metrics except PSNR. Figure 4.1 shows the results for all three methods along with the true T_2w image. FUSION ranks highest for PSNR, which can be explained by the fact that unlike the cortex, large, homogeneous regions of white matter have been reproduced quite well. However, it also produces anatomically incorrect images, especially in the presence of abnormal tissue anatomy (lesions for example) and the cortex (see Fig. 4.1(c)). MIMECS has a lower PSNR primarily due to boundary voxels, which are mostly skull voxels that were incorrectly synthesized as CSF (see Fig. 4.1(d)). Overall, REPLICA produces an image that is visually closest to the true T_2w image and has the highest UQI and SSIM values.

2.3.2 Synthesis of Whole Head Images

Synthesis of whole head images, i.e., head images that are not skull-stripped, is challenging due to the high variability in tissues and tissue intensities. The intensity transformation that needs to be learned is highly nonlinear and is generally one-to-many, especially if small patches are used as features. Small patches are useful if there are

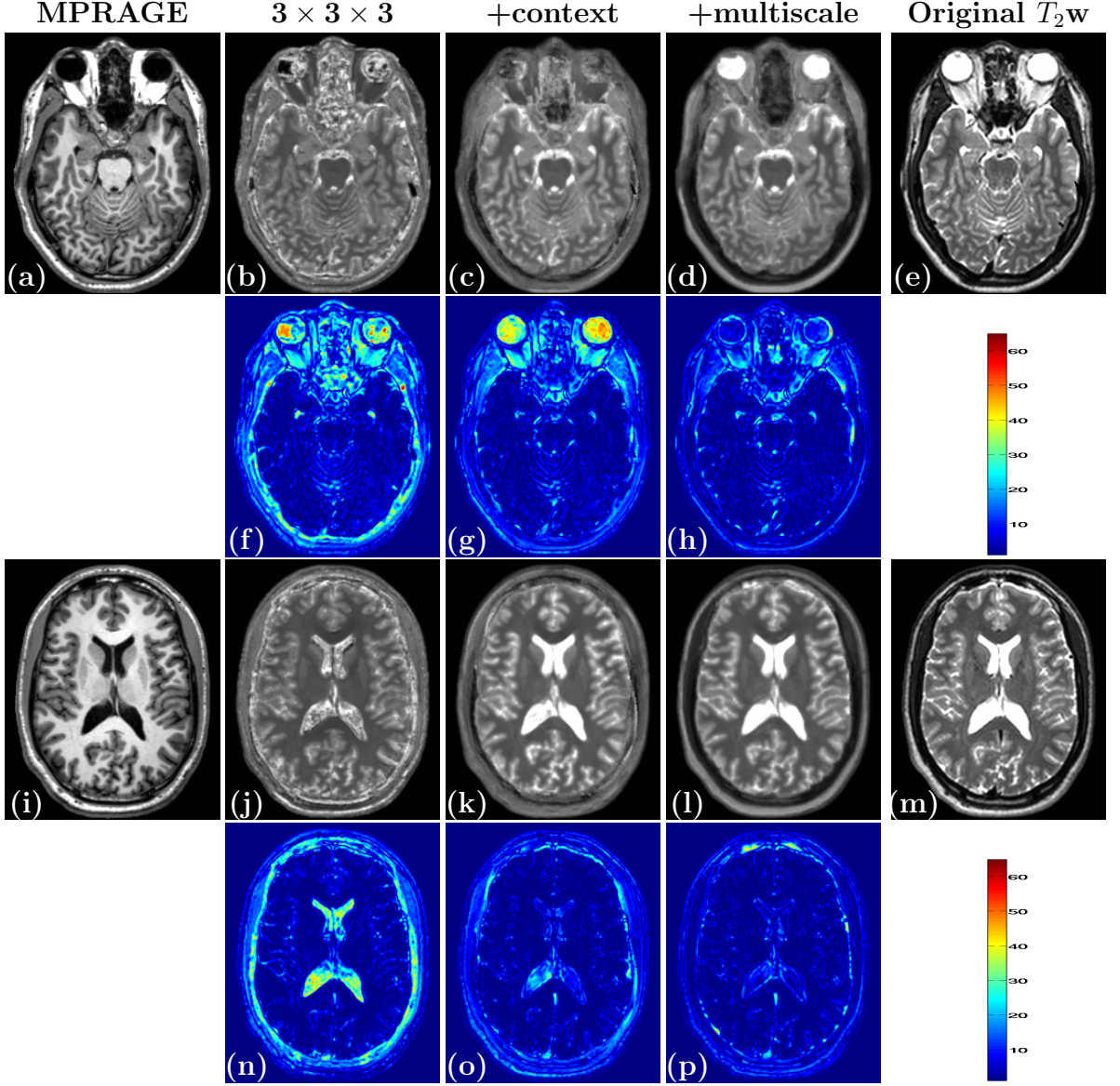


Figure 2.4: (a) Original input MPRAGE, (b) REPLICA T_2w synthesis using $3 \times 3 \times 3$ patch as feature vector, followed by (c) with additional high resolution context descriptor feature, and (d) using the full, multi-resolution REPLICA framework. (e) shows the ground truth T_2w image. In the next row we have the corresponding difference images with respect to the real T_2w image in (f), (g) and (h) respectively. It is clear that using multi-resolution REPLICA produces a higher quality synthesis for the challenging task of synthesis of full-head images. (i)-(p) Show the same images for a more superior slice. The maximum intensity in these images is 255. The difference images have a maximum intensity around 60.

CHAPTER 2. REPLICA

multiple input modalities capable of providing the necessary information to synthesize extra-cerebral tissues, as [37] demonstrated by synthesizing CT images from input ultrashort echo time (UTE) images. The two UTE images together provide enough intensity information to differentiate the bone from soft tissues. In general, local patch features can be adequate to accurately synthesize skull-stripped images, but additional context features are needed when extra-cerebral tissues are included. Multi-resolution and context features, as described in Section 2.2.1, provide additional information about the location of a voxel within the brain, enabling better synthesis of these tissues. This experiment demonstrates the impact of these additional features. We use the T_1 w MPRAGE images from the MMRR dataset [50] and synthesize corresponding full-head T_2 w images. We used data from a randomly selected subject for training. The atlas and subject images have the following specifications:

a_1 : MPRAGE image (3 T, TR = 6.7 ms, TE = 3.1 ms, TI = 842 ms, $1.0 \times 1.0 \times 1.2$ mm³ voxel size),

a_2 : T_2 w image from the second echo of a DSE (3 T, TR = 6653 ms, TE₁ = 30 ms, TE₂ = 80 ms, $0.82 \times 0.82 \times 1.5$ mm³ voxel size).

b_1 : MPRAGE image (3 T, TR = 6.7 ms, TE = 3.1 ms, TI = 842 ms, $1.0 \times 1.0 \times 1.2$ mm³ voxel size)

REPLICA parameters were set as follows: number of trees = 60, and $t_c = 5$, $t_p = 2t_c$, and $\epsilon = 10^{-6}$.

CHAPTER 2. REPLICA

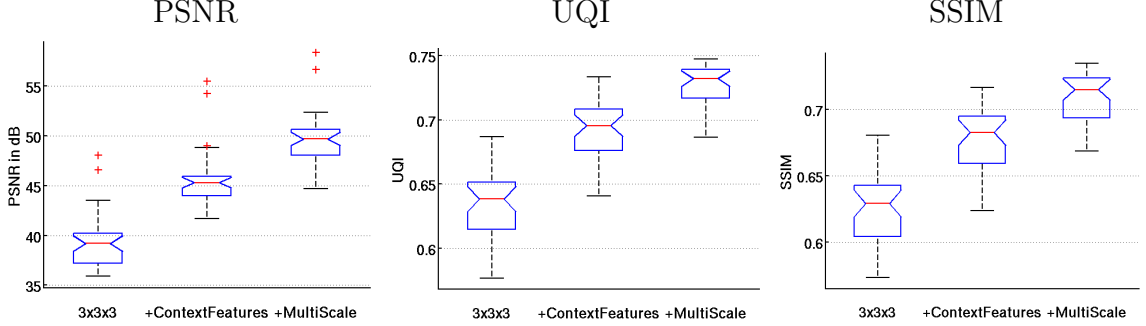


Figure 2.5: PSNR, UQI, and SSIM as functions of the features and multi-resolution framework of REPLICA.

We ran REPLICA with three different settings for input features: (a) only local $3 \times 3 \times 3$ patches, (b) local patches + context features, and (c) local patches + context features + multi-resolution framework. Using each of these settings, we synthesized $20 \times 2 = 40$ synthetic images. Figure 2.4 shows a progression of REPLICA results for a synthetic T_2w image of a subject from the corresponding MPRAGE with both brain and non-brain tissues using a $3 \times 3 \times 3$ local patch (Figs. 2.4 (b), (j)), a combination of the local patch and high resolution context descriptor (Figs. 2.4 (c), (k)), and the full REPLICA multi-resolution framework (Figs. 2.4 (d), (l)). Figures 2.4 (f), (g), (h), (n), (o), (p) also show the corresponding difference images with the ground truth T_2w image in Figs. 2.4 (e), (m). In this figure we show two slices, one from the inferior head region with a high variability of tissue intensities and structures, and another from a slightly superior region with more structured regions. This qualitative comparison reveals that the full multi-resolution framework works best.

For a quantitative comparison, we calculated PSNR, UQI, and SSIM from the 40 synthetic images to compare the effect of these feature settings. UQI and SSIM

CHAPTER 2. REPLICA

Table 2.2: Mean and standard deviation (Std. Dev.) of the PSNR, UQI, and SSIM values for synthesis of T_2w images from 32 MPRAGE scans.

	PSNR	UQI	SSIM
	Mean (Std)	Mean (Std.)	Mean (Std)
$3 \times 3 \times 3$ -patches	39.34 (2.70)	0.63 (0.02)	0.62 (0.02)
+context	45.68 (2.89)	0.69 (0.02)	0.67 (0.02)
+multiscale	49.69 (2.65)*	0.73 (0.02)*	0.70 (0.02)*

* Statistically significantly better than either of the other two methods ($p < 0.01$) using a right-tailed test.

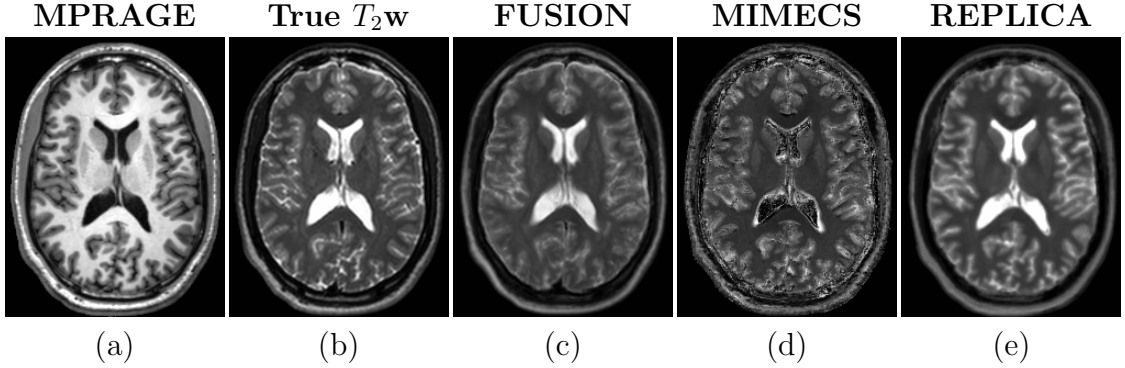


Figure 2.6: (a) The input MPRAGE image, (b) the real T_2w image, (c) FUSION result, (d) MIMECS result, (e) REPLICA result. Note the synthesis errors in the cortex for FUSION and in the ventricles for MIMECS.

were computed for individual slices and averaged over the number of slices. Boxplots for these metrics are shown in Fig. 2.5. These plots show quantitatively that the multi-resolution version of REPLICA is statistically significantly better ($p < 0.01$, one-tailed t-test) than the other two approaches in synthesizing T_2w images from MPRAGE images when the whole head is present. These values are tabulated in Table 2.2.

Further, we qualitatively compared REPLICA against MIMECS and FUSION for this task. The results are shown in Fig. 2.6. At first glance the FUSION result (Fig. 2.6

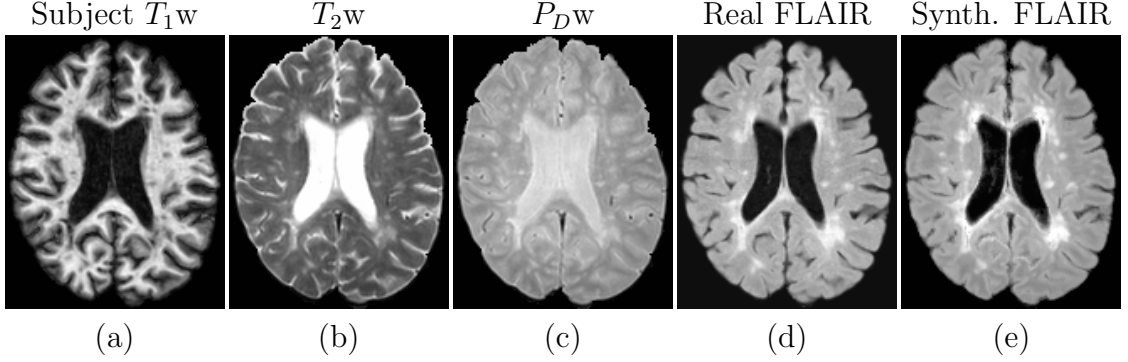


Figure 2.7: Subject input images along with the synthetic and true FLAIR images.

(c)) looks appealing, but has similar errors in the cortex region as for skull-stripped brain synthesis in Section 2.3.1. MIMECS uses a small $3 \times 3 \times 3$ -sized patch and is unable to disambiguate between skull and CSF resulting in large errors, especially in the ventricles (Fig. 2.6 (d)). The REPLICA result (Fig. 2.6(e)) looks visually closer to the truth, although appears smoother in some regions due to dependence on low-resolution information coming from the lower levels of the multi-resolution framework.

This experiment showcases the capability of REPLICA to handle complex image synthesis scenarios, with limited intensity information at its disposal.

2.3.3 Synthesis of FLAIR Images

In this experiment, we used REPLICA to synthesize a FLAIR image from T_1w , T_2w , and P_Dw images. The FLAIR sequence is routinely used to image patients with multiple sclerosis (MS) and other diseases. It is particularly useful to visualize white matter lesions (WML) that are observed in MS patients. White matter lesions

CHAPTER 2. REPLICA

appear hyperintense in FLAIR images and can be easily delineated using automated segmentation algorithms. The FLAIR sequence needs a long TI value and, therefore, is generally acquired at a lower resolution for quicker scan time. FLAIR images also suffer from artifacts, which result in hyperintensities that can be mistaken for lesions [16]. Missing FLAIR images can also pose hurdles in lesion segmentation, as most leading lesion segmentation algorithms use FLAIR as an input modality [55–57]. Synthesizing missing FLAIR images can help avoid these issues and enable segmentation and further image analysis.

The atlas set we used for this experiment was:

a_1 : MPRAGE image (3 T, TR = 10.3 ms, TE = 6 ms, $0.82 \times 0.82 \times 1.17$ mm³ voxel size),

a_2 : T_2 w image from the second echo of a DSE (3 T, TR = 4177 ms, TE₁ = 3.41 ms, TE₂ = 80 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size),

a_3 : P_{Dw} from the first echo of a DSE (3 T, TR = 4177 ms, TE₁ = 3.41 ms, TE₂ = 80 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size),

a_4 : FLAIR (3 T, TI = 11000 ms, TE = 68 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size)

The subject image set was:

b_1 : MPRAGE image (3 T, TR = 10.3 ms, TE = 6 ms, $0.82 \times 0.82 \times 1.17$ mm³ voxel size)

CHAPTER 2. REPLICA

b_2 : P_D w image from the first echo of a DSE (3 T, TR = 4177 ms, TE₁ = 3.41 ms,

TE₂ = 80 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size)

b_3 : T_2 w image from the second echo of a DSE (3 T, TR = 4177 ms, TE₁ = 3.41 ms,

TE₂ = 80 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size).

All the modalities were registered and resampled to the MPRAGE image. This experiment did not need the multi-resolution framework as the image analysis takes place on skull-stripped [58] images. The input MPRAGE, and DSE images were intensity standardized by scaling such that the white matter peak intensity in the histogram was 1. We also tweaked the prediction of the decision trees in the random forest ensemble in this experiment. Instead of calculating the mean of the sample predictions in a single leaf, we calculated the mode. Calculating the mean resulted in oversmooth images, especially at the lesion-WM boundaries, which resulted in overestimation of lesion size by the segmentation algorithm. Using the mode results in crisper edges and better lesion segmentation. Input images along with the real and synthetic FLAIR images are shown in Fig. 2.7.

We used our in-house MS patient image dataset with 125 images belonging to 84 subjects, with some subjects having images acquired longitudinally. We compared the synthesized images with existing true images using image similarity metrics (see Table 2.3). These values indicate that the synthetic FLAIR images are visually similar to the corresponding real FLAIR images. FLAIR synthesis results using FUSION (Figs. 2.8(c), (h)), MIMECS (Fig. 2.8(d), (i)) and REPLICA (Fig. 2.8(e),

CHAPTER 2. REPLICA

Table 2.3: Mean (Std. Dev.) of PSNR (in decibels), UQI, and SSIM values over 125 FLAIR images synthesized by REPLICA.

	PSNR	UQI	SSIM
	Mean (Std)	Mean (Std.)	Mean (Std)
FUSION	18.05 (0.47)	0.57 (0.04)	0.60 (0.04)
MIMECS	16.22 (3.10)	0.81 (0.03)	0.57 (0.05)
REPLICA	21.73 (1.95)*	0.84 (0.03)*	0.81 (0.03)*

* Statistically significantly better than either of the other two methods ($p < 0.01$) using a right-tailed test.

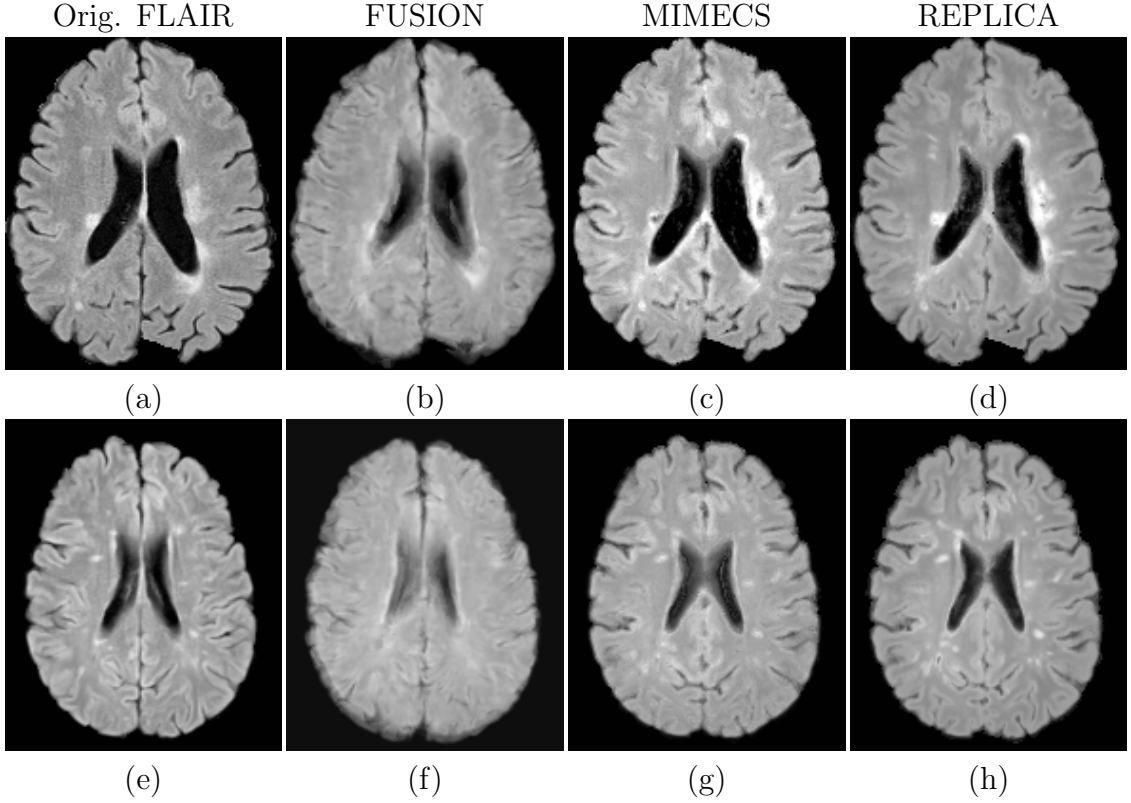


Figure 2.8: Visual comparison of FUSION, MIMECS and REPLICA for the FLAIR synthesis task. The two rows show images from different subjects. (a) and (e) real FLAIRs, (b) and (f) FUSION results, (c) and (g) MIMECS results, (d) and (h) REPLICA results.

(j)) for two different subjects can be compared visually with the corresponding real FLAIR images. FUSION was run with the same parameters as for T_2w synthesis and

CHAPTER 2. REPLICA

Table 2.4: Mean (Std. Dev.) of Dice coefficients based on LesionTOADS segmentation of the real FLAIR and synthetic FLAIR over 125 images.

Algorithm	WM	GM	CSF	WML
LesionTOADS	0.97 (0.01)	0.99 (0.0005)	0.96 (0.01)	0.46 (0.22)

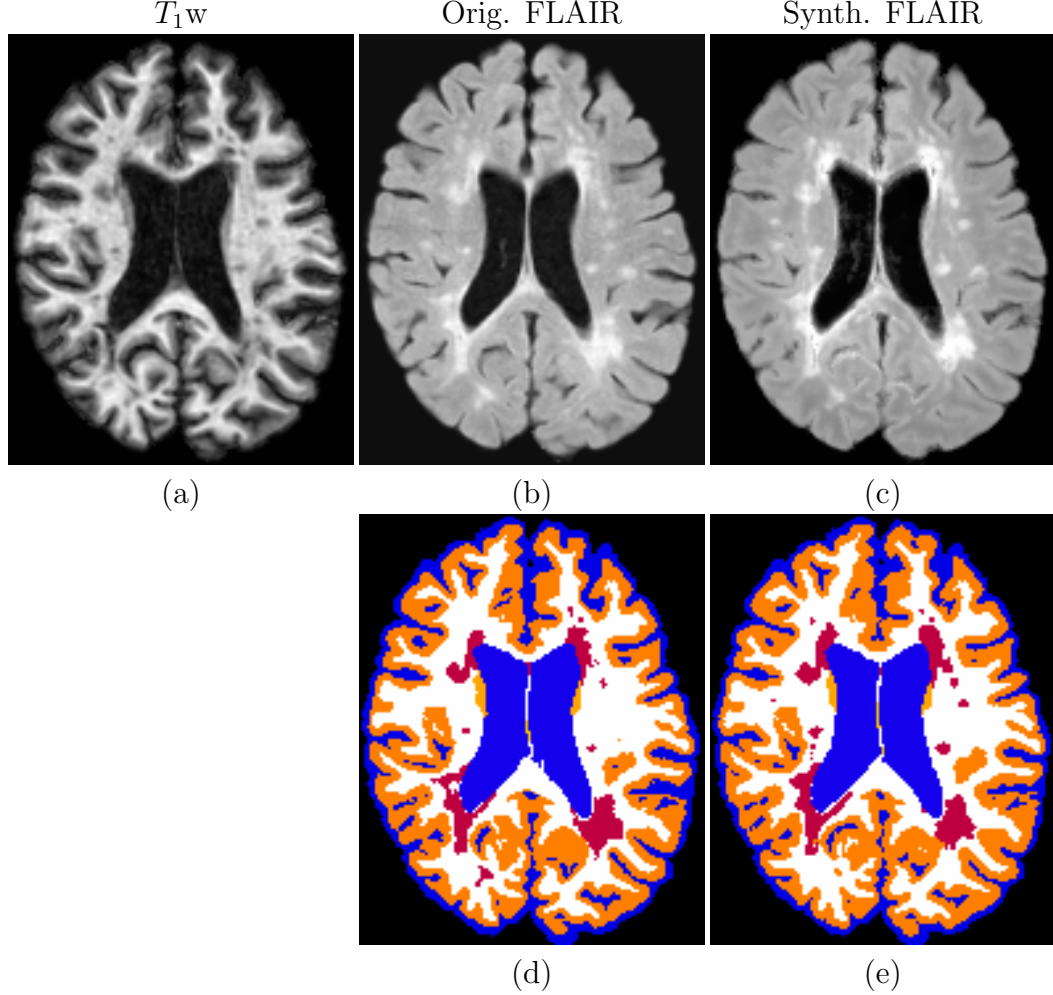


Figure 2.9: (a) Input MPRAGE for LesionTOADS segmentation, (b) original FLAIR, (c) synthetic FLAIR generated from T_1w , T_2w and P_Dw images, (d) LesionTOADS segmentation using real FLAIR + MPRAGE, (e) LesionTOADS segmentation using synthetic FLAIR + MPRAGE.

it is unable to faithfully construct a synthetic image that is close enough to the ground truth. FUSION is based on registration and intensity fusion of multiple atlas images;

CHAPTER 2. REPLICA

it cannot synthesize lesion intensities at lesion locations if those intensities are not present in the atlas images at those exact locations. The presence of lesions also affects the quality of the registrations itself, thus leading to a worse than expected result. MIMECS works better than FUSION but has errors synthesizing large lesion areas and very small lesions. Overall, REPLICA produces the most visually acceptable synthetic FLAIR image. Quantitative comparison with MIMECS and FUSION shown in Table 2.3 confirms that REPLICA produces a better synthetic FLAIR.

Next, we used the synthetic FLAIR images as inputs to a tissue segmentation algorithm. If synthesis has been done correctly, the segmentation algorithm should behave identically for real and synthetic images as inputs. To test this, we used the LesionTOADS algorithm [55]. LesionTOADS uses a T_1w image and a corresponding FLAIR image to generate a multi-class, topologically correct segmentation in the presence of lesions. We compared the overlap of segmentations obtained using synthetic FLAIR images to those obtained using real FLAIR images in terms of Dice coefficients. The Dice coefficients for WM, GM, CSF, and WML classes are shown in Table 2.4. Figure 2.9 shows the segmentations by LesionTOADS on real and synthetic FLAIR images. The overlap is high for WM, GM, and CSF; however it is relatively low for the WML class. The lesions are small and diffuse and even a small difference in the overlap can cause a low value for the Dice coefficient [56]. So we looked at the lesion volumes as provided by LesionTOADS for real and synthetic FLAIR images.

To understand how different the lesion volumes are for the synthetic images

CHAPTER 2. REPLICA

as compared to the real images, we created a Bland-Altman plot [59] for these measurements (see Fig. 2.10). If \mathbf{y}_1 and \mathbf{y}_2 are two measurements by two different methods consisting of n samples each, then the Bland-Altman plot is a scatter plot of $\mathbf{y}_1 - \mathbf{y}_2$ versus $(\mathbf{y}_1 + \mathbf{y}_2)/2$. The measurements are considered to be interchangeable if 0 lies within $\pm 1.96\sigma$ where σ is the standard deviation of $\mathbf{y}_1 - \mathbf{y}_2$. Figure 2.10 shows the Bland-Altman plot where \mathbf{y}_1 are the lesion volumes for synthetic FLAIR images and \mathbf{y}_2 are the lesion volumes for real FLAIR images, as produced, in both cases, by LesionTOADS. Even though the difference $(\mathbf{y}_1 - \mathbf{y}_2)$ is not zero-mean ($p > 0.05$ using a one sample t-test), both the measurements can be used interchangeably because 0 lies within $\pm 1.96\sigma$ in the plot [59].

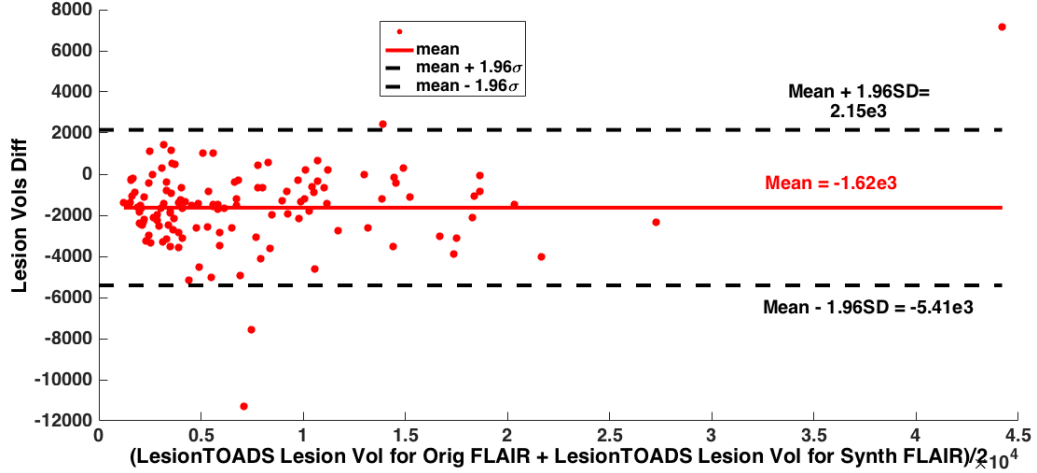


Figure 2.10: A Bland-Altman plot of lesion volumes for synthetic FLAIRs vs lesion volumes of real FLAIRs. Since zero lies within the $\pm 1.96\sigma$ range, these two measurements of the same quantity are interchangeable.

Next, we compared lesion volumes with those estimated by an expert manual rater. We can consider volumes produced by a manual rater to be an additional

CHAPTER 2. REPLICA

method of measurement, and compare it with LesionTOADS lesion volumes on real and synthetic FLAIRs in a Bland-Altman plot in Fig. 2.11. In Fig. 2.11, we show the Bland-Altman plots of manual lesion volumes vs LesionTOADS lesion volumes using real FLAIRs (blue) and LesionTOADS lesion volumes using synthetic FLAIRs (red). The bold blue and red lines depict the mean value whereas the dashed lines depict their respective $\pm 1.96\sigma$ ranges. This plot shows that LesionTOADS lesion volumes of real FLAIR images can be interchangeably used with the manual lesion volumes, validating the choice of LesionTOADS as a segmentation algorithm. More importantly, the LesionTOADS lesion volumes from the synthetic FLAIR images are marginally closer to the manual volumes as evinced by the proximity of the mean.

Manual volumes were obtained from real FLAIRs, so ideally LesionTOADS volumes on real FLAIRs should have been closer to the manual volumes. However, we observe the opposite. A speculative explanation for this observation is as follows: The process of manual delineation relies on the ability of the human rater to account for the intensity variation observed in real FLAIRs and provide a consistent segmentation across subjects. Synthetic FLAIRs on other hand are already consistent in their intensity characteristics because all of them are generated from a fixed set of atlas FLAIRs. Thus LesionTOADS results on synthetic FLAIRs are less affected by the variations than real FLAIRs, thus bringing them closer to the manual delineations. A manually segmented set of synthetic FLAIRs would help us in validating this assertion.

Synthesis of FLAIR images is an important application and one that has not been

demonstrated successfully before. REPLICA is the first algorithm to create usable FLAIRs from input T_1w , T_2w , and P_Dw images.

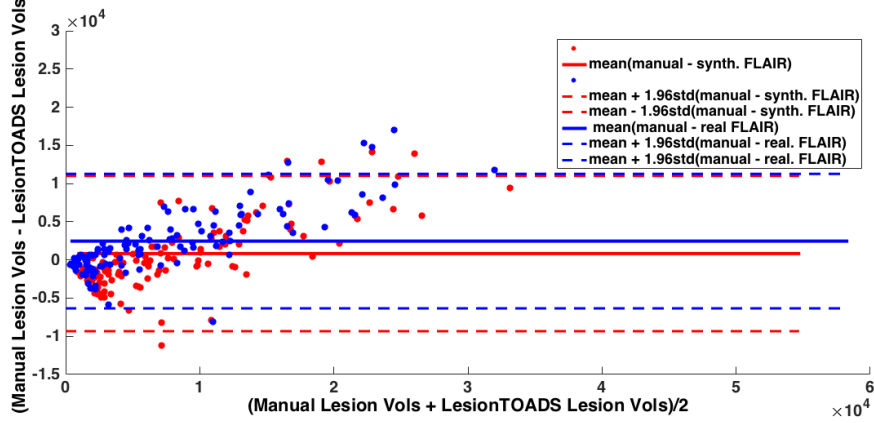


Figure 2.11: The red Bland-Altman plot is for lesion volumes as measured by a manual rater versus LesionTOADS lesion volumes on synthetic FLAIRs. The blue Bland-Altman plot is for lesion volumes as measured by a manual rater versus LesionTOADS lesion volumes on real FLAIRs. This plot shows that the lesion volumes obtained using the synthetic FLAIRs are marginally closer to those found by the manual rater as the red mean line is closer to zero.

2.3.4 Intensity Standardization

Tissue segmentation and cortical reconstruction in MRI are generally reliant on T_1w images acquired with pulse sequences like MPRAGE and SPGR [10, 60, 61]. However, most segmentation algorithms are not robust to variabilities in the T_1w image contrasts [62]. Intensity standardization of different contrasts has been proposed to alleviate this problem [2, 62, 63]. Image synthesis can be used to standardize intensities by creating synthetic, standardized images from given images. These synthetic images can belong to a given, reference modality, on which the algorithm behavior is well-

CHAPTER 2. REPLICA

understood. We demonstrate such an intensity standardization application using the Baltimore Longitudinal Study of Aging (BLSA) dataset [64]. In this experiment, we used a subset of the dataset, consisting of 82 scans of 60 subjects, some of which are longitudinal acquisitions. Each scanning session was carried out on a Philips 1.5 T scanner and has an SPGR (see Fig. 2.12(a)) and an MPRAGE (see Fig. 2.12(b)) acquisition from the same session. We chose this dataset as it can mimic the scenario of the multi-site data acquisition where protocols differ across the sites. The input SPGR images were intensity standardized by scaling such that the white matter peak intensity in the histogram is 1. We use an in-house implementation of the probabilistic atlas-driven, EM-based segmentation algorithm [61], which we refer to as AtlasEM. AtlasEM provides us with a 4-class segmentation, the classes being sulcal CSF, GM, WM, and ventricles. We segmented the SPGR and the MPRAGE images using AtlasEM. In an ideal scenario, where the algorithm is impartial to the underlying T_1w input, the segmentations should be identical. However, as we can observe in Figs. 2.12(e), (f), the segmentations are quite different. We used REPLICA to generate a synthetic MPRAGE from input SPGR images, and ran AtlasEM segmentations on the synthetic MPRAGEs. The atlas set used for this experiment was:

a_1 : SPGR image (1.5 T, TR = 35 ms, TE = 5 ms, $\alpha = 45^\circ$, $0.938 \times 0.938 \times 1.5 \text{ mm}^3$ voxel size),

a_2 : MPRAGE (1.5 T, TR = 6.92 ms, TE = 3.4 ms, $\alpha = 8^\circ$, $0.938 \times 0.938 \times 1.5 \text{ mm}^3$ voxel size).

CHAPTER 2. REPLICA

The subject images were:

b_1 : SPGR image (1.5 T, TR = 35 ms, TE = 5 ms, $\alpha = 45^\circ$, $0.938 \times 0.938 \times 1.5 \text{ mm}^3$ voxel size).

The synthesis did not need the multi-resolution framework as we worked with skull-stripped images because the final segmentation takes place on skull-stripped [58] images. We used the synthetic MPAGE as an input for AtlasEM. Our goal is to show that the segmentations are closer to those obtained by a real MPAGE. Figure 2.12(c) shows the REPLICA-generated synthetic MPAGE and Fig. 2.12(f) shows the segmentation for the same. Visually, it is closer to the segmentation obtained from the real MPAGE Fig.(2.12(e)), especially at the CSF-GM interface.

We also looked at tissue volumes provided by AtlasEM on all three sets of images, SPGR, MPAGE, and synthetic MPAGE. As stated earlier, if the AtlasEM algorithm were robust to the input modality, the tissue volumes for a particular tissue should be identical for SPGR and MPAGE. In Fig. 2.13 (a)–(d), we show scatter plots for the tissue volumes obtained on real MPAGES (x -axis) and those obtained for SPGRs (blue) and synthetic MPAGES (red). We also show the least square line fits to the scatter plots (blue for SPGR, red for synthetic MPAGE). In the ideal scenario, the least square line fits should be close to the $x = y$ line. We can see that for sulcal CSF (Fig. 2.13(a)), GM (Fig. 2.13(c)), and WM (Fig. 2.13(d)), synthetic MPAGE is closer to the identity line than SPGR. The ventricle volumes are similar in both the modalities (Fig. 2.13(b))

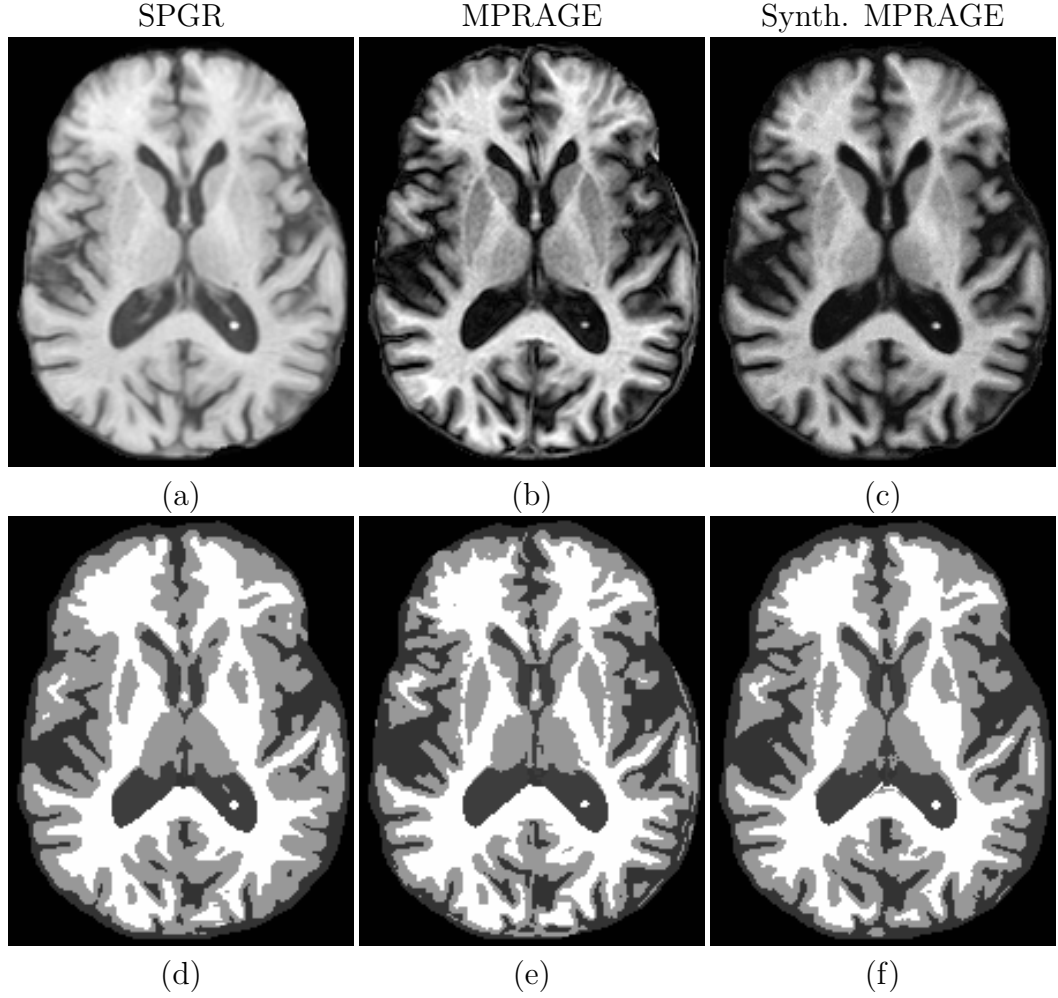


Figure 2.12: (a) SPGR, (b) MPRAGE, (c) Synthetic MPRAGE generated from the SPGR. (d–f) their respective AtlasEM segmentations. The SPGR segmentation is quite different from the MPRAGE one. The synthetic MPRAGE tries to bridge this difference.

This experiment is a validation scenario for a large multi-center, multi-scanner MRI study, where images obtained from two different scanners can lead to differences in image segmentations. We have used REPLICA to standardize these datasets and reconcile their segmentation results, enabling scientific analyses of them together.

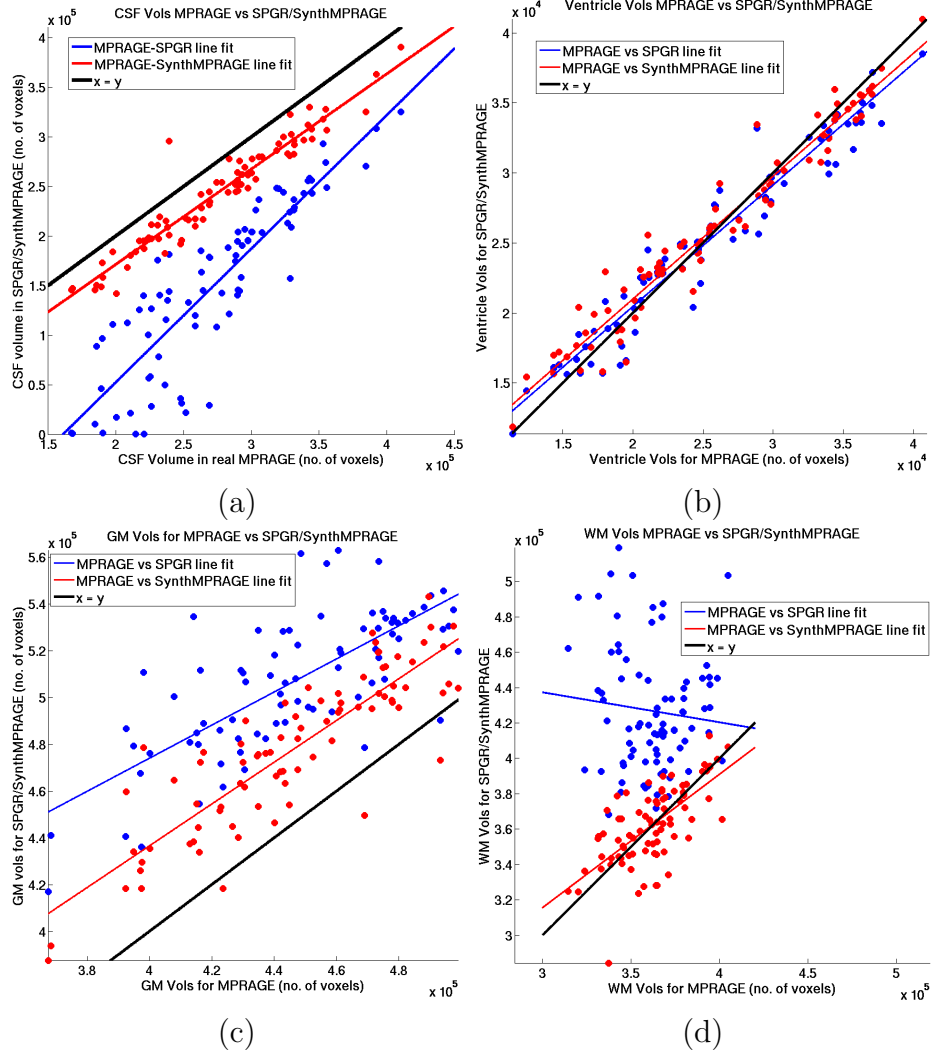


Figure 2.13: (a) CSF AtlasEM volume scatter plot for SPGR vs MPRAGE and synthetic MPRAGE vs MPRAGE, (b) Ventricles AtlasEM volumes, (c) GM AtlasEM volumes, and (d) WM AtlasEM volumes. The blue scatter plots are of volumes observed in SPGR vs those in MPRAGE. The red scatter plots are of volumes observed in synthetic MPRAGE vs those in MPRAGE. The black line indicates the identity transform $x = y$.

2.3.5 Super-resolution using REPLICA

During a clinical or research scan, not all pulse sequences are acquired at the same resolution. In order to perform multimodal analysis, it is necessary to align the data

CHAPTER 2. REPLICA

from different modalities in a common reference frame and at a common resolution. Super-resolution algorithms enhance the resolution of low resolution data. There exists a large body of research on super-resolution techniques in general and also specific to MRI. These include pre-processing [65, 66] and post-processing [1, 24, 31, 67–70] techniques, a subset of which is example-based super-resolution. Initially proposed for MR images by Rousseau [24], example-based methods leverage the high resolution information extracted from a high resolution (HR) image—an MPRAGE, for example—in conjunction with a low resolution (LR) input image—corresponding T_2 w image—to generate a super-resolution (SR) version of the LR image. A number of approaches have followed up on this idea by using self-similarity [1] and a generative model [32]. We have applied REPLICA to this problem previously [48]. Here we perform additional experiments on larger datasets and show segmentation improvements due to super-resolution.

We used REPLICA to synthesize super-resolution images from a joint input of LR and HR images. In the first experiment, we validated our approach on real data by generating super-resolution images of low resolution T_2 w images and comparing them with known high resolution T_2 w images.

2.3.5.1 T_2 w Super-resolution

For this experiment, the atlas set consisted of:

a_1 : HR MPRAGE image (3 T, TR = 6.7 ms, TE = 3.1 ms, TI = 842 ms, $1.1 \times$

CHAPTER 2. REPLICA

$1.1 \times 1.5 \text{ mm}^3$ voxel size),

a_2 : LR T_2 w image from the second echo of a DSE (3 T, TR = 6653 ms, TE₁ = 30 ms, TE₂ = 80 ms, $1.1 \times 1.1 \times 1.5 \text{ mm}^3$ voxel size, downsampled to $1.1 \times 1.1 \times 4 \text{ mm}^3$ and interpolated (trilinear) back to HR) .

a_3 : HR T_2 w image from the second echo of a DSE (3 T, TR = 6653 ms, TE₁ = 30 ms, TE₂ = 80 ms, $1.1 \times 1.1 \times 1.5 \text{ mm}^3$ voxel size),

The subject set includes:

b_1 : HR MPRAGE image (3 T, TR = 6.7 ms, TE = 3.1 ms, TI = 842 ms, $1.1 \times 1.1 \times 1.5 \text{ mm}^3$ voxel size),

b_2 : LR T_2 w image from the second echo of a DSE (3 T, TR = 6653 ms, TE₁ = 30 ms, TE₂ = 80 ms, $1.1 \times 1.1 \times 1.5 \text{ mm}^3$ voxel size, downsampled to $1.1 \times 1.1 \times 4 \text{ mm}^3$ and interpolated (trilinear) back to HR) .

Table 2.5: Mean PSNR (dB) and standard deviation for each of three methods across 40 scans. The three methods are nearest neighbor (NN) and trilinear (TL) interpolation, b-spline (BSP) and our method—REPLICA. The state-of-the-art self similarity based super-resolution (SSS) [1] failed to run on any of the real data. We made several attempts to transform the real data into a usable form for SSS, which did not help.) * denotes statistically significantly better than either of the other two methods (α level of 0.01) using a right-tailed test.

Mean PSNR (Std. Dev.)			
NN	TL	BSP	REPLICA
21.26 (0.73)	22.81 (0.72)	22.63 (0.70)	26.51 (0.85)*

The feature set includes 3D patches concatenated from the HR MPRAGE and LR T_2 w images, and the dependent variable is the intensity of the corresponding center

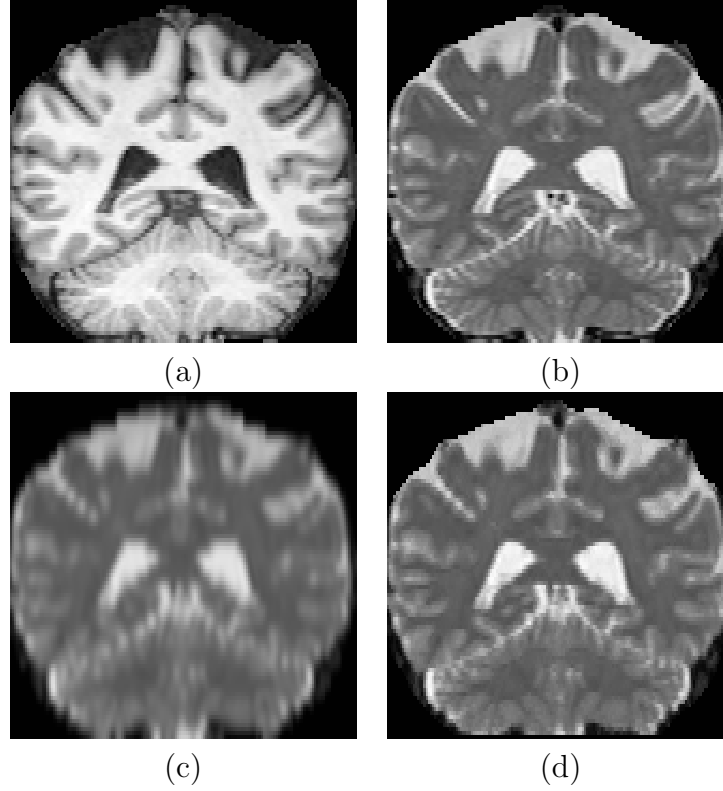


Figure 2.14: Shown here are (a) HR T_1 w image with 1.5 mm slice thickness, (b) HR T_2 w image with 1.5 mm slice thickness, (c) LR T_2 w image with 4 mm slice thickness, upsampled using cubic b-splines, and (d) SR T_2 w image using REPLICA on LR T_2 w of slice thickness 4 mm.

voxel of the HR T_2 w image. The training is done on the atlas images and the learned regression is applied to the features extracted from the subject images to produce a super-resolution T_2 w image. We used 20 subjects of the MMRR dataset [50], each with two imaging sessions, making it 40 scans. One subject was used for training. We synthesized the super-resolution T_2 w images and compared them to the existing original HR T_2 w using PSNR as a metric. Table 2.5 shows the PSNR values, compared against nearest neighbor, trilinear, and cubic b-spline interpolation methods. Figure 2.14 shows the original input HR T_1 w MPRAGE, the ground truth HR T_2 w image, the

CHAPTER 2. REPLICA

result of upsampling using b-spline interpolation, and the REPLICA result. The improvement in resolution is visible when compared against the upsampled image. We tried to compare against the self-similarity method described in [1] but were unable to run it despite our best efforts. REPLICA was consistently superior in a previous comparison we published in [48] on the Brainweb phantom data.

2.3.5.2 FLAIR Super-resolution

In this experiment we applied REPLICA to generate a super-resolution (SR) FLAIR, which along with a HR MPAGE, should provide a better tissue segmentation than a LR FLAIR and HR MPAGE. We used 13 subjects from our in-house MS dataset. The atlas set for this experiment was:

a_1 : HR MPAGE image (3 T, TR = 10.3 ms, TE = 6 ms, $0.82 \times 0.82 \times 1.17$ mm³, resampled to $1 \times 1 \times 1$ mm³ voxel size),

a_2 : LR FLAIR (3 T, TI = 11000 ms, TE = 68 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size, resampled to $1 \times 1 \times 4$ mm³) interpolated back to HR (trilinear),

a_3 : HR FLAIR (3 T, TI = 11000 ms, TE = 68 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size, resampled to $1 \times 1 \times 1$ mm³)

The subject set was:

b_1 : HR MPAGE image (3 T, TR = 10.3 ms, TE = 6 ms, $0.82 \times 0.82 \times 1.17$ mm³, resampled to $1 \times 1 \times 1$ mm³ voxel size),

CHAPTER 2. REPLICA

b_2 : LR FLAIR (3 T, TI = 11000 ms, TE = 68 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size, resampled to $1 \times 1 \times 4$ mm³, interpolated back to HR (trilinear))

We ran the LesionTOADS [55] segmentation algorithm on three cases for each subject: (a) HR MPRAGE + LR FLAIR; (b) HR MPRAGE + SR FLAIR; (c) HR MPRAGE + HR FLAIR. The last case acts as the ground truth for how the segmentation algorithm should perform on the best case data. We aim to show that the tissue segmentation using SR FLAIR is closer to that achieved using HR FLAIR, than using LR FLAIR. Figures 2.15 (d), (k) show the super-resolution results, the LR FLAIR images are shown in Figs. 2.15 (b), (i), and the HR FLAIR images in Figs. 2.15 (c), (j). Visually it is apparent that the lesion boundaries are more visible on the SR FLAIR than on the LR FLAIR. The corresponding LesionTOADS segmentations are shown in Figs. 2.15 (e)–(g) and Figs. 2.15 (l)–(n). The lesion boundaries as well as the cortex is overestimated when a LR FLAIR is used. These errors are not visible on the segmentation obtained using SR FLAIR. The corresponding HR MPRAGE slices are shown in Figs. 2.15 (a) and (h), respectively.

We attempted to compare against the state-of-the-art self similarity based super-resolution (SSS) [1] failed to run on any of the real data. We made several attempts to transform the real data into a usable form for SSS, but those did not help. We looked at lesion volumes as produced by LesionTOADS for all three cases for 13 subjects. Figure 2.16 shows the lesion volumes produced for SR FLAIR are more similar to the ones produced by HR FLAIR.

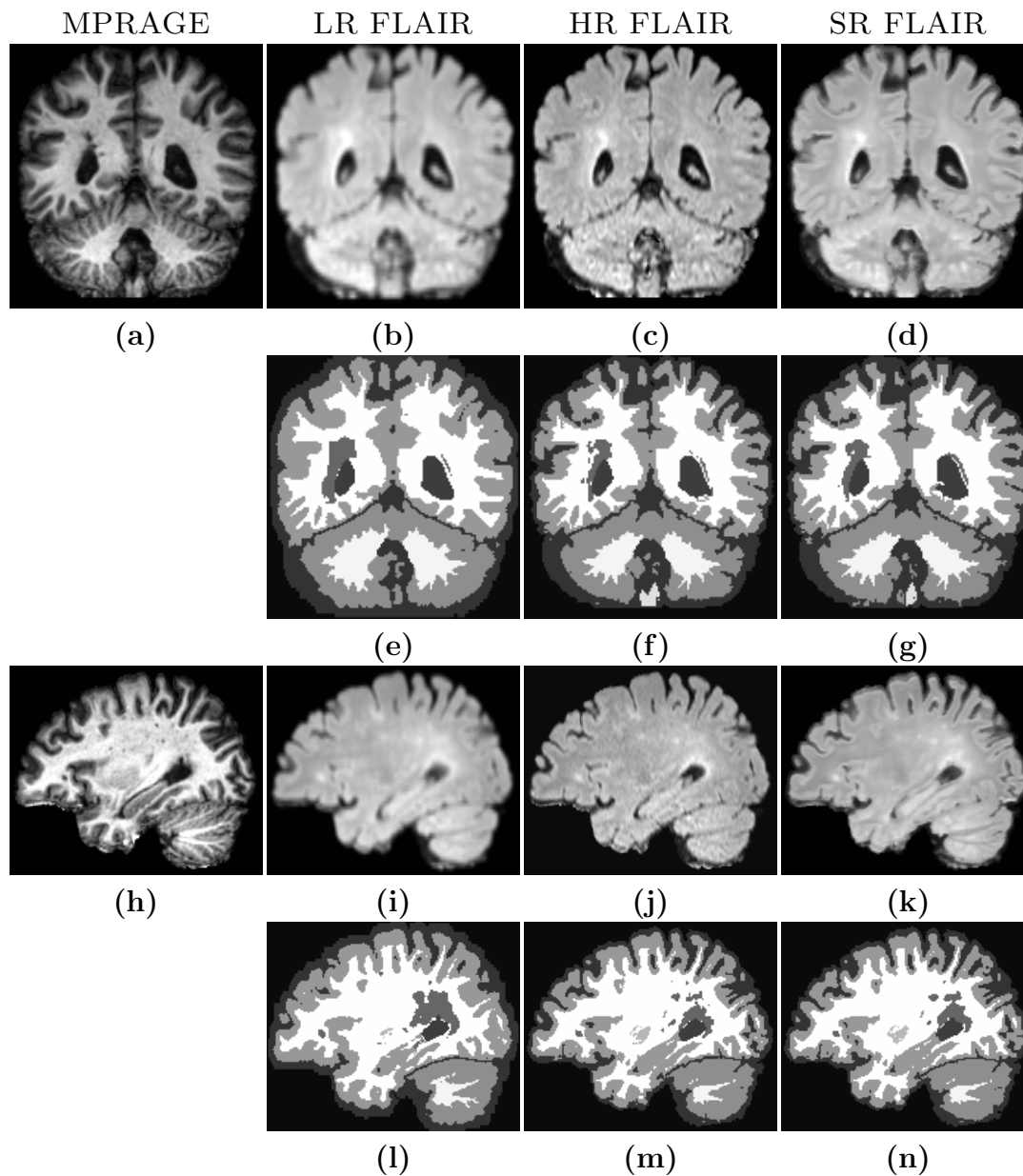


Figure 2.15: Coronal slices of: (a) HR MPRAGE, (b) LR FLAIR, (c) HR FLAIR, (d) SR FLAIR, followed by coronal slices of LesionTOADS lesion segmentations of (e) HR MPRAGE + LR FLAIR, (f) HR MPRAGE + HR FLAIR, and (g) HR MPRAGE + SR FLAIR. Sagittal slices of: (h) HR MPRAGE, (i) LR FLAIR, (j) HR FLAIR, (k) SR FLAIR, followed by sagittal slices of LesionTOADS lesion segmentations of (l) HR MPRAGE + LR FLAIR, (m) HR MPRAGE + HR FLAIR, and (n) HR MPRAGE + SR FLAIR.

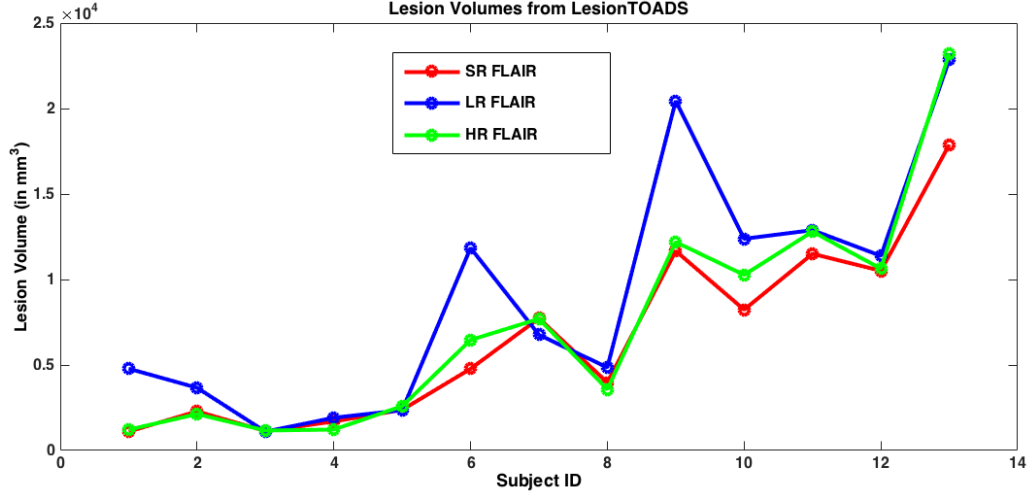


Figure 2.16: The x -axis axis denotes subject IDs. The y -axis denotes the lesion volumes provided by LesionTOADS in mm^3 . The green plot shows lesion volumes acquired by LesionTOADS on HR FLAIR+HR MPRAGE. The blue plot shows LesionTOADS LR FLAIR lesion volumes, and the red plot shows LesionTOADS SR FLAIR lesion volumes. Notice that the green plot is closer to the red plot than the blue plot for most of the subjects.

Thus, using REPLICA we can improve the resolution of all the acquired images to the one image that is acquired at the highest resolution. Improving resolution enables improved segmentation and further image processing.

2.4 Summary and Discussion

We have described a new image synthesis algorithm called REPLICA. We have shown that REPLICA demonstrates significant improvement in image quality over other state-of-the-art synthesis algorithms (Section 2.3.1). We have also described applications, where image synthesis in general and REPLICA in particular, can be beneficial as a preprocessing step for subsequent image processing steps.

CHAPTER 2. REPLICA

The T_2w synthesis for full-head images described in Section 2.3.2 also highlights the capability of REPLICA to handle complex image synthesis scenarios, with limited intensity information at its disposal. To our knowledge, REPLICA is the first intensity transformation-based synthesis approach to do this.

Synthesis of FLAIR images is a key application (Section 2.3.3) that has not been demonstrated before; this process can be useful if FLAIR images were not acquired, if the acquired FLAIR images are corrupted, or if higher-resolution FLAIR images are desired.

We have also demonstrated the use of REPLICA for intensity standardization between two T_1w modalities such as SPGR and MPRAGE in Section 2.3.4. This example mimics the scenario of a typical large multi-center, multi-scanner MRI study, where images obtained from two different scanners can lead to differences in image segmentations. REPLICA can be used to standardize these datasets and bridge the gap between their respective segmentation results, enabling scientific results with higher statistical power.

In Section 2.3.5 we used REPLICA to perform example-based super-resolution. The resulting super-resolution images provided a better segmentation than available low resolution ones. An MRI scanning session almost always includes scans which are of heterogenous quality, especially in terms of voxel resolution. Using this method, we can improve the resolution of all the acquired images to the one image that is acquired at the highest resolution, thus immediately improving the following image analysis.

CHAPTER 2. REPLICA

This can also be extended to different subjects by super-resolving available images to a common resolution to provide more consistent image analysis across subjects.

REPLICA is computationally fast. Training an ensemble can take up to 20 minutes, but needs to be done only once. With (easy) parallelization over eight cores, synthesis of a $256 \times 256 \times 173$ image takes less than a minute on a 3 GHz computer. Given the typical times of neuroimaging pipelines (usually many hours), this makes the use of REPLICA as a preprocessing step quite feasible.

REPLICA has some limitations that should be addressed in the future. Since the predicted value of a random forest is the average of the results of all trees (each of which is an average of at least five values), the synthetic images often appear to have lower noise and are slightly smoother than their real counterparts. Lower noise may benefit algorithms but reduction in resolution is not typically beneficial. In a previous publication we have demonstrated super-resolution as an application of REPLICA [48], so there may be a relatively straightforward way to enhance resolution in the exact amount needed to offset the inherent loss of resolution due to averaging. Also, we used the mode instead of the average in certain applications, but this is also an empirical strategy and not guaranteed to address the problem. Another limitation concerns the features that are used to train REPLICA. Although our features are sensible and effective for the applications we have explored, they are nevertheless empirically selected and they may not be optimal for these tasks or new scenarios that may be encountered in the future.

CHAPTER 2. REPLICA

The development of REPLICA has been driven by empirical choices. The nonlinear regression using random forests, predicts a synthetic image voxel-by-voxel, each of them independent of the other. However, it is known that voxels in medical images (and natural images) have neighborhood dependencies. Two neighboring voxels are more likely to have similar intensities. The regular anatomy can lead to many more regional dependencies that are not modeled by REPLICA. REPLICA does not provide a probabilistic interpretation of its result. It is not apparent if the resulting synthetic image is “optimal” by any well-defined criterion. We describe our attempt to create such a probabilistic framework, SynthCRAFT, which is a discriminative model of image synthesis, in Chapter 4.

REPLICA operates on the fundamental assumption that the atlas images a_i have the same acquisition parameters as the input subject images b_i , or they can be intensity standardized using a simple transform, for example linear or piecewise linear. However, this is not always the case. The atlas data is a very specialized collection of high quality images from a variety of pulse sequences. More often than not, the subject data is not going to be acquired from the same imaging protocol or even the same scanners. If the atlas and subject images are not intensity standardized well, the training data is not going to be representative of the test data. No regression algorithm is robust enough to handle this scenario gracefully and all supervised image synthesis algorithms, including REPLICA will fail to produce the desired results. To solve this problem, we have developed Ψ -CLONE, a framework to standardize the subject images to the

CHAPTER 2. REPLICA

atlas images by estimating the pulse sequence parameters used to acquire the subject images and then perform synthesis via regression. In this framework, our atlas image set also contains maps of intrinsic NMR parameters, in addition to the MR images. We describe Ψ -CLONE in depth in the next chapter.

In conclusion, the REPLICA image synthesis method was described and shown to be effective in medical image processing tasks. REPLICA was shown to be beneficial when images are missing or corrupted for subsequent processing steps. It is a simple, fast, and effective approach that can be readily employed as a preprocessing step in many neuroimage processing pipelines.

Appendix

2.A Parameter Selection for REPLICA

The REPLICA framework has a number of free parameters that can be tuned to improve the resulting synthesis. In this section we discuss and justify our choice of the following parameters: (a) number of trees, (b) number of samples in a leaf node, (c) number of training samples (d) size of local 3D patch, (e) use of more than one individual in the atlas, (f) use of alternate atlas images.

For this task, we used T_1 -w MPRAGE images from the publicly available MMRR dataset [50]. The MMRR data consists of 21 subjects, each with two imaging sessions acquired within an hour of each other. The task involved synthesizing T_2 w images of the DSE sequence from the corresponding T_1 -w MPRAGE images. We used skull-stripped MPRAGE images from both sessions of 20 subjects and synthesized their corresponding T_2 w images using REPLICA. We used one subject for training. The atlas and subject images have the following specifications:

a_1 : MPRAGE image (3 T, TR = 6.7 ms, TE = 3.1 ms, TI = 842 ms, $1.0 \times 1.0 \times$

CHAPTER 2. REPLICA

1.2 mm³ voxel size),

a_2 : T_2 w image from the second echo of a DSE (3 T, TR = 6653 ms, TE₁ = 30 ms, TE₂ = 80 ms, 0.82 × 0.82 × 1.5 mm³ voxel size).

b_1 : MPRAGE image (3 T, TR = 6.7 ms, TE = 3.1 ms, TI = 842 ms, 1.0 × 1.0 × 1.2 mm³ voxel size)

For evaluation of synthesis quality, we used PSNR (peak signal to noise ratio), which is a mean squared error-based metric, UQI (universal quality index) [53], and SSIM (structural similarity) [54]. UQI and SSIM are more sensitive to perceptual differences in image structure than PSNR since they take into account properties of the human visual system. A value of 1 for both UQI and SSIM indicates that images are equal to each other; otherwise their values lie between 0 and 1.

2.A.1 Number of Trees

In this experiment, we varied the number of trees used in the random forest. The feature vector for this experiment was $\mathbf{f}^l(\mathbf{x}) = [\mathbf{p}^1(\mathbf{x}), \mathbf{v}(\mathbf{x})]$, where $\mathbf{p}^1(\mathbf{x})$ is the local $3 \times 3 \times 3$ patch at the highest resolution and $\mathbf{v}(\mathbf{x})$ is the high resolution context descriptor, calculated at voxel \mathbf{x} . Each tree in the random forest was grown until there were no fewer than five samples in a leaf node, i.e., $t_c = 5$, $t_p = 2t_c$, and $\epsilon = 10^{-6}$. We successively changed the number of trees from the set $\{1, 10, 30, 60, 150\}$. The resulting synthetic images were compared against the ground truth acquired images

CHAPTER 2. REPLICA

using PSNR, UQI, and SSIM, as shown in Fig. 2.A.1. Each box plot is generated from metrics calculated over $20 \times 2 = 40$ synthetic images. There is a statistically significant increase ($p < 0.05$ two-tailed t-test) in these metrics between the number of trees = 1 to the number of trees = 10. However, although the scores improve with an increasing number of trees, after 30 trees the values are not statistically different. To keep a suitable balance between performance and computation time, we used 60 trees in all subsequent experiments.

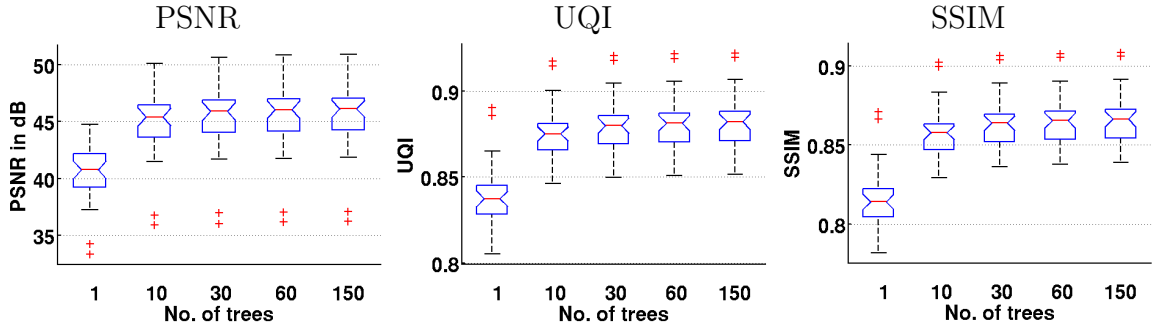


Figure 2.A.1: Box plots for PSNR, UQI, and SSIM values obtained by comparing the synthetic REPLICA-generated images with the ground truth.

2.A.2 Number of samples in leaf nodes

The number of samples in each leaf node can have a substantial effect on the performance of a random forest. Here, we experiment with t_c , the minimum number of samples in leaf nodes. Keeping other factors that affect tree depth as $t_p = 2t_c$ and $\epsilon = 10^{-6}$, if t_c is made larger then the tree stops growing earlier and the leaf nodes have more values to average as the output of the tree. To assess the quantitative

CHAPTER 2. REPLICA

impact of this parameter, we varied t_c in the set $\{5, 20, 50, 200, 1000\}$ and used the feature vector $\mathbf{f}^l(\mathbf{x}) = [\mathbf{p}^l(\mathbf{x}), \mathbf{v}(\mathbf{x})]$, where $\mathbf{p}^l(\mathbf{x})$ is the local $3 \times 3 \times 3$ patch at the highest resolution and $\mathbf{v}(\mathbf{x})$ is the high resolution context descriptor. We fixed the number of trees to 60.

From the plots in Fig. 2.A.2, we can see that the performance metrics decrease slightly as t_c increases, but the change is not statistically significant. Since it does not affect computation time much, we set $t_c = 5$ in all remaining experiments.

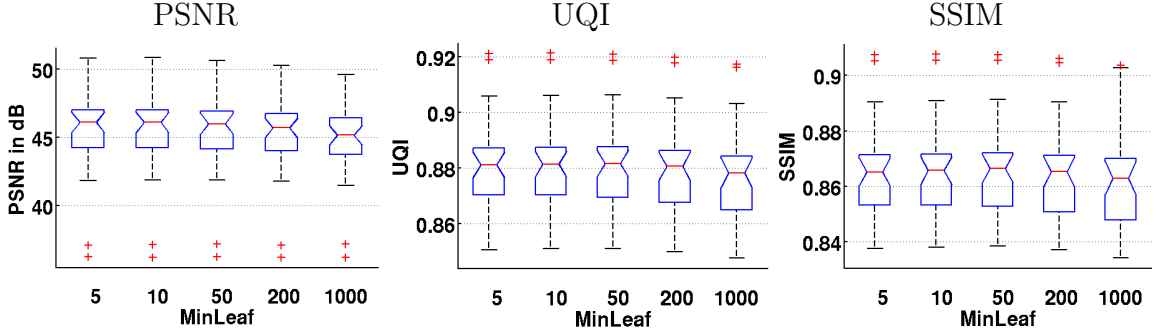


Figure 2.A.2: Synthesis performance as a function of t_c .

2.A.3 Number of training samples

The number of samples used for training REPLICA can affect the quality of synthesis. Ideally, synthesis quality should improve with an increasing number of training samples. However due to computational constraints such as limited available memory and a finite amount of training time, we need to choose a value that balances the computational burden and synthesis quality. The training data is generated by sampling foreground voxels of the atlas image set and calculating the features at

CHAPTER 2. REPLICA

these voxels. A typical atlas image has $\sim 10^7$ foreground voxels. We cannot use all of them as memory is limited. We increase the number of training samples in the set $\{10^3, 10^4, 10^5, 2 \times 10^5, 5 \times 10^5, 10^6\}$. The feature vector for this experiment was $\mathbf{f}^l(\mathbf{x}) = [\mathbf{p}^1(\mathbf{x}), \mathbf{v}(\mathbf{x})]$, where $\mathbf{p}^1(\mathbf{x})$ is the local $3 \times 3 \times 3$ patch at the highest resolution, and $\mathbf{v}(\mathbf{x})$ is the high resolution context descriptor. The number of trees was fixed to 60 and $t_c = 5$, $t_p = 2t_c$, and $\epsilon = 10^{-6}$. We calculated PSNR, UQI and SSIM from comparison with the ground truth images, the boxplots of which are shown in Fig 2.A.3. All the performance metrics increase as number of training samples are increased from 10^3 to 10^5 . The change is statistically significant for UQI and SSIM, but not PSNR. The metrics plateau after 10^5 samples. The training times and memory requirements were obtained from a 16 core, 3.42 GHz machine with 193 GB memory. These are shown in Table 2.A.1. Based on the synthesis performance and the computational resources available in a realistic scenario, we chose number of samples to be 10^5 in our experiments.

Table 2.A.1: Time and memory requirements of training REPLICA as function of number of training samples

No. of samples	Time (seconds)	Memory (GB)
10^3	0.9	2.6
10^4	1.2	2.8
10^5	13.2	3.9
2×10^5	27.3	6.7
5×10^5	79.0	11.3
10^6	194.2	18.7

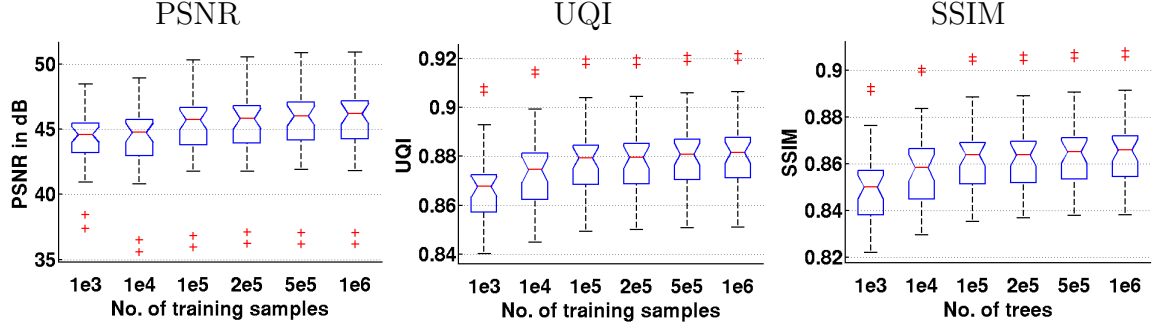


Figure 2.A.3: As number of training samples increases, all metrics improve but plateau later. UQI and SSIM improve statistically significantly from 1e3 to 1e5, but not PSNR.

2.A.4 Additional Atlases

In this experiment we keep the number of training samples the same, but pick these from a different number of atlases (individuals), from 1–4. We synthesize T_2w images from MPRAGE images as in previous experiments. The feature vector for this experiment was $\mathbf{f}^1(\mathbf{x}) = [\mathbf{p}^1(\mathbf{x}), \mathbf{v}(\mathbf{x})]$, where $\mathbf{p}^1(\mathbf{x})$ is the local $3 \times 3 \times 3$ patch at the highest resolution, and $\mathbf{v}(\mathbf{x})$ is the high resolution context descriptor. The number of trees was fixed to 60 and $t_c = 5$, $t_p = 2t_c$, and $\epsilon = 10^{-6}$. Since we used data from four subjects in this experiment, the box plots were generated using only $17 \times 2 = 34$ synthetic images.

The image quality metrics box plots in Fig. 2.A.4 show that the PSNR increases significantly (2-sample t-test, $p < 0.05$) when the number of atlases increases from 1 to 2, but not after. UQI and SSIM do not reflect a similar improvement. Thus, for the task of synthesizing normal anatomy images, use of one individual as the atlas is sufficient.

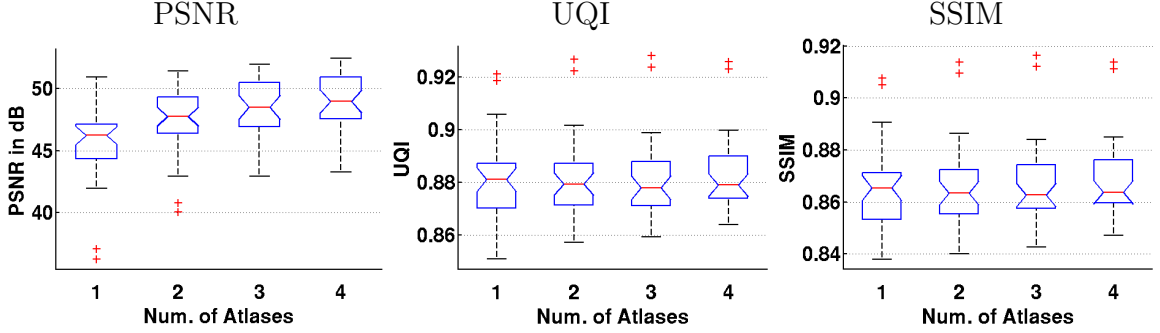


Figure 2.A.4: Box plots for all the metrics when an increasing number of atlases are used. PSNR for number of atlases=1 is significantly different than the rest of the atlases ($p < 0.05$) using the two-sample t-test. However other metrics and further increase in number of atlases do not show any significant differences.

2.A.5 Using Different Atlas Sets

Although one atlas is sufficient for synthesis of normal anatomy, there may be differences in which subject is chosen as the atlas. In this experiment, we used the same four individuals as in the previous section to evaluate the impact of this difference. The feature vector for this experiment was $\mathbf{f}^l(\mathbf{x}) = [\mathbf{p}^1(\mathbf{x}), \mathbf{v}(\mathbf{x})]$, where $\mathbf{p}^1(\mathbf{x})$ is the local $3 \times 3 \times 3$ patch at the highest resolution, and $\mathbf{v}(\mathbf{x})$ is the high resolution context descriptor. The number of trees was fixed to 60 and $t_c = 5$, $t_p = 2t_c$, and $\epsilon = 10^{-6}$. Since we used have used data from four subjects as atlases in this experiment, the box plots were generated using only $17 \times 2 = 34$ synthetic images.

The PSNR, UQI, and SSIM plots shown in Fig. 2.A.5 show that the changes introduced by different atlas images are minimal. The x-axis in these plots is the atlas index used in the training data and ranges from 1–4. We note that in Fig. 2.A.5 the PSNR observed for Atlas 1 is statistically significantly different from the other

CHAPTER 2. REPLICA

atlases, but the same does not hold true for the other metrics or atlases. Thus, for synthesizing normal anatomy images, the choice of the atlas does not systematically affect the synthesis results.

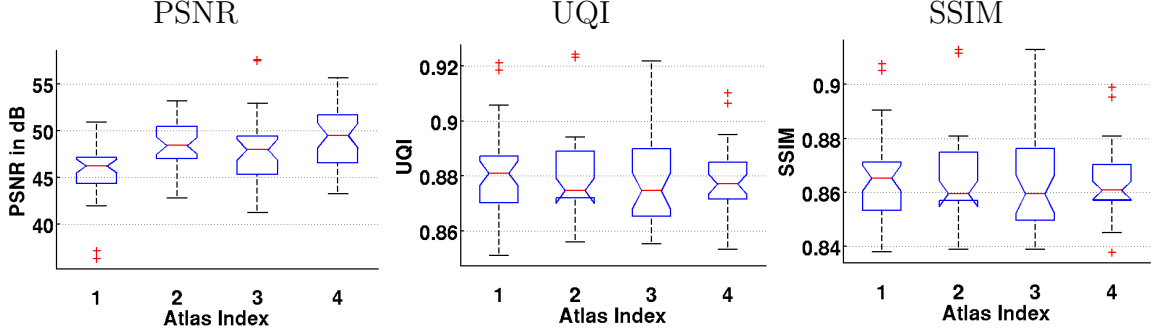


Figure 2.A.5: Box plots for all the metrics when four different atlases are used. PSNR for Atlas 1 is significantly different than the rest of the atlases ($p < 0.05$) using the two-sample t-test. However other metrics and other atlases show no significant differences.

2.A.6 Size of Local Patch

Next, we looked at selecting the best local features for synthesis. We considered the same task of synthesizing skull-stripped T_2w images from MPRAGE images of 20 subjects of the MMRR dataset with the number of trees fixed to 60 and $t_c = 5$, $t_p = 2t_c$, and $\epsilon = 10^{-6}$. The feature vector for this experiment was $\mathbf{f}^l(\mathbf{x}) = [\mathbf{p}^1(\mathbf{x}), \mathbf{v}(\mathbf{x})]$, where $\mathbf{p}^1(\mathbf{x})$ is the local $p \times p \times p$ patch at the highest resolution, and $\mathbf{v}(\mathbf{x})$ is the high resolution context descriptor. We varied the size of the local cubic $p \times p \times p$ patch for $p \in \{1, 3, 5, 7, 9\}$.

The resulting PSNR, UQI, and SSIM plots are shown in Fig. 2.A.6. All the metrics

CHAPTER 2. REPLICA

show a statistically significant increase from patch size of $1 \times 1 \times 1$ to $3 \times 3 \times 3$, and again from patch size of $3 \times 3 \times 3$ to $5 \times 5 \times 5$. Further increase in the patch size does not result in a further statistically significant increase in the metrics.

Despite the improvements in our performance metrics with increasing patch size, in the experiments below we used a $3 \times 3 \times 3$ patch. Our rationale is carefully reasoned as follows. First, we observed that improvements in our performance metrics from patch sizes $3 \times 3 \times 3$ to $5 \times 5 \times 5$ appear to be primarily due to noise reduction in large, homogeneous white matter regions. Such noise reduction yields synthetic images that are much smoother in appearance than real images. There is a concern that such images will not perform the same as real images in subsequence image processing.

A second reason for sticking with $3 \times 3 \times 3$ patches is due to the increased computer memory burden (almost a factor of 3) of $5 \times 5 \times 5$ patches. A third reason is that with the increase in feature dimensionality comes a requirement to increase the number of training samples and this puts a further memory burden on the software as well as a computation time increase. In particular, the training time goes from about 5 minutes to 40 minutes when going from $3 \times 3 \times 3$ to $5 \times 5 \times 5$ patches. Taking into consideration the concerns over unnatural noise reduction and the computational costs, we chose to use a $3 \times 3 \times 3$ -sized patch in all of our experiments.

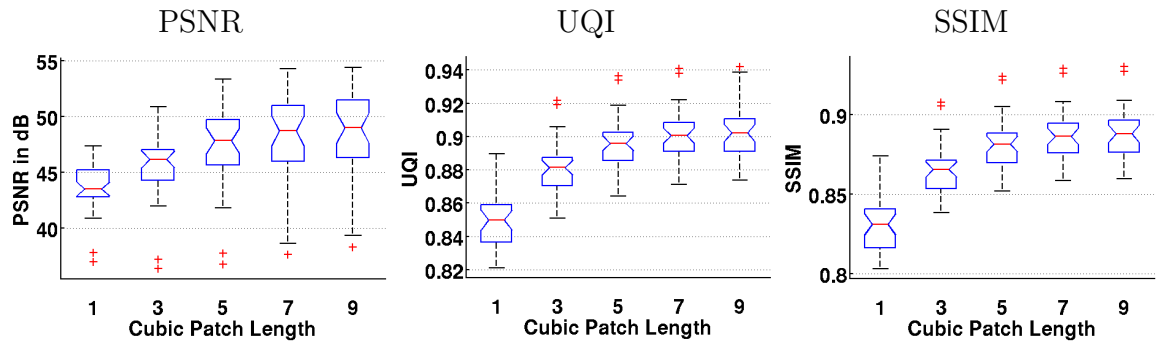


Figure 2.A.6: Increase in patch size from $1 \times 1 \times 1$ to $3 \times 3 \times 3$ and further to $5 \times 5 \times 5$ are statistically significant ($p < 0.05$, two-tailed t-test).

Chapter 3

Ψ -CLONE: Pulse Sequence

Information Driven MR Image

Synthesis

3.1 Introduction

All state-of-the-art MR image synthesis algorithms have so far ignored a vital aspect of MR image processing—the MR image formation process. As described in Chapter 1, MR imaging physics is responsible for image formation. Voxel intensity in MRI is primarily dependent on (1) intrinsic NMR parameters such as P_D , T_2 , T_1 , and, (2) pulse sequence parameters such as scanner gain (G_0), repetition time (TR), echo times (TE), etc. It is important to take into account how these quantities affect

CHAPTER 3. Ψ -CLONE

MR image formation in order to synthesize an MR image correctly. We propose an approach that attempts to achieve this. Like previous approaches, our atlas image set also consists of images from a variety of pulse sequences and acts as the training data. However, in addition, our atlas also consists of the P_D , T_1 , and T_2 maps of the corresponding images. We assume that the MRI signal is primarily dependent on these three intrinsic NMR parameters. This is an approximation since we know that it is also dependent on values like T_2^* and can be affected by boundary artifacts. In this work, we assume that the signal, i.e. the voxel intensity is directly dependent only on P_D , T_1 , and T_2 . Given a subject image (and the pulse sequence used to acquire it) we have developed a procedure to estimate the pulse sequence parameters using information derived solely from the image intensities. We apply this pulse sequence with the estimated parameters to quantitative P_D , T_1 , and T_2 images in our atlas. Thus, a new atlas image with the same imaging characteristics as the subject image, is generated. This step essentially performs intensity standardization between the atlas and the subject image. It is how the atlas image would have looked had we acquired it with the subject pulse sequence. This step is crucial because it means that we need not have an existing atlas image that is of the same contrast or pulse sequence of the subject image. We can always generate a new atlas image from the intrinsic NMR parameters. We then learn a nonlinear regression [30] between this standardized atlas image and the desired, target atlas contrast. This learned regression is then applied to the subject image directly, thereby synthesizing a subject image with the desired

CHAPTER 3. Ψ -CLONE

contrast.

The core idea of estimating tissue parameters was previously used by Fischl et al. [71]. However, that approach required the acquisition of very specific pulse sequences—a limitation that our approach does not share. We refer to our method as Pulse Sequence Information-based Contrast Learning On Neighborhood Ensembles (PSI-CLONE)—which we stylize as Ψ -CLONE. We describe our four-step algorithm to perform image synthesis in Section 3.2. We describe results of validation experiments on the Brainweb phantom data [72] in Section 3.3 and on real data in Section 3.4; these experiments include scanner intensity standardization and synthesis of T_2w images. In Section 3.5, we present additional synthesis applications: super-resolution and FLAIR synthesis. In Section 3.6, there is a discussion about the potential impact of this work and concluding remarks.

3.2 Method

Let $\mathcal{B} = \{b_1, b_2, \dots, b_m\}$ be the given subject image set, imaged with pulse sequences $\Gamma_1, \dots, \Gamma_m$. This set can contain images from different pulse sequences such as MPRAGE, SPGR, and DSE and others. Let $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ be the atlas collection, with images of contrasts $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_n$, generated by pulse sequences Ψ_1, \dots, Ψ_n , respectively.

The pulse sequence sets $\{\Gamma_1, \dots, \Gamma_m\}$ and $\{\Psi_1, \dots, \Psi_n\}$ need not intersect, which

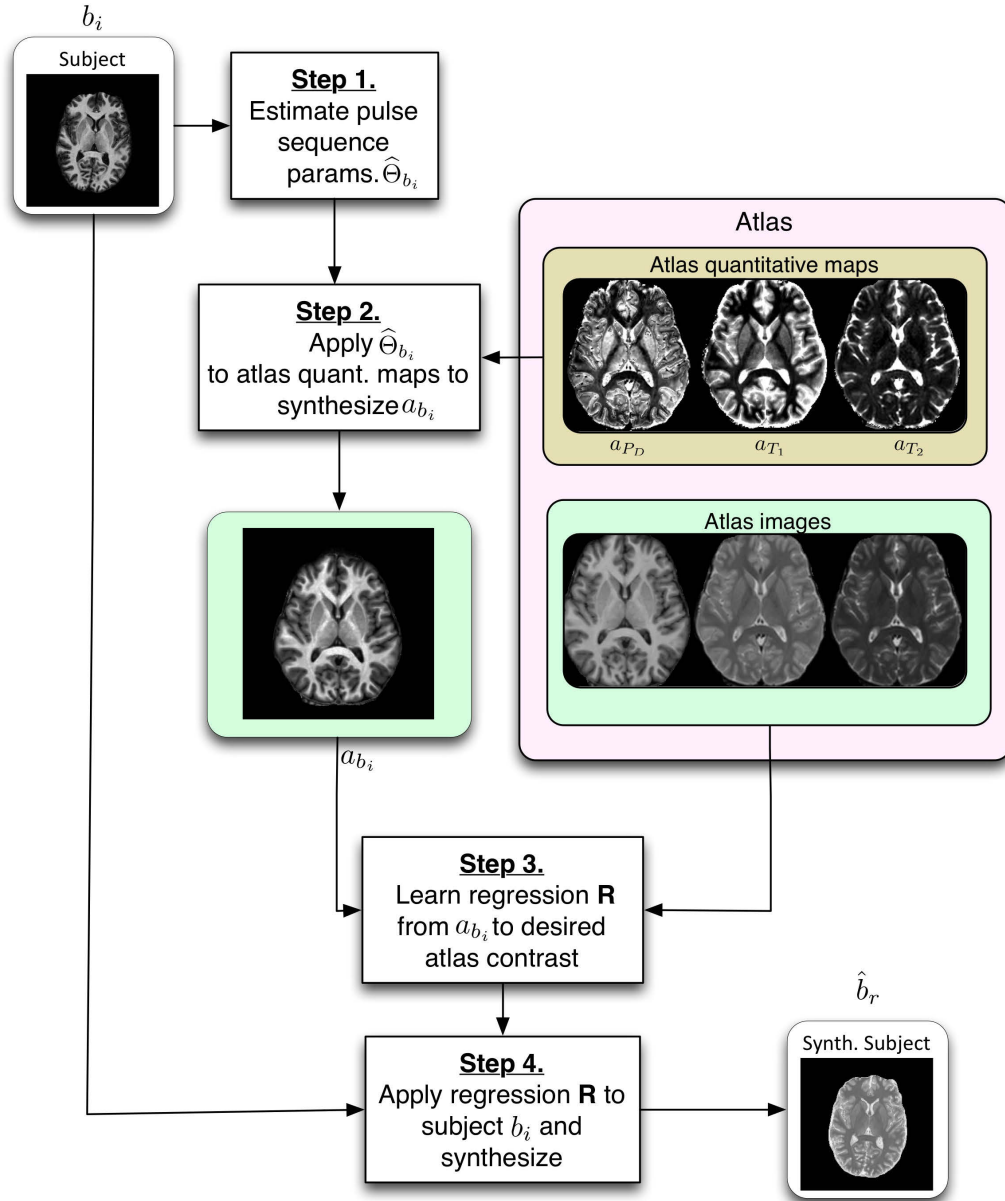


Figure 3.1: A flow chart of the Ψ -CLONE algorithm.

CHAPTER 3. Ψ -CLONE

represents an important distinction between Ψ -CLONE and all other atlas-based image synthesis methods. The atlas also contains quantitative P_D , T_1 , and T_2 maps, denoted a_{P_D} , a_{T_1} , and, a_{T_2} . Our goal is to synthesize the subject image \hat{b}_r , $r \in \{1 \dots, n\}$, which is how the subject brain would look had it been imaged with pulse sequence Ψ_r used to acquire the atlas image a_r . The steps of our algorithm are as follows:

1. Estimate the pulse sequence parameters used to acquire b_i , $i \in \{1, \dots, m\}$
2. With this estimate, generate a_{b_i} the atlas image with the same contrast, \mathcal{C}_i , as b_i
3. From the expanded atlas collection $\mathcal{A} \cup \{a_{b_i}\}$ we learn the nonlinear intensity transformation between contrast \mathcal{C}_i and the target contrast image a_r (of contrast \mathcal{C}_r), using patch-based random forest regression
4. The intensity transformation is then applied to b_i , generating the subject image \hat{b}_r , of the desired contrast.

These steps are outlined graphically in Fig. 3.1 and detailed in the following sections.

3.2.1 Estimation of Subject Pulse Sequence Parameters

The intensity observed at voxel location \mathbf{x} in b_i is assumed to be a result of the underlying tissue parameters—proton density P_D , longitudinal relaxation time T_1 , and transverse relaxation time T_2 —denoted by $\boldsymbol{\beta}(\mathbf{x}) = [P_D(\mathbf{x}), T_1(\mathbf{x}), T_2(\mathbf{x})]$. The intensity

CHAPTER 3. Ψ -CLONE

is also a result of the pulse sequence used, Γ_i , and its pulse sequence parameters (also referred to as imaging parameters) denoted Θ_{b_i} . Thus we denote the imaging equation as,

$$b_i(\mathbf{x}) = \Gamma_i(\boldsymbol{\beta}(\mathbf{x}); \Theta_{b_i}). \quad (3.1)$$

For the DSE pulse sequence, the equation is

$$\begin{aligned} b_{DSE}(\mathbf{x}) &= \Gamma_{DSE}(\boldsymbol{\beta}(\mathbf{x}); \Theta_{DSE}) \\ &= G_{DSE} P_D(\mathbf{x}) \left(1 - 2e^{-\frac{\text{TR} - \frac{\text{TE}_1 + \text{TE}_2}{2}}{T_1(\mathbf{x})}} + 2e^{-\frac{\text{TR} - \frac{\text{TE}_1}{2}}{T_1(\mathbf{x})}} - e^{-\frac{\text{TR}}{T_1(\mathbf{x})}} \right) e^{-\frac{\text{TE}_2}{T_2(\mathbf{x})}}, \end{aligned} \quad (3.2)$$

where $\Theta_{DSE} = \{\text{TR}, \text{TE}_1, \text{TE}_2, G_{DSE}\}$ consists of repetition time TR, two echo times TE_1 and TE_2 and scanner gain G_{DSE} [6]. For the T_1 w SPGR sequence the imaging equation is

$$\begin{aligned} b_S(\mathbf{x}) &= \Gamma_S(\boldsymbol{\beta}(\mathbf{x}); \Theta_S) \\ &= G_S P_D(\mathbf{x}) \sin \theta \frac{\left(1 - e^{-\frac{\text{TR}}{T_1(\mathbf{x})}} \right)}{1 - \cos \theta e^{-\frac{\text{TR}}{T_1(\mathbf{x})}}} e^{-\frac{\text{TE}}{T_2^*(\mathbf{x})}}, \end{aligned} \quad (3.3)$$

where $\Theta_S = \{\text{TR}, \text{TE}, \theta, G_S\}$ consists of repetition time TR, echo time TE, and flip angle θ [6]. The imaging equation for the MPRAGE sequence can be approximated

CHAPTER 3. Ψ -CLONE

from the mathematical formulation calculated by [5] as

$$\begin{aligned} b_M(\mathbf{x}) &= \Gamma_M(\boldsymbol{\beta}(\mathbf{x}); \Theta_M) \\ &= G_M P_D(\mathbf{x}) \left(1 - \frac{2e^{\frac{-\text{TI}}{T_1(\mathbf{x})}}}{1 + e^{\frac{-(\text{TI}+\text{TD}+\tau)}{T_1(\mathbf{x})}}} \right), \end{aligned} \quad (3.4)$$

where $\Theta_M = \{\text{TI}, \text{TD}, \tau, G_M\}$ consists of inversion time TI, delay time TD, and the slice imaging time τ [5]. We assume that we know one of these parameters from the image header and estimate the rest by fitting the imaging equation to average tissue intensities.

Given an input subject image, b_i , we want to estimate a subset (such as scanner gain, flip angle, echo times) of pulse sequence parameters Θ_{b_i} of Γ_{b_i} . We make certain assumptions about the tissues being imaged barthereby simplifying the system of equations we need to solve. As the human brain is dominated by three primary tissues, cerebrospinal fluid (CSF), gray matter (GM), and white matter (WM), we can use the known average values of $\boldsymbol{\beta}$ to solve for the imaging parameters. The mean values of $\boldsymbol{\beta}$ for CSF, GM, and WM denoted by $\bar{\boldsymbol{\beta}}_C$, $\bar{\boldsymbol{\beta}}_G$, and $\bar{\boldsymbol{\beta}}_W$, respectively have been reported previously for 1.5 T [73] and 3 T MRI [74]. We have tabulated them in Table 3.1.

To identify the three tissue classes, we run a simple three-class fuzzy c-means [75] algorithm on the T_{1w} image (b_S or b_M), and choose voxels with high class memberships (≥ 0.8) to compute the mean intensities of CSF, GM, and WM in b_i as \bar{b}_{iC} , \bar{b}_{iG} , and

CHAPTER 3. Ψ -CLONE

Table 3.1: Average T_1 , T_2 and relative P_D values of CSF, GM and WM at 1.5 T and 3 T

Tissue	Relative P_D		T_2 (ms)		T_1 (ms)	
	1.5 T	3 T	1.5 T	3 T	1.5 T	3 T
White Matter	0.61	0.61	67	79	656	832
Gray Matter	0.69	0.69	77	110	1188	1331
Cerebrospinal Fluid	1.00	1.00	280	1100	4070	4400

\bar{b}_{iW} , respectively for $i = 1, \dots, m$.

We make the assumption that these mean intensities are a result of the mean tissue parameter values. This relationship is written as,

$$\bar{b}_{iC} = \Gamma_i(\bar{\beta}_C; \Theta_{b_i}), \quad \bar{b}_{iG} = \Gamma_i(\bar{\beta}_G; \Theta_{b_i}), \quad \bar{b}_{iW} = \Gamma_i(\bar{\beta}_W; \Theta_{b_i}). \quad (3.5)$$

The only unknown is Θ_{b_i} , which for DSE type pulse sequences is parametrized by four terms. Similarly, MPRAGE and SPGR pulse sequences [5, 6] have four parameters. Thus, we have three equations (Eqn. 3.5) and four unknowns, and we can solve this system of equations using Newton’s method after assuming knowledge of one of the unknowns. For example, in the SPGR pulse sequence, we assume that the repetition time, TR is known from the image header and the unknowns that are often not well-calibrated in an MR scanner—e.g, flip angle and scanner gain—are estimated. For the given subject image set $\mathcal{B} = \{b_1, b_2, \dots, b_m\}$, we can thus estimate $\{\hat{\Theta}_{b_1}, \dots, \hat{\Theta}_{b_m}\}$ for each of the subject images at the end of Step 1.

There are many factors that affect the accuracy of a pulse sequence equation,

CHAPTER 3. Ψ -CLONE

and more unknowns are likely to be actually involved than we may know about. To address this, we have simplified the problem by using theoretical equations or approximations of theoretical equations that describe the basic, functional relationship between the NMR parameters (P_D, T_1, T_2) and the signal intensity. For instance, the signal intensity equation for the MPRAGE pulse sequence, given in Eqn. 3.4, is an approximation derived from a complex theoretical derivation of a simple MPRAGE sequence by [5]. In practice, the MPRAGE sequence implemented on the scanner has many additional parameters that are not accounted for by this derivation. Therefore, in most cases, our estimates of the pulse sequence parameters are not particularly close to the parameters recorded in the image headers. It turns out that this is not important for the problem at hand, since we do not need to know the exact imaging parameters of any given pulse sequence. Instead, we need to be able to generate a realistic synthetic image from our atlas that has the same intensity characteristics as the subject image using the estimated parameters and approximate pulse sequence. Approximate imaging equations and their estimated parameters are sufficient for this purpose.

3.2.2 Synthesizing a New Atlas Image with Subject Pulse Sequence Estimates

Now we describe Step 2 of the process, as illustrated in Fig. 3.1. It is unlikely that our atlas would contain an image with the exact pulse sequence parameters Θ_{b_i} estimated from b_i , which is why we synthesize an atlas image with the same parameters. Using the estimated imaging parameters, $\hat{\Theta}_{b_i}$, we apply the pulse sequence to the atlas β . The atlas $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ consists of a set of co-registered brain MR images of a single brain with different pulse sequences. It also consists of a_{P_D} , a_{T_1} , and a_{T_2} , the quantitative P_D , T_1 , and T_2 maps for the atlas. Thus, we can directly apply the subject pulse sequence, Γ_i , and its estimated $\hat{\Theta}_{b_i}$ to synthesize a new atlas image a_{b_i} . We thereby create an atlas image that looks as if the atlas brain was imaged with the pulse sequence Γ_i with parameters $\hat{\Theta}_{b_i}$. We need this intermediate step so that we can learn the intensity transformation between the subject pulse sequence Γ_i and the reference pulse sequence Ψ_r in a common image space, which is the atlas image space.

In practice, the atlas collection \mathcal{B} may lack the quantitative P_D , T_1 , and T_2 maps—the relaxometry sequence data is generally not available for most clinical data. We can approximately estimate these maps from the images present in the atlas collection by solving for P_D , T_1 , and T_2 at each voxel. Since we are estimating three quantities in $\beta(\mathbf{x}) = [P_D(\mathbf{x}), T_1(\mathbf{x}), T_2(\mathbf{x})]$, we require at least three atlas images, a_u , a_v , and a_w . We reiterate that all images are co-registered using rigid registration. From the

CHAPTER 3. Ψ -CLONE

method described in Section 3.2.1, we can estimate $\hat{\Theta}_{a_u}$, $\hat{\Theta}_{a_v}$, and $\hat{\Theta}_{a_w}$. For each voxel \mathbf{x} , we have three intensity values from three images, thus leading to three equations,

$$\begin{aligned}\Psi_u(\boldsymbol{\beta}(\mathbf{x}); \hat{\Theta}_{a_u}) &= a_u(\mathbf{x}), \\ \Psi_v(\boldsymbol{\beta}(\mathbf{x}); \hat{\Theta}_{a_v}) &= a_v(\mathbf{x}), \\ \Psi_w(\boldsymbol{\beta}(\mathbf{x}); \hat{\Theta}_{a_w}) &= a_w(\mathbf{x}).\end{aligned}\tag{3.6}$$

This system of simultaneous nonlinear equations can be solved by Newton's method for each voxel to provide us with an estimate $\hat{\boldsymbol{\beta}}(\mathbf{x})$. The component parts of $\hat{\boldsymbol{\beta}}(\mathbf{x})$ are $[\hat{\beta}_1(\mathbf{x}), \hat{\beta}_2(\mathbf{x}), \hat{\beta}_3(\mathbf{x})]$ that are our estimates of $[P_D(\mathbf{x}), T_1(\mathbf{x}), T_2(\mathbf{x})]$. Thus, we add the images $a_{\hat{\beta}_1}$, $a_{\hat{\beta}_2}$, and $a_{\hat{\beta}_3}$ to our atlas to represent the P_D , T_1 , and T_2 quantitative maps, respectively. This calculation needs to be done only once, during the construction of a suitable atlas. We used this approach as an intermediate step to perform intensity standardization [63].

3.2.3 Learning and Applying Nonlinear Regression on Image Patches

Having synthesized the atlas image a_{b_i} that has the pulse sequence characteristics of the subject image b_i , we next learn the intensity transformation that will convert the intensities in a_{b_i} to the corresponding intensities in the target atlas image a_r . We

CHAPTER 3. Ψ -CLONE

depict this as Step 3 in Fig. 3.1. This is achieved through a nonlinear regression by considering the image patches of a_{b_i} together with the corresponding central voxel intensities in a_r . We extract $p \times q \times r$ sized patches from a_{b_i} , centered at the v^{th} voxel—in our experiments $p = q = r = 3$. We stack the 3D patch into a $d \times 1 = 27 \times 1$ vector and denote it by $\mathbf{f}_v \in \mathbb{R}^d$, which we refer to as a feature vector of the v^{th} voxel. The corresponding intensity at the v^{th} voxel of a_r is denoted by y_v and acts as the dependent variable in our regression; we denote these training data pairs as $\langle \mathbf{f}_v, y_v \rangle$. We use patches as intensity features to learn this transformation. A small patch captures the local context at a voxel and ensures spatial smoothness. We could use other synthesis methods like MIMECS [4] to learn this transformation, however we chose to use the random forest regression as it was shown to produce better quality synthetic images at an order of magnitude lower computation time [30]. Using small patches and random forest regression for synthesis was an early version of REPLICA.

We use a bagged ensemble of regression trees to learn this nonlinear regression [47]. This standalone regression ensemble for synthesis was previously explored in [30]. A single regression tree learns a nonlinear regression by partitioning the d -dimensional space. This is done by performing binary comparison splits at each node of the tree, based on a particular attribute value which is compared to a threshold. The tree is built by minimizing the least squares criterion during training. The growth of the tree is limited by fixing the maximum number of vectors allowed at each leaf, in our experiments this was limited to five data vectors. This stops a tree from

CHAPTER 3. Ψ -CLONE

becoming too deep and hence over-fitting the training data. A single regression tree is considered a weak learner and in general has higher error [47], therefore we use a bagged ensemble of regression trees (30 in our experiments), which reduces errors by bootstrap aggregation [47]. To create a bootstrapped data set, a training sample is picked at random with replacement N times, where N is the size of the training data, $\sim 10^6$ in our experiments.

Once the training is complete, the trained regression ensemble transforms intensities of a_{b_i} to those of a_r . This ensemble is used to synthesize the subject image \hat{b}_r by extracting image patches from b_i and applying the trained regression ensemble to each patch to synthesize the corresponding \hat{b}_r voxel intensities, which is the last step, Step 4 in Fig. 3.1.

Thus, starting with a subject image and a set of atlas images, we estimate the pulse sequence parameters of the subject image, create an additional atlas image by applying those parameters to the atlas quantitative images, learn an intensity transformation from the additional atlas image to the target atlas contrast image using random forest regression, and lastly apply the regression to the given subject image to create a synthetic image of the required contrast. Ψ -CLONE is summarized in Algorithm 3. In the following sections, we describe validation experiments and applications of Ψ -CLONE in different image analysis contexts.

Algorithm 3 Ψ -CLONE

- 1: Data: Subject image b_i . Co-registered atlas images, target contrast a_r , NMR maps, a_{PD} , a_{T_1} , and a_{T_2}
 - 2: Perform tissue classification of subject b_i using fuzzy c-means clustering
 - 3: Calculate mean tissue intensities of CSF (\bar{b}_{iC}), GM (\bar{b}_{iG}) and WM (\bar{b}_{iW})
 - 4: Estimate pulse sequence parameters $\hat{\Theta}_{b_i}$ of subject pulse sequence by solving a nonlinear simultaneous system of equations in Eqn. 3.5
 - 5: Synthesize a new atlas image $a_{b_i}(\mathbf{x}) = \Gamma_i(\{a_{PD}(\mathbf{x}), a_{T_1}(\mathbf{x}), a_{T_2}(\mathbf{x})\}, \hat{\Theta}_{b_i})$
 - 6: Extract $3 \times 3 \times 3$ patches from a_{b_i}
 - 7: Extract corresponding voxel intensities from a_r to create training data
 - 8: Train a random forest on this training data to predict intensities in a_r
 - 9: Extract patches from subject image b_i and apply learned random forest to predict synthetic image \hat{b}_r
-

3.3 Computational Phantom Experiments

The goal of our method is to produce synthetic images that are useful substitutes for real images for image processing tasks. Thus, one aspect of algorithm performance evaluation consists of using image quality metrics to compare synthetic images with known ground truth images. The ground truth images are simulated with known pulse sequence parameters on brain voxels with known NMR parameters. We compare our synthetic images with these known simulated images to validate our method in a controlled experimental setting. In this section, we used the Brainweb image phantom for intensity standardization and synthesis of T_2 w images from T_1 w images.

3.3.1 Brainweb SPGR: Estimating a_{b_i}

In this validation experiment the atlas set \mathcal{A} consisted of images from the Brainweb [72] phantom, consisting of:

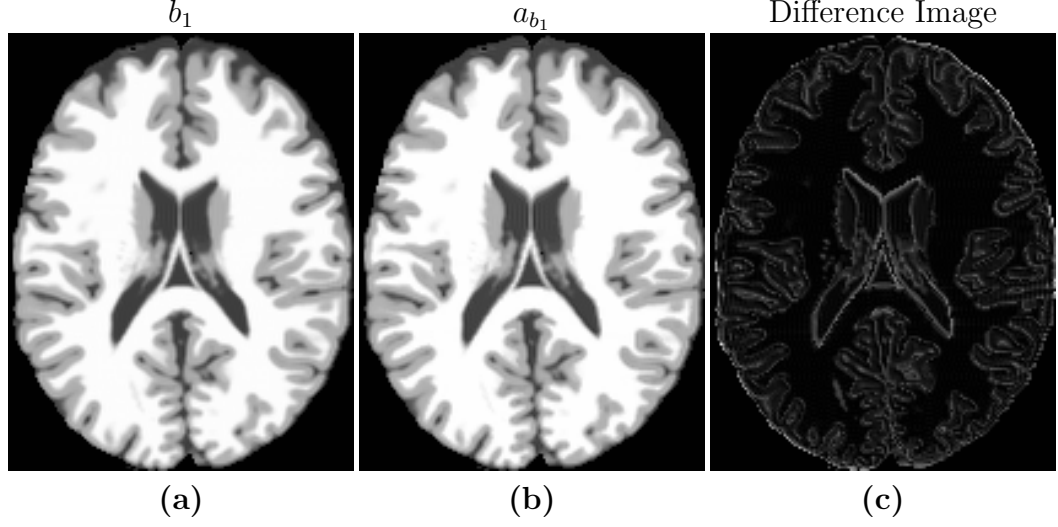


Figure 3.1: Shown are **(a)** an SPGR image ($TR = 18$ ms, $TE = 10$ ms, $\alpha = 45^\circ$ with 0% additive noise) which we use as our subject image b_1 , the maximum intensity value is 1080, the **(b)** new atlas image, a_{b_1} , with pulse sequence parameters estimated from b_1 , and **(c)** the difference image $|a_{b_1} - b_1|$, the maximum value is 5.

a_1 : SPGR image (1.5 T, $TR = 18$ ms, $\alpha = 30^\circ$, $TE = 10$ ms) with 0% noise,

a_{T_1} : Quantitative T_1 map derived from two different SPGR images, with two different flip angles ($TR = 100$ ms, $TE = 15$ ms, $\alpha_1 = 15^\circ$ and $\alpha_2 = 30^\circ$) using the dual flip angle method [76],

a_{T_2} : Quantitative T_2 map derived from a DSE sequence ($TR = 6653$ ms, $TE_1 = 30$ ms, $TE_2 = 80$ ms) by the two point method [50],

a_{P_D} : Quantitative P_D map derived from the reference SPGR and the DSE images using the method described in Section 3.2.2.

The aim of this experiment is to validate Step 1 and Step 2 of Ψ -CLONE. We do this by taking several images as a potential subject image and carry out just the first two

CHAPTER 3. Ψ -CLONE

steps of Ψ -CLONE which results in the image a_{b_i} . Specifically the subject images were:

b_1 : SPGR image (1.5 T, TR = 18 ms, $\alpha = \{15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ\}$, TE = 10 ms) with 0% noise. (Parameters, excepting TR, not provided to the algorithm)

We note that the subject imaging parameters are shown for the sake of the reader and are not provided to the algorithm (except TR).

The first step of Ψ -CLONE estimates the imaging parameters from the subject image. As both the atlas and subject images come from the same phantom—they have the same *phantom anatomy* and *NMR parameters*—this allows us to validate Step 2, in which we create a synthetic atlas image using the pulse sequence parameters of the subject image. Since the anatomy of subject and atlas is the same in this special case, we can directly compare the synthetic atlas image with the subject image to validate if the pulse sequence parameters are producing an identical image. We would prefer to compare the estimated pulse sequence parameters with the known true parameters. Unfortunately, this comparison is not suitable for all parameters because we use theoretical equations or their approximations to estimate the pulse sequence parameters and the actual simulation or scanner implementation can be more complex with a larger number of parameters. As a small example, we carried out an experiment on the Brainweb phantom data to measure the error between the true parameters and the estimated parameters. We simulated different SPGR images by keeping the repetition time TR = 18 ms fixed, and varying the flip angle. We next

CHAPTER 3. Ψ -CLONE

estimated the imaging parameters of these images by the method described above. The estimated flip angles for these images and the true flip angles used to simulate these images are recorded in Table 3.1. As can be seen, these estimates are close to but not equal to the truth. This error increases in realistic settings.

Table 3.1: Flip angles used in Brainweb SPGR simulation vs estimated Flip angles after fitting.

True flip angle ($^{\circ}$)	Estimated flip angle ($^{\circ}$)
30	32.08
45	48.70
60	65.08
75	84.79
90	106.57

Hence we focus on the images that these parameter estimates create when applied to the estimated P_D , T_1 , and T_2 values. We compute the root mean squared error (RMSE) and peak signal to noise ratio (PSNR) between the new atlas image (a_{b_i}) and the subject image (b_i), as shown in Table 3.2. Figure 3.1 shows the input b_1 image and the estimated a_{b_1} , along with the difference image for the case of $\alpha = 45^{\circ}$ (third column of Table 3.2). The images are very similar to each other with small differences at tissue boundaries. The differences are only visible at the boundaries because, as the histogram in Fig. 3.3 shows, Brainweb phantom images with no noise typically have very homogenous intensity distribution in each of the tissue classes (WM, GM, CSF). Hence the error inside the tissues is very close to zero, but is slightly higher at the intermediate intensity voxels that are closer to the boundaries. We note that RMSE and PSNR are computed over the non-zero voxels in the image. RMSE is reported in

CHAPTER 3. Ψ -CLONE

terms of percentage error with respect to the maximum intensity in the image. The high PSNR values in Table 3.2 along with the visual result in Fig. 3.1 confirm that the theoretical pulse sequence equations and the underlying quantitative P_D , T_1 , and T_2 maps produce images that are close to the ground truth images, thus validating the first two steps of our algorithm.

This experiment establishes that even though Ψ -CLONE is unable to estimate the pulse sequence parameters exactly, the estimated parameters produce an image that is close enough to the ground truth when applied to the NMR parameters.

Table 3.2: RMSE between a_{b_1} and b_1 (as a % w.r.t max intensity) and PSNR (dB) values for Brainweb T_1 w SPGR atlas synthesis for subject pulse sequences with varying flip angles.

Flip angle	15°	30°	45°	60°	75°	90°
RMSE %	5.09	2.49	1.63	1.18	0.88	0.68
PSNR	29.82	34.24	35.42	35.76	35.88	35.94

3.3.2 Brainweb SPGR Intensity Standardization

The goal of this experiment is to standardize a Brainweb SPGR subject image to an atlas target SPGR image. Different subject SPGR images were simulated using different input pulse sequence parameters. Using the same Brainweb atlas collection described in Section 3.3.1, we evaluated the results of Steps 3 and 4 of our method, the regression based image synthesis. Our subject images are:

b_1 : SPGR image (1.5 T, TR = 18 ms, $\alpha = \{15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ\}$, TE = 10 ms)

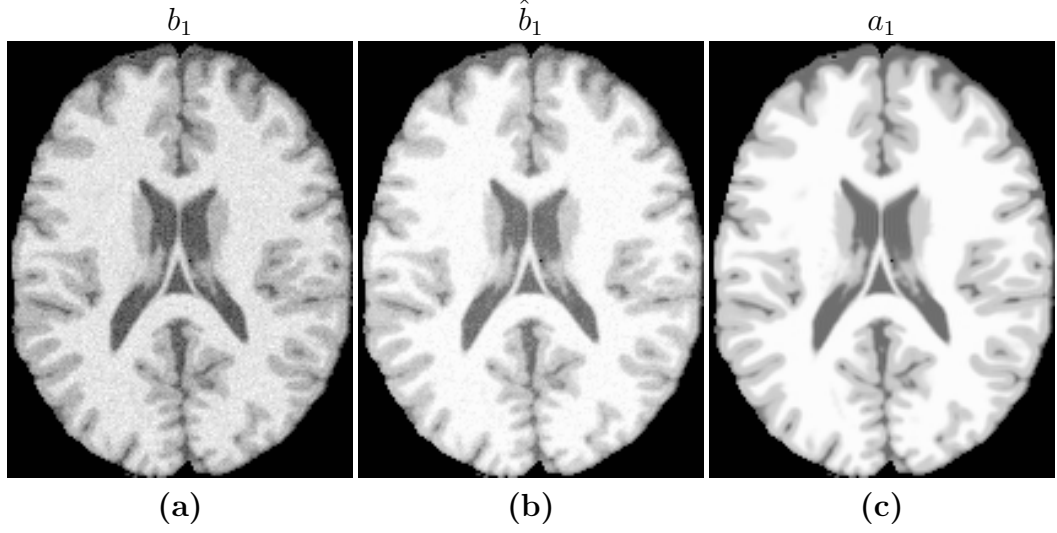


Figure 3.2: Shown are (a) an SPGR image ($TR = 18\text{ms}$, $TE = 10\text{ms}$, $\alpha = 45^\circ$ with 3% additive noise) which we use as our subject image b_1 , the (b) reconstruction, \hat{b}_1 , of the subject image with the same pulse sequence as used to image (c) the atlas target image a_1 ($TR = 18\text{ms}$, $\alpha = 30^\circ$, $TE = 10\text{ms}$ with 0% noise).

with $\{0\%, 3\%\}$ noise levels. (Parameters, excepting TR , not provided to the algorithm)

As the target atlas and subject images are both from the SPGR pulse sequence, this is a special case of synthesis, normally referred to as intensity standardization or normalization. We can compare the standardized subject image to the target atlas image directly as they have the same anatomy. We also compared the performance of our standardization with a landmark-based piecewise linear scaling method (UPL) [2], reporting PSNR for the input subject image in Table 3.3. UPL estimates the landmarks in the images for each of the three tissue classes (CSF, GM, and WM) and then uses a piecewise linear scaling between the target and subject histogram to normalize the images.

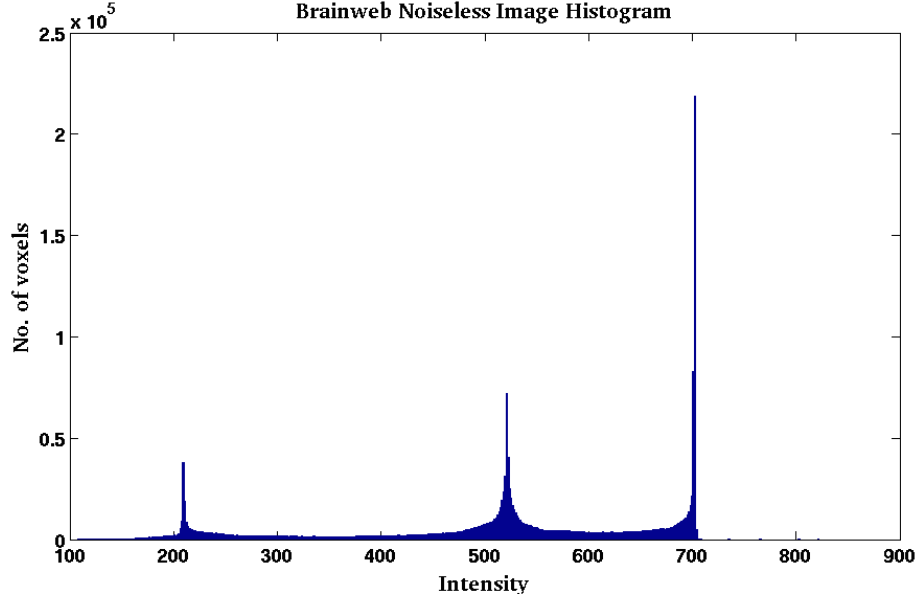


Figure 3.3: Histogram of a typical noiseless Brainweb phantom

The histogram of a noise free Brainweb SPGR phantom is shown in Fig. 3.3. The histogram landmarks consist of very sharp peaks, indicating that a large number of voxels have very similar intensities. A piecewise linear transform can map exactly between two (subject and atlas) such histograms of noiseless Brainweb phantoms. This explains why the UPL method performs better than our method in the 0% noise case. However, with the introduction of noise our method outperforms UPL in four out of the five cases, as shown in Table 3.3.

Using simulated images we can thus show that Ψ -CLONE is able to standardize to a reference better than UPL, in noisy images.

CHAPTER 3. Ψ -CLONE

Table 3.3: PSNR (dB) values between a_1 and \hat{b}_1 , for standardization of Brainweb phantoms with varying flip angles ($^\circ$) and noise levels are shown for UPL and Ψ -CLONE. In the noise free case, UPL is better, however the introduction of noise causes UPL results to deteriorate.

Flip Angle	0% Noise		3% Noise	
	UPL	Ψ -CLONE	UPL	Ψ -CLONE
15 $^\circ$	37.95	27.95	27.27	26.00
30 $^\circ$	62.66	35.90	28.93	30.96
45 $^\circ$	49.26	37.32	28.88	31.48
60 $^\circ$	46.94	37.67	28.81	31.39
75 $^\circ$	46.08	37.79	28.78	31.52

3.3.3 Brainweb T_2 w Synthesis

Ψ -CLONE was next used to synthesize a T_2 w image from a subject SPGR image, using the Brainweb atlas collection described in Section 3.3.1 with the addition of:

a_2 : T_2 w image from the second echo of a DSE (1.5 T, TR = 3000 ms, TE₁ = 17 ms, TE₂ = 80 ms) with 0% noise.

We use the following input subject Brainweb SPGR images:

b_1 : SPGR image (1.5 T, TR = 18 ms, $\alpha = \{15^\circ, 30^\circ, 45^\circ, 60^\circ\}$, TE = 10 ms) with $\{0\%, 1\%, 3\%, 5\%\}$ noise levels. (Parameters, excepting TR, not provided to the algorithm)

The UPL method is unable to synthesize a T_2 w image from an SPGR, as it is primarily an intensity standardization approach; thus, for this experiment we compare Ψ -CLONE to MIMECS [4, 28]. MIMECS is a state-of-the-art MR contrast synthesis approach that uses an example-based sparse reconstruction from image patches to

CHAPTER 3. Ψ -CLONE

perform intensity standardization and missing tissue contrast recovery. MIMECS, unlike Ψ -CLONE, is blind to the MR physics and solves the synthesis problem based on patch similarity between the subject and the atlases. As the atlas and the subject have the same phantom anatomy, an ideal synthesis would result in an image that is exactly equal to the atlas T_2w image. Thus, we evaluate the quality of synthesis by calculating the PSNR between the atlas image and the synthesized subject image. We also evaluate the quality of synthesis using the universal quality index (UQI) [53]. The results for this experiment are shown in Table 3.4. The top half of the table shows results for changing noise levels with a fixed flip angle (30°) for the input SPGR, while the bottom half shows the results for changing flip angles with 0% noise in the input SPGR. For both metrics, Ψ -CLONE outperforms MIMECS. An example of an input SPGR used in this experiment, the true T_2w image and the outputs of both MIMECS and Ψ -CLONE are shown in Fig. 3.4.

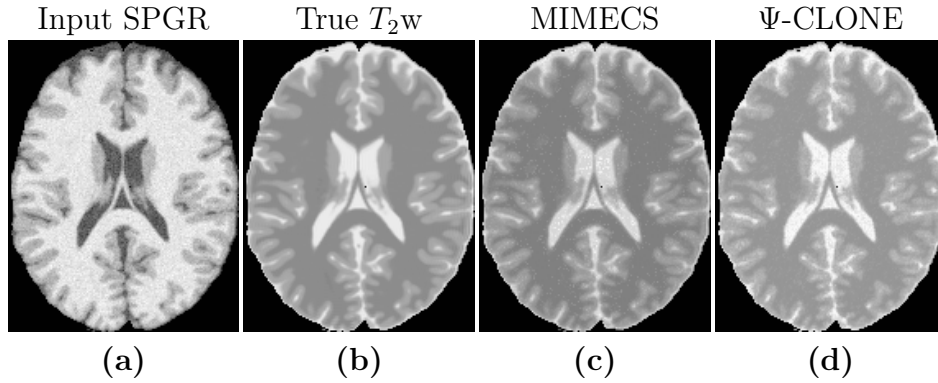


Figure 3.4: (a) An example input subject SPGR from which we synthesize a T_2w image. (b) The true T_2w image and the outputs of synthesis produced by (c) MIMECS and (d) Ψ -CLONE.

Table 3.4: PSNR (dB) and UQI values for Brainweb T_2 w synthesis with varying noise levels (%) and flip angles ($^\circ$) are shown for MIMECS and our algorithm (Ψ -CLONE).

Noise (%)		0	1	3	5
PSNR	MIMECS	26.25	19.96	19.09	19.21
	Ψ -CLONE	30.33	30.71	29.09	26.63
UQI	MIMECS	0.93	0.88	0.83	0.82
	Ψ -CLONE	0.95	0.94	0.91	0.88
Flip Angle ($^\circ$)		15	30	45	60
PSNR	MIMECS	24.51	26.25	25.27	24.82
	Ψ -CLONE	25.99	30.33	30.94	31.06
UQI	MIMECS	0.90	0.93	0.92	0.92
	Ψ -CLONE	0.90	0.95	0.96	0.96

3.4 Real Data Experiments

In this section, we present intensity standardization and image synthesis experiments performed on real datasets.

3.4.1 Human Stability Data

A normal, healthy human subject was imaged at weekly intervals using the same scanner and pulse sequence for nine weeks. We demonstrate that image segmentation is more consistent on data which is intensity standardized using Ψ -CLONE. To do this we standardize each time point to an atlas consisting of:

a_1 : MPRAGE image (3 T, TR = 10.3 ms, TE = 6 ms, $0.82 \times 0.82 \times 1.17$ mm³ voxel size),

CHAPTER 3. Ψ -CLONE

a_{T_1} : Quantitative T_1 map computed as described in Section 3.2.2,

a_{T_2} : Quantitative T_2 map derived as described in Section 3.2.2,

a_{P_D} : Quantitative P_D map derived as described in Section 3.2.2,

and each of the nine subject images is:

b_1 : MPRAGE image (3 T, TR = 10.3 ms, TE = 6 ms, $0.82 \times 0.82 \times 1.17$ mm³ voxel size. (Parameters, excepting TR, not provided to the algorithm)

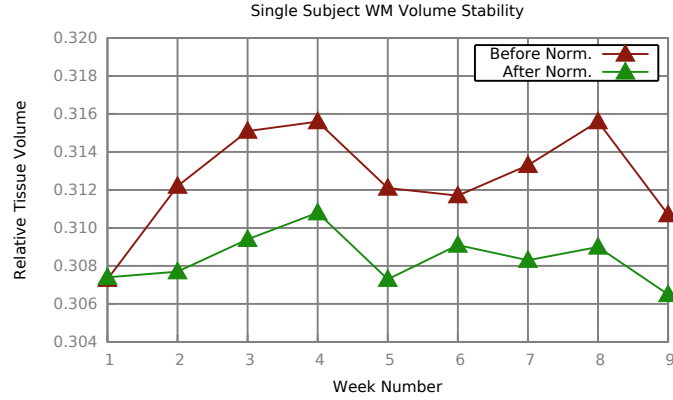


Figure 3.1: The brown plot illustrates relative WM volumes (with respect to the ICV) over nine weeks before any standardization while the green plot illustrates the same values after intensity standardization using Ψ -CLONE.

We compare the segmentations of the images pre- and post-standardization using Ψ -CLONE, based on segmentations generated by TOADS [77]. We specifically compare the relative tissue volumes (relative to the intra-cranial volume (ICV)), over the nine weeks. Ideally, a normal healthy subject should not present any changes in tissue volumes over such a short period of time.

Table 3.1: Coefficient of Variation (CV) of the relative tissue volumes ($\times 10^{-3}$) over the nine weeks on the original data and after intensity standardization with Ψ -CLONE for each of white matter (WM), cortical gray matter (Cort. GM), subcortical gray matter (Sub. GM), cerebrospinal fluid (CSF), and the ventricles (Vent.). Table 3.2 shows further breakdown of Sub. GM structures.

Vol. CV	Classes				
	WM	Cort. GM	Sub. GM	CSF	Vent.
Original	8.6	6.4	20.4	11.9	30.7
Ψ-CLONE	4.3	5.7	19.4	12.0	23.1

Fig. 3.1 shows the relative WM volumes before and after Ψ -CLONE was applied. Visually, it is apparent that the WM volumes change less when standardized images are segmented. Table 3.1 shows the coefficient of variation of relative tissue volumes for CSF, cortical GM, subcortical GM, ventricles, and cortical WM. The coefficients of variation from segmentation after standardization by Ψ -CLONE are smaller than those without standardization for cortical WM, cortical and subcortical GM, and ventricles, indicating that the segmentation is more stable. As there are only nine time-points there is insufficient data to determine significance. The subcortical gray matter class consists of the thalamus, caudate and putamen structures. We compared the volumes of these structures before and after standardization and calculated the coefficient of variation in both cases. Results are shown in Table 3.2. The thalamus volumes for Ψ -CLONE standardized images are most stable as evinced by the reduced coefficient of variation. The coefficient of variation decreases slightly for putamen and increases slightly for caudate. However the accuracy of LesionTOADS segmentations for subcortical structures is not as high as it is for the cortical GM or WM, hence these

CHAPTER 3. Ψ -CLONE

numbers may not be reliable indicators of segmentation consistency. Additionally, these are only nine time-points of a single subject, so we cannot claim statistical significance in these measurements at this point.

This experiment shows that, overall, Ψ -CLONE enables us to perform more consistent segmentations.

Table 3.2: Coefficient of Variation (CV) of the relative tissue volumes ($\times 10^{-3}$) over the nine weeks on the original data and after intensity standardization with Ψ -CLONE for each of caudate, putamen, and thalamus.

Vol. CV	Structures		
	Caudate	Putamen	Thalamus
Original	9.9	9.9	51.9
Ψ-CLONE	10.3	9.7	45.3

3.4.2 MR Intensity Scale Standardization for MS Patients

To allow us to have statistical power in our exploration of MR intensity scale standardization, we employ a cohort of 15 Multiple Sclerosis (MS) patients with 57 scans. Each patient has at least three scans (mean # of scans per subject is 3.8) acquired approximately a year apart. Preprocessing of the images included skull-stripping [58] and bias field inhomogeneity correction [78]. For this experiment our atlas consisted of:

a_1 : MPRAGE image (3 T, TR = 10.3 ms, TE = 6 ms, $0.82 \times 0.82 \times 1.17$ mm³ voxel

CHAPTER 3. Ψ -CLONE

size),

a_2 : T_2 w image from the second echo of a DSE (3 T, TR = 4177 ms, TE₁ = 12.31 ms, TE₂ = 80 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size),

a_3 : P_D w from the first echo of a DSE (3 T, TR = 4177 ms, TE₁ = 12.31 ms, TE₂ = 80 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size),

a_4 : FLAIR (3 T, TI = 11000 ms, TE = 68 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size)

a_{T_1} : Quantitative T_1 map computed as described in Section 3.2.2,

a_{T_2} : Quantitative T_2 map derived as described in Section 3.2.2,

a_{P_D} : Quantitative P_D map derived as described in Section 3.2.2,

and our subject image was:

b_1 : MPRAGE image (3 T, TR = 10.3 ms, TE = 6 ms, $0.82 \times 0.82 \times 1.17$ mm³ voxel size). (Parameters, excepting TR, not provided to the algorithm)

We use an atlas MPRAGE as the target pulse sequence to which we standardize the 57 data sets. We treat each data set independently, handling the intensity standardization as a cross-sectional task. To validate the intensity standardization, we segmented [55] the original MPRAGE datasets giving us ten labeled structures: ventricles, sulcal CSF, cerebellar GM (Cereb. GM), cortical GM (Cort. GM), thalamus, putamen, cerebellar WM (Cereb. WM), cortical WM (Cort. WM), and lesions.

CHAPTER 3. Ψ -CLONE

Using these structures as reference, we compared the mean intensity within these structures prior to standardization and after standardization with UPL—using the target MPRAGE in our atlas as a standardization target—and our method (Ψ -CLONE). We note that the atlas images did not have lesions. For applications like T_1w intensity standardization, we observed that the presence or absence of lesion samples in the training data did not affect the synthesis result. The primary reason is that white matter lesion intensities are similar to GM (and rarely CSF) in T_1w contrasts. Thus, the database has a large number of normal appearing GM and CSF samples available to reconstruct the lesion intensities.

Table 3.3: The mean intensity value for the atlas used in the MS standardization experiment are shown for ten structures. We also show the mean ($\times 10^4$) and std ($\times 10^4$) (over 57 images) of the average intensity value for each structure, based on the original unnormalized data (**Original**) and after standardization with both **UPL** [2] and our method (**Ψ -CLONE**).

	Atlas Mean	Original Mean Std	UPL Mean Std	Ψ -CLONE Mean Std
Ventricles	3.92	5.41 1.815	4.25 0.664	4.11 [†] 0.271*
Sulcal CSF	2.50	3.95 1.132	3.11 0.543	2.97 0.478
Lesions	—	18.54 5.880	14.62 1.293	14.51 1.516
Cereb. GM	12.27	15.86 5.295	12.68 1.013	12.27 [†] 0.433*
Cort. GM	10.47	13.40 4.722	10.62 1.199	10.27 [†] 0.260*
Caudate	12.93	16.98 5.866	13.53 1.130	13.10 [†] 0.526*
Thalamus	14.99	21.20 7.036	16.96 1.010	16.63 [†] 0.548*
Putamen	15.53	20.26 7.286	16.23 1.196	15.77 [†] 0.485*
Cereb. WM	20.79	27.66 9.333	21.05 0.209	22.02 0.266
Cort. WM	20.53	25.77 8.778	20.22 0.266	20.41 [†] 0.186*

[†] Difference between atlas mean and normalized mean is significantly smaller than UPL (α level of 0.05) using a two-sample one-tailed T-test.

* Standard deviation is significantly smaller than UPL (α level of 0.05) based on a two-sample F-test.

Results are shown in Table 3.3. The mean intensity values of the original images

CHAPTER 3. Ψ -CLONE

and the Ψ -CLONE standardized images recorded in Table 3.3, demonstrate that our method is moving the MS data intensities closer to the atlas intensities, as desired by standardization. A one tailed F-test on the mean structure intensities after standardization shows that the standard deviation of the mean structure intensities across the 57 datasets for Ψ -CLONE is significantly smaller in comparison to UPL for seven of the ten structures. We also note that the statistical significance does not change if the segmentation is done on the original data or on the standardized versions.

Table 3.4: Contrast values between neighboring structures for original, synthetic, and atlas images. * indicates that the contrast in synthetic images higher than the original images (statistically significant using Student’s one-tailed T test with $p < 0.05$).

Struct. Boundaries	Structure Boundary Contrasts		
	Original	Ψ -CLONE	Atlas
Cort CSF–Cort GM	0.594	0.604*	0.625
Cort GM–WM	0.480	0.482	0.489
WM–Lesions	0.279	0.300	—
WM–Caudate	0.341	0.367*	0.370
WM–Putamen	0.214	0.219	0.240
WM–Thalamus	0.176	0.208*	0.269
WM–Ventricles	0.846	0.863*	0.878
Ventricles–Caudate	0.765	0.784*	0.804
Ventricles–Thalamus	0.814	0.827*	0.834

Next, we ran LesionTOADS segmentation on images before and after standardization. These segmentations were used to calculate image contrast between neighboring structures to indicate the effect of synthesis-based standardization on subsequent segmentation. We looked at the following structure boundaries, Cortical CSF-Cortical

CHAPTER 3. Ψ -CLONE

GM, Cortical GM-WM, WM-lesions, WM-Ventricles, WM-Caudate, WM-Putamen, WM-Thalamus, Caudate-Ventricles, and Thalamus-Ventricles and have tabulated the results in Table 3.4. We can show that on average these contrast values for synthetic images are higher than the real images, significantly in most cases. The contrast values for synthetic images are also closer to the reference contrast values for the same structures, than the original images. We have defined contrast between two neighboring structures f and g as $\frac{\mu(f)-\mu(g)}{\mu(f)}$, where $\mu(f)$ is the mean intensity of the brighter structure f and $\mu(g)$ is the mean intensity of the structure g . The higher the contrast, the easier it is to differentiate structures.

Thus, in this experiment we have demonstrated that Ψ -CLONE is able to standardize a large dataset so that the average tissue intensities are closer together for the whole dataset than before standardization. This allows us to perform more consistent segmentation.

3.4.3 T_2w Synthesis from Real MPRAGE Data

To demonstrate T_2w synthesis from MPRAGE on real data, we used the 21 subjects from the publicly available Multi-Modal MRI Reproducibility Resource (MMRR) [50]. We held out a single subject as the atlas:

a_1 : MPRAGE image (3 T, TR = 6.7 ms, TE = 3.1 ms, TI = 842 ms, $1.0 \times 1.0 \times 1.2$ mm³ voxel size),

CHAPTER 3. Ψ -CLONE

a_2 : T_2 w image from the second echo of a DSE (3 T, TR = 6653 ms, TE₁ = 30 ms, TE₂ = 80 ms, $1.5 \times 1.5 \times 1.5$ mm³ voxel size),

a_3 : P_D w image from the first echo of a DSE (3 T, TR = 6653 ms, TE₁ = 30 ms, TE₂ = 80 ms, $1.5 \times 1.5 \times 1.5$ mm³ voxel size),

a_{T_1} : Quantitative T_1 map computed from two flip angles (3 T, TR = 100 ms, TE = 15 ms, $\alpha_1 = 15^\circ$, $\alpha_2 = 60^\circ$, $1.5 \times 1.5 \times 1.5$ mm³ voxel size),

a_{T_2} : Quantitative T_2 map derived from a two-point method (3 T, TR = 6653 ms, TE₁ = 30 ms, TE₂ = 80 ms, $1.5 \times 1.5 \times 1.5$ mm³ voxel size),

a_{P_D} : Quantitative P_D map derived from the MPRAGE and the DSE images using the method described in Section 3.2.2.

Our subject image is:

b_1 : MPRAGE image (3 T, TR = 6.7 ms, TE = 3.1 ms, TI = 842 ms, $1.0 \times 1.0 \times 1.2$ mm³ voxel size). (Parameters, excepting TR, not provided to the algorithm)

We note that the atlas image did not have lesions. For an application like T_2 w synthesis, we again observed that the presence or absence of lesion samples in the training data did not affect the synthesis result. The primary reason is that white matter lesion intensities are similar to GM (and rarely CSF) in T_1 w and T_2 w contrasts. Thus, the database has a large number of normal appearing GM and CSF samples available to reconstruct the lesion intensities.

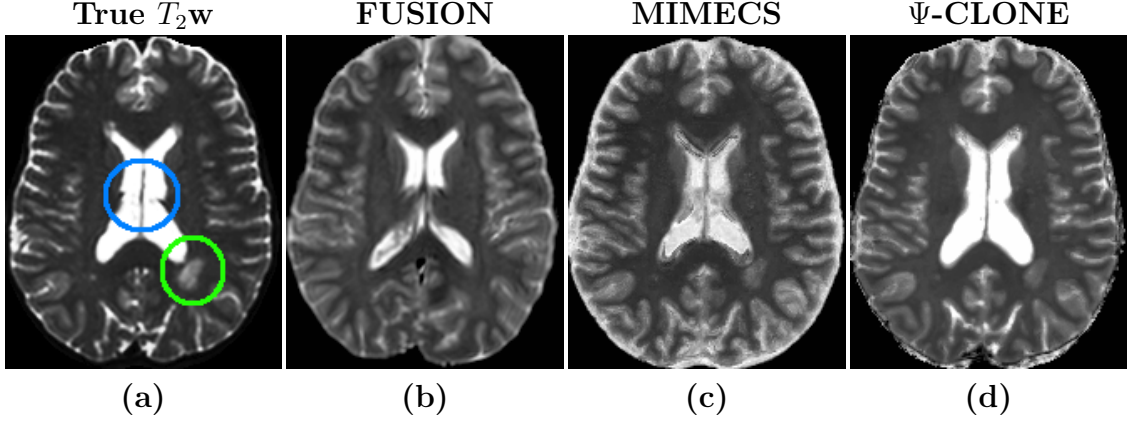


Figure 3.2: Shown are (a) the true T_2w image, and the synthesis results from the MPRAGE for each of (b) FUSION, (c) MIMECS, and (d) Ψ -CLONE (our method). The lesion (in the green circle) and the ventricles (in the blue circle) in the true image are synthesized by MIMECS and Ψ -CLONE, but not by FUSION.

The remaining 20 available subjects each have two MPRAGE acquisitions and two corresponding DSE images which are co-registered to the MPRAGE. These images were acquired on the same scanner within a short duration of each other. For each of these 40 images (20 subjects \times 2 MPRAGE scans) we synthesized a T_2w image. As the atlas was imaged on the same scanner we directly compare the synthesized T_2w image with the true T_2w image from the same scanning session, using PSNR and UQI. We compared to MIMECS [28] and to a deformable registration-based synthesis. To carry out synthesis using deformable registration, the atlas image is registered deformably to the subject image of the same contrast. The same deformation is then applied to the atlas image of the desired contrast to produce the synthetic subject image. We use the state-of-the-art registration method SyN [79] for this synthesis and refer to this method as FUSION. We used only one atlas for FUSION in this experiment as the multi-atlas FUSION was not available when these experiments were done. Table 3.5 shows the

CHAPTER 3. Ψ -CLONE

PSNR and UQI for these three methods. We observe that our method (Ψ -CLONE) provides a significantly ($\alpha < 0.01$, using the right-tailed two sample t-test) better quality synthesis in comparison to both MIMECS and FUSION. Figure 3.2 shows the

Table 3.5: Mean and standard deviation (Std. Dev.) of the PSNR and UQI values for synthesis of T_2w images from 40 MPRAGE scans.

	PSNR	UQI
	Mean (Std. Dev.)	Mean (Std. Dev.)
FUSION	16.59 (1.35)	0.64 (0.06)
MIMECS	15.01 (0.84)	0.78 (0.03)
Ψ -CLONE	18.59 (1.09)*	0.79 (0.02)*

* Statistically significantly better than either of the other two methods (α level of 0.01) using a right-tailed test.

results for each synthesis approach in comparison to the ground truth image. Though the PSNR values for FUSION are better than MIMECS (see Table 3.5), the FUSION synthesis result is anatomically incorrect—the ventricle boundary is incorrect and lesions posterior to the ventricles are not synthesized. As the output of FUSION is based on deformably registering the atlas to the subject, if the atlas does not contain certain tissue features—lesions, for example—then the synthesized subject will not contain them as well. The lesion in Fig. 3.2 is a white matter lesion. We know that lesion boundaries appear slightly different in the MPRAGE than in the real T_2w image, and hence it cannot be perfectly reproduced in the synthetic T_2w image. However we aim to synthesize it as correctly as possible and do a better job than currently available synthesis algorithms. The result of MIMECS is quite noisy and Ψ -CLONE yields an image that is most visually similar to the true image. The values in Table 3.5

CHAPTER 3. Ψ -CLONE

were obtained on the MMRR dataset that was processed differently than the one used for the comparison experiment for REPLICA. Therefore values in Table 3.5 and in Table 2.1 are unfortunately not comparable. However, for this particular experiment we would expect REPLICA to perform better than Ψ -CLONE for two reasons: 1) The atlas and the subject images are from the same pulse sequence acquisition and the intensity standardization in Ψ -CLONE may not offer any advantages, and 2) the random forest in REPLICA is better tuned to this task than the one used in Ψ -CLONE.

We also used the atlas images of this dataset to evaluate our estimation procedure for the intrinsic parameters T_2 and T_1 . The median T_2 values calculated by the two point method [50] using the DSE images were 76 ms for WM, 85 ms for GM, and 175 ms for CSF. We used our estimation procedure (as described in Section 3.2.2) and the estimated median T_2 values obtained were 76 ms for WM, 91 ms for GM, and 762 ms for CSF. Both CSF distributions have a very large standard deviation ($\sim 10^4$), due to numerical errors. The intensities determined by the imaging equations also plateau off after a certain T_2 value due to their inverse exponential nature (see Eq. 3.2). Thus the intensities produced for high enough T_2 values are very close to each other. The T_1 map was estimated via two flip angle spoiled gradient images, as mentioned in the atlas description. This is not an ideal approach as the images acquired were noisy and the flip angle calibration is not considered accurate enough. The median T_1 values thus calculated using the dual flip angle image were 775 ms for WM, 1074 ms for GM,

and 1616 ms for CSF. Our estimation procedure returned the following median T_1 values of 779 ms for WM, 1151 ms for GM, and 2916 ms for CSF. As with T_2 values, the intensities produced by high T_1 values plateau off after a certain point due to the inverse exponential dependence on T_1 (see Eqs. 3.2, 3.3, 3.4). Thus, despite being slightly different from the expected values, our estimated T_2 and T_1 values are good enough to provide a realistic synthesis.

3.5 Further Synthesis Applications

In this final experimental section, we present additional results that demonstrate potential uses of Ψ -CLONE.

3.5.1 Synthesizing Higher Resolution T_2 w Data

Example-based synthesis of high resolution brain MR images has been explored in many recent works [1, 24, 32, 70]. We applied Ψ -CLONE to synthesize higher resolution T_2 w images than those acquired on the scanner. Our atlas collection \mathcal{A} is:

a_1 : MPRAGE image (3 T, TR = 6.7 ms, TE = 3.1 ms, TI = 842 ms, $1.1 \times 1.1 \times 1.1$ mm³ voxel size),

a_2 : T_2 w image from the second echo of a DSE (3 T, TR = 6653 ms, TE₁ = 30 ms, TE₂ = 80 ms, $1.1 \times 1.1 \times 1.1$ mm³ voxel size),

CHAPTER 3. Ψ -CLONE

and our subject images come from our MS cohort:

b_1 : MPAGE image (3 T, TR = 10.3 ms, TE = 6 ms, $0.82 \times 0.82 \times 1.17$ mm³ voxel size). (Parameters, excepting TR, not provided to the algorithm)

Pulse sequences like the DSE or FLAIR tend to have large TR or TI values to achieve the right contrast. To reduce the scan time while imaging patients, these pulse sequences are usually acquired at a lower resolution than a T_1 w sequence such as MPAGE. Multimodal analysis of such datasets requires all images to exist in the same coordinate system at the same digital resolution. This is usually achieved by upsampling the low resolution scans to the high resolution ones using interpolation, which results in blurring of the image data. Using Ψ -CLONE, we can synthesize a T_2 w image from the high resolution MPAGE. This synthetic image will have the same resolution as that of the MPAGE and hence can replace the acquired low resolution image. As we have no ground truth for the higher resolution T_2 w image we visually compare it with the acquired T_2 w image. The acquired T_2 w image has a through-plane resolution of 2.2 mm whereas the subject MPAGE, and consequently the synthetic T_2 w image have a through-plane resolution of 1.1 mm. Both are shown in Fig. 3.1. The quality and resolution of the newly synthesized image is visually superior to the original acquisition.

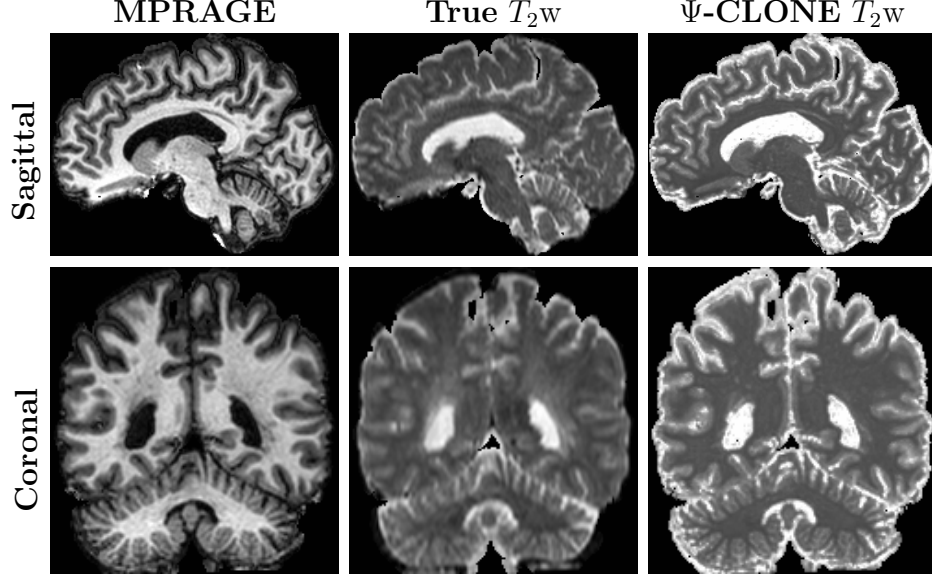


Figure 3.1: The MPRAGE has a through-plane resolution of 1.1 mm, while the original T_2w has through-plane resolution of 2.2 mm. This is evident as the true interpolated T_2w image shows blurring while the Ψ -CLONE synthesized image is crisp.

3.5.2 FLAIR Synthesis

FLAIR is the pulse sequence of choice when identifying white matter lesions present in MS patients. The lesions appear hyperintense with respect to the rest of the tissue which makes delineating them easier. Most leading lesion segmentation algorithms rely on the FLAIR image to provide intensity information for accurate classification [55, 80, 81]. FLAIR images are prone to certain artifacts (see Fig. 3.4 (e)) for a variety of reasons [16]. The long inversion times make it difficult to acquire high resolution scans in a short time. We demonstrate that if we acquire T_1w , P_Dw , and T_2w images of a subject, we have enough information to generate a synthetic FLAIR using Ψ -CLONE. The lesion intensity signature in FLAIR images is very distinct from the rest of the tissues. Hence the presence of lesion samples in the atlas set is essential

CHAPTER 3. Ψ -CLONE

in order to learn to reproduce it correctly. For this experiment, the atlas brain we use has a moderate lesion load. The atlas contained the following images:

a_1 : MPRAGE image (3 T, TR = 6.7 ms, TE = 3.1 ms, TI = 842 ms, $1.0 \times 1.0 \times 1.2$ mm³ voxel size),

a_2 : T_2 w image from the second echo of a DSE (3 T, TR = 6653 ms, TE₁ = 30 ms, TE₂ = 80 ms, $1.5 \times 1.5 \times 1.5$ mm³ voxel size),

a_3 : P_D w image from the first echo of a DSE (3 T, TR = 6653 ms, TE₁ = 30 ms, TE₂ = 80 ms, $1.5 \times 1.5 \times 1.5$ mm³ voxel size),

a_{T_1} : Quantitative T_1 map computed as described in Section 3.2.2,

a_{T_2} : Quantitative T_2 map derived as described in Section 3.2.2,

a_{P_D} : Quantitative P_D map derived as described in Section 3.2.2.

Our subject images are:

b_1 : MPRAGE image (3 T, TR = 6.7 ms, TE = 3.1 ms, TI = 842 ms, $1.0 \times 1.0 \times 1.2$ mm³ voxel size), (Parameters, excepting TR, not provided to the algorithm)

b_2 : T_2 w image from the second echo of a DSE (3 T, TR = 6653 ms, TE₁ = 30 ms, TE₂ = 80 ms, $1.5 \times 1.5 \times 1.5$ mm³ voxel size), (Parameters, excepting TR, not provided to the algorithm)

CHAPTER 3. Ψ -CLONE

b_3 : P_D w image from the first echo of a DSE (3 T, TR = 6653 ms, TE₁ = 30 ms, TE₂ = 80 ms, $1.5 \times 1.5 \times 1.5$ mm³ voxel size). (Parameters, excepting TR, not provided to the algorithm)

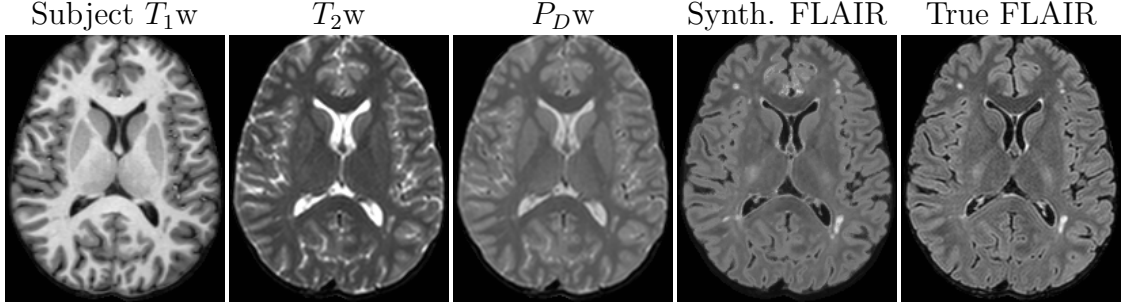


Figure 3.2: Subject input images along with the synthetic and true FLAIR images.

We would like to stress that the set of subject pulse sequence parameters (except TR) are unknown and these are extracted using the first step of Ψ -CLONE. Next, the new atlas T_1 w, P_D w, and T_2 w atlas images are generated by applying the respective pulse sequence equations to the atlas P_D , T_1 , and T_2 values. The following step of learning a patch-based regression is slightly different from the previous experiments. The feature vector \mathbf{b}_i for a voxel i , is created by concatenating the corresponding $3 \times 3 \times 3$ -sized patches centered on voxel i , from all three images. The dependent variable r_i is the corresponding target atlas FLAIR intensity at voxel i . Thus, the training data consists of pairs of $\langle \mathbf{b}_i, r_i \rangle$ from the extracted synthetic atlas images and the atlas FLAIR image. A nonlinear regression is learned using random forests and the trained regression is then applied to the extracted patches from the subject images to synthesize the subject FLAIR. Figure 3.2 displays the subject input images

and synthetic FLAIR with the true FLAIR for visual comparison.

We ran LesionTOADS on the real and synthetic FLAIR images shown in Fig. 3.2. The resulting segmentations are shown in Fig. 3.3. The lesion volume obtained from the real FLAIR using LesionTOADS was 3107.5 mm^3 whereas that obtained from a synthetic FLAIR was 5900.3 mm^3 . The excess seems to come from slightly enlarged regions with lesion-like intensities in the synthetic FLAIR. Our result, though visual in nature, is still a large improvement on the FLAIR synthesis result demonstrated in [4].

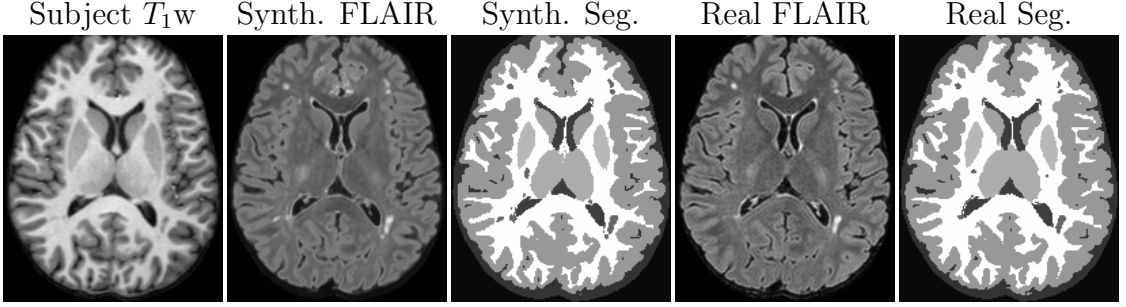


Figure 3.3: (a) Real T_1w image, (b) synthetic FLAIR, (c) LesionTOADS segmentation of real T_1w + synth. FLAIR, (d) real FLAIR, (e) LesionTOADS segmentation of real T_1w image + real FLAIR.

Synthesizing FLAIR images is especially useful when the original FLAIR has artifacts, which can lead to erroneous tissue segmentation. In the next experiment, we looked at data where the acquired FLAIR was of bad quality due to motion artifacts and created a synthetic FLAIR image for visual comparison. The atlas brain for this experiment also has lesion voxels, which are essential for training. The atlas set was:

a_1 : MPRAGE image (3 T, TR = 10.3 ms, TE = 6 ms, $0.82 \times 0.82 \times 1.17 \text{ mm}^3$ voxel

CHAPTER 3. Ψ -CLONE

size),

a_2 : T_2 w image from the second echo of a DSE (3 T, TR = 4177 ms, TE₁ = 3.41 ms, TE₂ = 80 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size),

a_3 : P_D w from the first echo of a DSE (3 T, TR = 4177 ms, TE₁ = 3.41 ms, TE₂ = 80 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size),

a_4 : FLAIR (3 T, TI = 11000 ms, TE = 68 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size)

a_{T_1} : Quantitative T_1 map computed as described in Section 3.2.2,

a_{T_2} : Quantitative T_2 map derived as described in Section 3.2.2,

a_{P_D} : Quantitative P_D map derived as described in Section 3.2.2.

The subject set consisted of:

b_1 : MPRAGE image (3 T, TR = 10.3 ms, TE = 6 ms, $0.82 \times 0.82 \times 1.17$ mm³ voxel size), (Parameters, excepting TR, not provided to the algorithm)

b_2 : P_D w image from the first echo of a DSE (3 T, TR = 4177 ms, TE₁ = 3.41 ms, TE₂ = 80 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size), (Parameters, excepting TR, not provided to the algorithm)

b_3 : T_2 w image from the second echo of a DSE (3 T, TR = 4177 ms, TE₁ = 3.41 ms, TE₂ = 80 ms, $0.82 \times 0.82 \times 2.2$ mm³ voxel size). (Parameters, excepting TR, not provided to the algorithm)

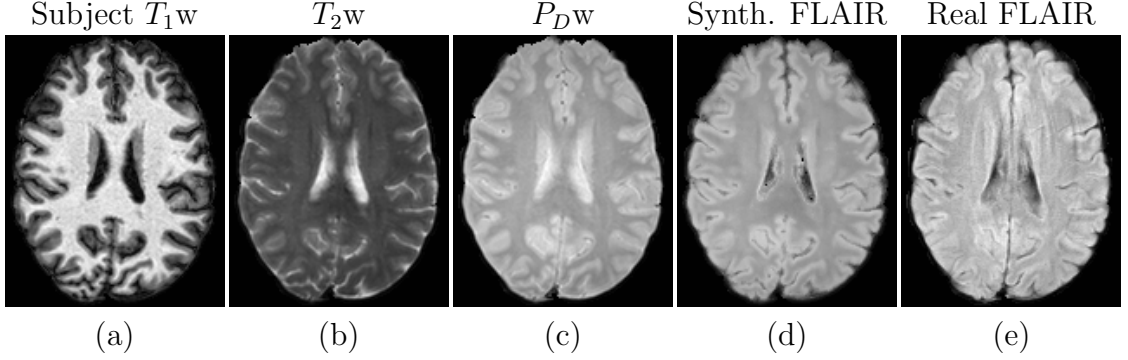


Figure 3.4: (a) Real T_1 w image, (b) real T_2 w image, (c) real P_D w image, (d) synthetic FLAIR, (e) real FLAIR. The real FLAIR shows motion artifacts in (e), which are not present in the synthetic FLAIR (d).

The results of this experiment are shown in Fig. 3.4. The synthetic FLAIR shown in Fig. 3.4(d) does not possess the motion artifacts present in the true FLAIR in Fig. 3.4(e), since these are not present in the input T_1 w, T_2 w, and P_D w images.

Segmentation errors can also stem from misalignment of multimodal images. In the next experiment we demonstrate the potential benefit of using Ψ -CLONE generated synthetic FLAIR images in multimodal analysis. We use the same set of atlas and subject image sets as above, but for a different individual subject in this experiment. The original FLAIR image (Fig. 3.5(e)) for this experiment has a voxel size of $0.82 \times 0.82 \times 4.4 \text{ mm}^3$, which is much larger than the MPRAGE voxel size of $0.82 \times 0.82 \times 1.17 \text{ mm}^3$. Registering the original FLAIR to the MPRAGE requires an upsampling by a factor of four in the through-plane direction. Upsampling via interpolation (trilinear, in this case) results in blurring in the through-plane direction. This blurring is clearly visible in the original FLAIR image in Fig. 3.5(e), especially in the ventricles. It is also visible to some extent in right posterior ventricle region

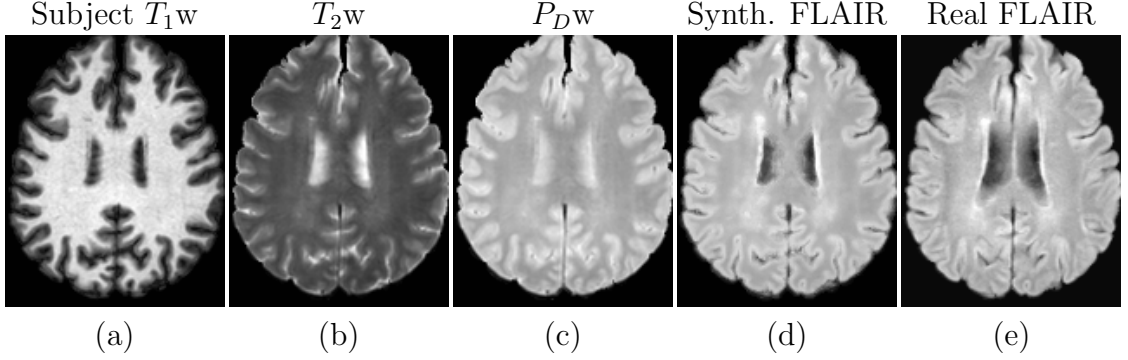


Figure 3.5: (a) Real T_1 w image, (b) real T_2 w image, (c) real P_D w image, (d) synthetic FLAIR, (e) real FLAIR. The real FLAIR shows motion artifacts in (e), which are not present in the synthetic FLAIR (d). The real FLAIR in (e) shows blurring because of interpolation in order to match the high resolution T_1 w.

of Fig 3.4(e). The blurring is also a cause of gross misalignment between the high resolution images and the low resolution FLAIR images. Such misalignment has the potential to adversely affect segmentation results. The synthetic FLAIR image is created by applying Ψ -CLONE on a high resolution T_1 w image (1.17 mm slice thickness) and intermediate resolution P_D w and T_2 w images (2.2 mm slice thickness). Thus, it is better aligned to the high resolution images and is at a higher resolution than the acquired FLAIR image. Fig. 3.5(d) shows the synthetic FLAIR along with the true FLAIR in Fig. 3.5(e). Visually it is apparent that it is better aligned to the rest of the images than the true FLAIR.

3.6 Summary and Discussion

We have proposed an MR image pre-processing framework, Ψ -CLONE, which allows us to perform MR image synthesis and scanner standardization. Ψ -CLONE represents

CHAPTER 3. Ψ -CLONE

a new direction in image synthesis. It works by estimating relevant information about the imaging parameters of a given image and incorporating these into a synthesis that respects MR image formation resulting in more correct synthetic images than previous synthesis methods. Prior work in MR image synthesis has ignored the image acquisition process when solving the synthesis problem. Our use of imaging equation approximations and the underlying NMR tissue parameters to construct atlases that are adaptive to the subject image is a unique feature. In Sections 3.3 and 3.4, we demonstrated the significantly higher quality of image standardization and synthesis than the existing state-of-the-art methods. In addition to this in Section 3.5, we showcased advanced capabilities of our synthesis approach—specifically FLAIR synthesis—which cannot be accomplished by methods other than REPLICA.

Application of Ψ -CLONE for image synthesis can be used to enhance and expand multimodal datasets for better image processing. Improved resolution for modalities like FLAIR and DSE pulse sequences which are often acquired at a low resolution can prove useful for tasks such as segmentation and registration. In addition, the ability to replace an artifact-ridden image with a synthetic one for better and more consistent processing of the entire dataset will help in providing more usable subject data, which should in turn help improve the statistical power of any derived scientific results.

In addition to improving the quality and capabilities of MR image synthesis, Ψ -CLONE is also quite fast, taking less than five minutes to synthesize a new image. In comparison, state-of-the-art methods like MIMECS take around 2–3 hours on the same

CHAPTER 3. Ψ -CLONE

computational resources. This makes Ψ -CLONE well-suited as a quick pre-processing tool or as an MR intensity standardization that can be done on the scanner prior to any other processing. The nonlinear simultaneous equation solver can sometimes lead to local minima in the absence of a good initialization. The robustness of estimation is something we want to work on in the immediate future, by incorporating more accurate models of the pulse sequence equations. This will help improve the pulse sequence parameter estimation. The atlases used in the experiments are obviously of critical importance in the quality of the synthesis that can be performed. As such we are working to acquire high resolution data from a small cohort of subjects on multiple scanners to have a complete picture of the NMR properties at various field strengths as is technically feasible.

Ψ -CLONE has certain limitations which we would like to address in the future. First, it requires a segmentation of the input image(s) to estimate the imaging equation parameters. Specifically, the pulse sequence parameter estimation depends on tissue class means provided by a fuzzy k-means algorithm on T_1 w images. For typical T_1 w sequences like MPRAGE, the fuzzy k-means algorithm is fairly robust in providing the class mean intensities. In the rare case that the algorithm fails, the estimated imaging parameters tend to have very large errors, which results in the formation of inferior synthetic atlas images that can be easily spotted as inaccurate and rectified at the end of Step 2. Some of the earlier image synthesis methods [24, 26, 28] had similar drawbacks that have since been overcome by dictionary selection techniques and use

CHAPTER 3. Ψ -CLONE

of a higher-dimensional space to normalize image patches [4]. Incorporation of these ideas is feasible, but we have yet to explore them. However, unlike MIMECS [4] we do not require a WM peak normalization step.

Second, our pulse sequence parameter estimation method allows us to generate images of high quality and similar characteristics to the ground truth images (see Section 3.3 for experiments). However, the estimated parameters are not identical to the truth for various reasons including the use of theoretical and approximate pulse sequence equations. This is not ideal, as even though the metrics we use to measure the similarity of the images are commonplace; they may have a subtle deficiency that could only be revealed by a far larger and more rigorous study. Our estimation of P_D , T_1 , and T_2 maps also requires (a) 3 different types of pulse sequence images, preferably T_1w , P_Dw , and T_2w images, of an acceptable resolution (the worst we have worked with is $1 \times 1 \times 5 \text{ mm}^3$ voxel-size in experiments described in [63] for a different application), (b) known pulse sequence name (for example SPGR or DSE), and (c) known imaging equation or approximation for each of the three. These can be potentially restrictive in some clinical scenarios and we are working towards relaxing these requirements.

A third deficiency is our chosen regression model (random forests), it is being used because of its expediency and its ability to handle nonlinear intensity transformations. When predicting, random forest regression takes the mean of all the training data that accumulates in a leaf node during model training. In addition, when using the random

CHAPTER 3. Ψ -CLONE

forest the output value from each tree is averaged to give the final prediction from the regression. Clearly all these averages diminish the quality of the results, which can be seen in Fig. 3.4 where the synthetic result appears smoother than the truth. This could be addressed by modifying the random forest to do a linear fit of the data in the leaf nodes or through the use of a different regression approach. Finally, for the next generation of Ψ -CLONE, we intend to replace the random forest regression in Ψ -CLONE by REPLICA. We have investigated the parameter selection for REPLICA extensively and have acquired the necessary knowledge to carry out this replacement.

We noted in Chapter 2 that REPLICA also has its deficiencies. The independent voxel-by-voxel prediction does not explicitly model the dependencies between voxels. The image produced by REPLICA is useful in that it performs admirably as a substitute input for a real image, however, it is not obvious if the resulting synthetic image is “the best” by any well-defined criterion. Next, in Chapter 4, we provide a description of a probabilistic formulation of image synthesis called SynthCRAFT. In SynthCRAFT, we model the conditional distribution of the to-be-synthesized image given the available images. The parameters of this distribution are stored in the leaves of a regression tree that we learn from the training data. Feature design experience and random forest regression in REPLICA and Ψ -CLONE helped us to understand why storing parameters in the leaves of a regression tree is a good idea. SynthCRAFT needs sophisticated parameter learning and inference strategies that we describe further in Chapter 4.

CHAPTER 3. Ψ -CLONE

In summary, we have described a new MR synthesis approach which incorporates principles based on the pulse sequence equations. The framework is validated on synthetic and real data demonstrating its superior synthesis to state-of-the-art approaches. In addition, we have demonstrated the capability to synthesize the FLAIR pulse sequence, which is a noted deficiency of the MIMECS algorithm [4]. Our estimation of pulse sequence parameters to generate a better atlas image could be used by any synthesis approach [4, 24, 31, 36, 48, 82] to help improve results immediately.

Chapter 4

SynthCRAFT: Tree-encoded Conditional Random Fields for Image Synthesis

All image synthesis approaches that are based on an intensity transformation [4, 37, 43], including REPLICA (described in Chapter 2) and Ψ -CLONE (described in Chapter 3), predict a synthetic image, voxel-by-voxel. Individual voxel predictions in the synthetic image are assumed be independent of each other, given the respective feature vectors. However, voxel intensities in brain MRI images exhibit considerable spatial correlation. Neighboring voxels tend to belong to the same tissue and thus have similar intensities. Long range dependencies may also exist in the image as the brain structure is bilaterally symmetric to some extent. These correlations are not

CHAPTER 4. SYNTHCRAFT

modeled by state-of-the-art image synthesis algorithms. MIMECS [4] considers a small patch in an image of contrast \mathcal{C}_1 and reconstructs it with sparse coefficients applied to \mathcal{C}_1 dictionary elements. These same coefficients are then applied to corresponding \mathcal{C}_2 dictionary elements to synthesize a patch of the \mathcal{C}_2 contrast and only the central voxel value of the patch is used to create the synthetic image voxel-by-voxel. Since neighboring patches in the \mathcal{C}_1 contrast image are going to be correlated, it is likely that their sparse reconstructions are also going to be correlated. However, these correlations are not modeled explicitly in MIMECS. Furthermore, long range dependencies are ignored completely. The multi-scale framework and long range context features introduced in REPLICA attempt to include these correlations in the \mathcal{C}_1 contrast features, but the prediction for these features is carried out independently for each voxel; any resulting correlation in the synthesized voxels of the \mathcal{C}_2 contrast image is not modeled explicitly.

MIMECS, REPLICA, and other intensity transformation-based methods produce a synthetic image in a somewhat ad-hoc manner. It is not easily justifiable why a certain synthetic image is the best, aside from using empirical results to prove its utility and distance to the ground truth image. Roy et al. [37] provide a probabilistic, generative model of patch selection and using this framework they can answer the question of why a particular synthetic patch is better than any other patch. They choose the most probable patch as estimated by their model. However, their approach does not answer the question of why an *entire synthetic image* is optimal. In this

chapter, we show how to specify a conditional distribution of the to-be-synthesized subject image given the available subject images. Knowing this distribution enables us to (a) produce a synthetic image that is optimal in the sense of maximizing this probability distribution and (b) evaluate whether any other synthetic image is as “good” as the best we can get.

4.1 Background

Probabilistic modeling spatial dependencies in images has been a fertile area of research for many years in the medical imaging and computer vision community [83]. We focus on modeling spatial interactions through graphical models that define a probability distribution over the entire image. Brain MRI images are usually very large with millions of foreground voxels. This makes modeling all possible spatial dependencies between all voxels computationally intractable. Instead we focus on models that can specify local neighborhood dependencies which can build up together to represent a global distribution defined over an entire image. One of the most popular ways to model images with spatial dependencies has been to use Markov random fields (MRFs).

An MRF is defined over a graph $G = (V, E)$, where V and E are the sets of vertices and edges respectively of G . In an image synthesis context, the set of all voxels i in the image domain form the vertex set V . A pair of voxels $(i, j), i, j \in V$, that are

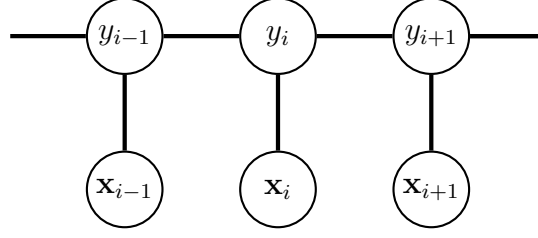


Figure 4.1: A typical hidden MRF. Shown here is the voxel location i with two neighbors, $i - 1$ and $i + 1$. The observed data at these voxels is given by \mathbf{x} . The underlying unknown y_i form the MRF.

neighbors according to a predefined neighborhood form an edge in E . An edge between two vertices indicates dependence between the random variables at those vertices. Let \mathbf{x} be the observed data. In the image synthesis context, \mathbf{x} represents the collection of available images from m pulse sequences, from which we want to synthesize a new image. At each vertex i , we have an observation \mathbf{x}_i , which is a vector of m values. Let $\mathbf{y} = \{y_i\} \ i \in V$ be a continuous-valued random variable over V that represents the synthetic image we want to predict. The posterior distribution is given by $p(\mathbf{y}|\mathbf{x})$. The posterior distribution is important because once we know all its parameters, we can estimate the synthetic image $\hat{\mathbf{y}}$ that has the maximum probability value according to $p(\mathbf{y}|\mathbf{x})$. This estimation is known as maximum a-posteriori (MAP) estimation. It is also called as inference on the posterior distribution, and is the prediction step. The posterior distribution can be expressed using Bayes' rule as follows,

$$p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{y}, \mathbf{x}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y}). \quad (4.1)$$

CHAPTER 4. SYNTHCRAFT

A probabilistic modeling and inference problem can be approached in two ways:

1. Specify the parameters of the joint distribution $p(\mathbf{x}, \mathbf{y})$ and learn them from the training data. Given a test case, use the learned joint distribution to perform MAP estimation on it, since the joint distribution is proportional to the posterior distribution $p(\mathbf{y}|\mathbf{x})$ in Eqn. 4.1. This approach is referred to as *generative* modeling of the data, where both the observations and the hidden variables are jointly modeled.
2. Specify the parameters of the posterior distribution $p(\mathbf{y}|\mathbf{x})$ directly. In this case, we model only the conditional distribution. We learn how to build such a conditional distribution from training data. Given a test case, we can perform a MAP estimate directly on this distribution. This modeling approach is referred to as *discriminative* modeling.

Generative modeling of data involves specifying the details of the joint distribution, which according to Eqn. 4.1 involves specifying two distributions $p(\mathbf{x}|\mathbf{y})$, the likelihood, and $p(\mathbf{y})$, the prior distribution. The distribution $p(\mathbf{y})$ is the prior for the unknown synthetic image and is modeled as an MRF. The Markov assumption can be described as follows,

$$p(y_i|\mathbf{y}_{V \setminus i}) = p(y_i|\mathbf{y}_{\mathcal{N}_i}), \quad (4.2)$$

where $V \setminus i$ denotes the set of all vertices excluding i , and \mathcal{N}_i represents the neighborhood of voxel i . Using the Hammersley Clifford theorem [83], we can express $p(\mathbf{y})$ as

CHAPTER 4. SYNTHCRAFT

a Gibbs distribution,

$$p(\mathbf{y}) = \frac{1}{Z_y} \exp\left(-\frac{1}{T}U(\mathbf{y})\right), \quad (4.3)$$

where Z_y is the normalizing constant called the *partition function*, $U(\mathbf{y})$ is the energy function, and T is a constant that is referred to as the temperature. The energy function $U(\mathbf{y})$ is given by,

$$U(\mathbf{y}) = \sum_{c \in C} V_c(\mathbf{y}), \quad (4.4)$$

where $V_c(\mathbf{y})$ are known as *clique potentials*, defined for each clique c in a set of all cliques C . A *clique* in this graph refers to a subset of vertices in which each vertex is connected to the remaining vertices in the subset. In most MRF-based solutions, two types of cliques are commonly considered. Single vertex cliques and two-vertex or pairwise cliques. In the hidden MRF shown in Fig. 4.1, there are three single vertex cliques, $\{i-1\}$, $\{i\}$, and $\{i+1\}$ respectively. There are two cliques of size two, which are the neighboring voxel pairs $\{i-1, i\}$ and $\{i, i+1\}$. In typical hidden MRF problems, the likelihood term, $p(\mathbf{x}|\mathbf{y})$ in Eq. 4.1, is assumed to be factorizable over the graph vertices i , i.e. $p(\mathbf{x}|\mathbf{y}) = \prod_i p(\mathbf{x}_i|y_i)$. This is visually apparent in Fig. 4.1; where \mathbf{x}_i are assumed to be independent of each other given y_i as there is no edge between them. It is generally a sub-optimal assumption for image synthesis but has been implicitly made by MIMECS and REPLICA. To understand why, let us consider that y_i , the synthetic intensity at voxel i , indicates the tissue type at voxel i . The neighboring voxels of i , \mathcal{N}_i are likely to have observed data $\mathbf{x}_{\mathcal{N}(i)}$ that are similar to \mathbf{x}_i ,

CHAPTER 4. SYNTHCRAFT

since they likely belong to the same tissue and have similar $\mathbf{y}_{\mathcal{N}_i}$ too. Hidden MRF's can model dependence between y_i and $\mathbf{y}_{\mathcal{N}_i}$, but dependencies between the observed data conditioned on y_i are not modeled.

Another limitation of the hidden MRF model is in the specification of the prior $p(\mathbf{y})$. In typical image denoising applications, the prior in Eq. 4.3 is described using pairwise clique potentials [83]. Typical pairwise potentials are designed to ensure smoothness and penalize differences among neighboring voxels. For example, a common pairwise potential used is quadratic, i.e. $U(y_i, y_j) = \lambda(y_i - y_j)^2$. Quadratic potentials tend to oversmooth sharp edges. More robust versions using a truncated quadratic [84, 85] have been proposed to mitigate oversmoothing. However, a more flexible approach would be to employ the observed data \mathbf{x} , and use the observed transitions between the neighbor intensities to modulate the edge penalty in \mathbf{y} in a data-driven manner [86]. It is clear from Fig. 4.1 that the observed data do not affect how the underlying hidden variables y_i interact with each other. Therefore, instead of using a generative hidden MRF to model the image synthesis problem, we propose to use a discriminative conditional random fields-based solution.

To recapitulate, generative modeling entails modeling the joint distribution $p(\mathbf{y}, \mathbf{x})$, which in turn entails modeling the prior $p(\mathbf{y})$ and the data likelihood $p(\mathbf{x}|\mathbf{y})$. As we have discussed, modeling both these distributions requires simplifying assumptions, which end up simplifying the model and making it too restrictive for the application at hand. Instead of modeling the joint distribution, it is sometimes much easier to

CHAPTER 4. SYNTHCRAFT

directly model the posterior distribution $p(\mathbf{y}|\mathbf{x})$, which is what is finally sought during prediction in a MAP-MRF framework. Learning a generative model without the simplifying independence assumptions can be hard because the model itself can be very complex. In contrast the posterior distribution can be relatively simpler to model and learn [87]. Vapnik [88] has argued with regards to solving classification problems that “one should solve the [classification] problem directly and never solve a more general problem as an intermediate step [such as modeling the likelihood $p(\mathbf{x}|\mathbf{y})$]”. Ng et al. [89] compared the solutions to a linear classification problem solved in a discriminative fashion (logistic regression) versus in a generative fashion (naive Bayes), and observed that discriminative learning has a lower asymptotic error, but generative learning might reach its higher error limit, faster. Thus, we propose that if we want to specify distributions of images that model spatial dependencies for the purpose of image synthesis, modeling the posterior distribution directly is a better design choice. The concept of conditional random fields was invented for this very purpose and we describe it next.

4.1.1 Conditional Random Fields

Conditional random fields (CRF) were proposed by Lafferty et al. [90] to directly model the posterior distribution $p(\mathbf{y}|\mathbf{x})$ in the context of text sequence labeling. The posterior $p(\mathbf{y}|\mathbf{x})$ was itself modeled as a Gibbs distribution, in contrast to hidden MRF’s where the prior, $p(\mathbf{y})$ is modeled as a Gibbs distribution. CRF’s that were

CHAPTER 4. SYNTHCRAFT

trained discriminatively for image segmentation were first proposed by Kumar et al. [86], wherein they extended the 1D CRF's of Lafferty et al. [90] to 2D image data and showed improved segmentation.

A CRF is defined as follows: (\mathbf{y}, \mathbf{x}) is a CRF if, conditioned on \mathbf{x} , y_i exhibit the Markov property, i.e.,

$$p(y_i | \mathbf{x}, \mathbf{y}_{V \setminus i}) = p(y_i | \mathbf{x}, \mathbf{y}_{\mathcal{N}_i}), \quad (4.5)$$

where $V \setminus i$ denotes the set of all vertices excluding i and $\mathcal{N}_i = \{j \mid (i, j) \in E\}$, is the neighborhood of i . Assuming $p(\mathbf{y} | \mathbf{x}) > 0, \forall \mathbf{y}$, from the Hammersley-Clifford theorem, we can express this conditional probability as a Gibbs distribution that is factorizable with known clique potentials as follows [83],

$$p(\mathbf{y} | \mathbf{x}) = \frac{1}{Z} \exp[-\{\sum_{i \in V} E_{\mathcal{U}}(y_i, \mathbf{x}; \theta) + \sum_{i \in V} \sum_{j \in \mathcal{N}_i} E_{\mathcal{P}}(y_i, y_j, \mathbf{x}; \theta)\}]. \quad (4.6)$$

Equation 4.6 assumes single vertex clique potentials and pairwise clique potentials. The single vertex cliques at each voxel location $i \in V$ define a *unary potential* $E_{\mathcal{U}}$ or association potential as defined by Kumar et al. [86]. The *pairwise clique potential* $E_{\mathcal{P}}(y_i, y_j, \mathbf{x}; \theta)$ is defined between a vertex $i \in V$ and its neighbor $j \in \mathcal{N}_i$, also referred to as interaction potential by Kumar et al. [86]. Z is the *partition function* and θ is the set of parameters that define both these potentials. Note the presence of \mathbf{x} in the definition of potentials in Eqn. 4.6 versus the lack of \mathbf{x} in the MRF prior (Eqn. 4.3 and Eqn. 4.4).

CHAPTER 4. SYNTHCRAFT

The unary potential $E_{\mathcal{U}}(y_i, \mathbf{x}; \theta)$ can be thought of as a measure of how likely the vertex i is to get a value of y_i , given the entirety of the observed data \mathbf{x} . In this aspect, it is different from a typical hidden MRF (see Fig. 4.1), where the likelihood of a vertex i having the value y_i , is directly dependent on the observed data at i , but not on other vertices. To be more specific, let \mathbf{f} be a function that calculates a feature vector at each voxel i of the observed data \mathbf{x} . We denote $\mathbf{f}_i(\mathbf{x})$ to be the feature vector calculated at vertex i . For instance, it could be the intensities of a small patch surrounding the vertex i or something more complicated like a SIFT descriptor [91]. The major point is that $\mathbf{f}_i(\mathbf{x})$ can use information from observed data \mathbf{x} of all the vertices to calculate a feature vector at the vertex i . In most CRF frameworks [86], $E_{\mathcal{U}}(y_i, \mathbf{x}; \theta)$ is directly modeled as a log-likelihood, i.e.,

$$E_{\mathcal{U}}(y_i, \mathbf{x}; \theta) = \log p(y_i | \mathbf{f}_i(\mathbf{x})). \quad (4.7)$$

Calculating such log-likelihoods can be easily achieved by learning a regression, which predicts y_i given the corresponding feature vector $\mathbf{f}_i(\mathbf{x})$.

The pairwise potential term $E_{\mathcal{P}}(y_i, y_j, \mathbf{x}; \theta)$ controls the interaction between the values y_i and y_j attained by neighboring vertices i and j , given the observed data \mathbf{x} . In smoothing MRF's this term is sometimes set to be proportional to $(y_i - y_j)^2$, which is quadratic [83, 85]. This ensures that neighboring vertices have similar values, thereby ensuring smoothing. A quadratic pairwise potential is known to oversmooth

CHAPTER 4. SYNTHCRAFT

by imposing a high penalty for sharper edges, thus resulting in blurring of the edges in addition to denoising. Robust penalty functions such as the Huber potential, which is proportional to $\min\{(y_i - y_j)^2, \beta^2 + |y_i - y_j - \beta|\}$, where β is a parameter, have been devised to mitigate the oversmoothing. In a CRF however, we can leverage the observed data \mathbf{x} to modulate the penalty. For instance, we can design a pairwise potential that takes into consideration the neighboring feature vectors $\mathbf{f}_i(\mathbf{x})$ and $\mathbf{f}_j(\mathbf{x})$, and compute a quantity $\kappa = \frac{\beta}{\|\mathbf{f}_i(\mathbf{x}) - \mathbf{f}_j(\mathbf{x})\|^2}$. We can then define $E_{\mathcal{P}}(y_i, y_j, \mathbf{x}; \theta) = \kappa(y_i - y_j)^2$. If the neighboring feature vectors are similar, κ is high and the penalty induced therefore is also high, resulting in smoothing. If the neighboring feature vectors are very different, hinting to the existence of an edge, κ is low and the penalty induced therefore is also low, and does not result in oversmoothing the edge.

To summarize, a CRF distribution is commonly specified by defining two main clique potentials: unary and pairwise. The unary potential can be calculated by training a regressor and calculating the log-likelihood. The pairwise potential controls the interaction between values attained by neighboring vertices and can be defined in a data-dependent manner as opposed to typical hidden MRF pairwise potentials. CRFs are more flexible than hidden MRFs and allow for more creative definitions of the unary and pairwise potentials, as befits a task. Next we describe Gaussian CRF's and their application in image synthesis.

4.1.2 Gaussian Conditional Random Fields

If the unary and pairwise potentials $E_{\mathcal{U}}(y_i, \mathbf{x}; \theta)$ and $E_{\mathcal{P}}(y_i, y_j, \mathbf{x}; \theta)$, respectively, in Eqn. 4.6 are quadratic functions of \mathbf{y} , then it is possible to express the conditional distribution $p(\mathbf{y}|\mathbf{x})$ as a multivariate Gaussian,

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \frac{1}{(2\pi)^{\frac{|V|}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}(\mathbf{x}))^T \Sigma(\mathbf{x})^{-1}(\mathbf{y} - \boldsymbol{\mu}(\mathbf{x}))\right) \\ &= \frac{1}{Z} \exp\left(-\frac{1}{2}(\mathbf{y}^T \mathbf{A}(\mathbf{x}) \mathbf{y}) - \mathbf{b}(\mathbf{x})^T \mathbf{y}\right). \end{aligned} \quad (4.8)$$

The parameters $\mathbf{A}(\mathbf{x})$ and $\mathbf{b}(\mathbf{x})$ are dependent on the definitions of unary and pairwise potentials. The mean $\boldsymbol{\mu}(\mathbf{x})$ and the covariance $\Sigma(\mathbf{x})$ are sometimes referred to in literature as *mean parameters* [92]. While performing parameter estimation, it is computationally hard to estimate the mean parameters; hence, the distribution is expressed in terms of $\mathbf{A}(\mathbf{x})$ and $\mathbf{b}(\mathbf{x})$, which are sometimes referred to as the *canonical parameters*.

Any problem that is cast in the framework of probabilistic graphical models needs to the solutions to three sub-problems: (1) parametrization, (2) parameter learning, and (3) inference. Parametrization refers to the specification of parameters that will eventually construct the distribution parameters. In case of Eqn. 4.8, it means defining a procedure to construct the matrices $\mathbf{A}(\mathbf{x})$ and $\mathbf{b}(\mathbf{x})$ given \mathbf{x} . Parameter learning is the capability of efficiently learning those parameters given training data. Again with reference to Eqn. 4.8, parameter learning involves estimating the parameters that

CHAPTER 4. SYNTHCRAFT

make up \mathbf{A} and \mathbf{b} . Inference means the test-time prediction for new data using the parameters learned from the previous two steps. In Eqn. 4.8, this means given test data $\hat{\mathbf{x}}$, calculate $\mathbf{A}(\hat{\mathbf{x}})$ and $\mathbf{b}(\hat{\mathbf{x}})$ and then use these to estimate $\hat{\mathbf{y}}$. In the image synthesis context, the inference task consists of predicting $\hat{\mathbf{y}}$, which is the synthetic image generated from the observed test data $\hat{\mathbf{x}}$. From the parameter learning procedure, we can determine all the parameters of the posterior distribution $p(\mathbf{y}|\hat{\mathbf{x}})$. We can simply choose the $\hat{\mathbf{y}}$ that maximizes the posterior distribution, thus making it a MAP estimation. Since $p(\mathbf{y}|\hat{\mathbf{x}})$ is Gaussian, the MAP estimate is also the mean of the distribution $E(\mathbf{y}|\hat{\mathbf{x}})$. Thus, we can summarize the inference step as,

$$\hat{\mathbf{y}} = \arg \max p(\mathbf{y}|\hat{\mathbf{x}}) = E(\mathbf{y}|\hat{\mathbf{x}}). \quad (4.9)$$

The MAP estimate can be determined by differentiating Eqn. 4.8 with respect to \mathbf{y} and setting the derivative to zero. This gives the following closed form solution for $\hat{\mathbf{y}}$,

$$\hat{\mathbf{y}} = \mathbf{A}(\hat{\mathbf{x}})^{-1}\mathbf{b}(\hat{\mathbf{x}}). \quad (4.10)$$

Previous research on Gaussian Conditional Random Fields (GCRFs) has focused on different ways to solve the three problems of parameterization, parameter learning, and inference. GCRF models were first investigated by Tappen et al. [93] for denoising natural images. In their model, the unknown, denoised image was \mathbf{y} , whereas the observed, noisy image was \mathbf{x} . They expressed $p(\mathbf{y}|\mathbf{x})$ as a Gaussian with weighted

CHAPTER 4. SYNTHCRAFT

quadratic differences between different filtered versions of \mathbf{y} and \mathbf{x} as follows,

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp\left(-\sum_{i=1}^F w_i \mathbf{c}_i * (\mathbf{y} - \mathbf{x})^2\right), \quad (4.11)$$

where \mathbf{c}_i corresponds to a particular derivative filter that is convolved with the images. It is not clear from their method description whether they included a pairwise potential in their framework. Given this parametrization, parameter learning using maximum likelihood was deemed infeasible. The authors instead minimize the quadratic energy (which is the term in the exponent) and ignore the partition function Z . The training time reported for a small amount of training data on a computing cluster was several hours [93]. Given a test image, inference was carried out, by minimizing the quadratic energy in Eqn. 4.11 using a conjugate gradient solver. There are many limitations to this model. In the context of denoising, a quadratic penalty function tends to oversmooth the sharp edges. The authors claim to mitigate this problem with the weights w_i assigned to each of the derivative filters \mathbf{c}_i . The parameter learning is also not optimal since energy-based minimization ignores the effect of parameters present in partition function Z , which needs to be evaluated and the evaluation is often intractable.

Recently, Jancsary et al. [94] came up with *regression tree fields* (RTF) to model GCRFs, by leveraging on certain theoretical insights gleaned from Wainwright et al.'s [92] work on exponential families of graphical models. RTFs parametrize \mathbf{A} and \mathbf{b}

CHAPTER 4. SYNTHCRAFT

by constructing regression trees from training data. For example, the unary potential $E_{\mathcal{U}}(y_i, \mathbf{x}; \theta)$ from Eqn. 4.6 can be defined as follows;

$$E_{\mathcal{U}}(y_i, \mathbf{x}; \theta) = \frac{1}{2}(a_{L(i)}y_i^2) - b_{L(i)}y_i, \quad (4.12)$$

where $\{a_{L(i)}, b_{L(i)}\} \in \theta$ are the parameters defined at the leaf $L(i)$. $L(i)$ is the leaf where the feature vector $\mathbf{f}_i(\mathbf{x})$ extracted for voxel i from the observed data \mathbf{x} , lands after having been passed through successive nodes of a learned regression tree Ψ . The pairwise potentials are also learned in a similar manner. Given a neighborhood configuration, RTFs build a separate tree for each type of neighbors. Thus, if we consider a neighborhood system of 26 neighbors in 3D, 26 trees need to be learned. Each pairwise tree is learned from training data that includes concatenated feature vectors $[\mathbf{f}_i(\mathbf{x}), \mathbf{f}_j(\mathbf{x})]$ from paired neighbor voxels i and j . The pairwise potential for i - j neighbors is defined in RTFs as follows,

$$E_{\mathcal{P}_r}(y_i, y_j, \mathbf{x}; \theta) = \frac{1}{2}(\alpha_{L(ij)_r}y_i^2 + \beta_{L(ij)_r}y_iy_j + \gamma_{L(ij)_r}y_j^2) - \omega_{L(ij)_{1r}}y_i - \omega_{L(ij)_{2r}}y_j, \quad (4.13)$$

where $\{\alpha_{L(ij)_r}, \beta_{L(ij)_r}, \gamma_{L(ij)_r}, \omega_{L(ij)_{1r}}, \omega_{L(ij)_{2r}}\} \in \theta$, $L(ij)$ is the leaf of a regression tree Ψ_r , which is trained using feature vectors $[\mathbf{f}_i(\mathbf{x}), \mathbf{f}_j(\mathbf{x})]$ paired with corresponding dependent values $[y_i, y_j]$, $r \in \{1, \dots, 26\}$ is the neighbor type, and j is the r^{th} neighbor of i . Thus, it is not a typical regression tree that is trained on a 1D output; instead it is a multi-output regression tree. If we consider 26 neighbors, we need to train

CHAPTER 4. SYNTHCRAFT

26 such trees. Given the above parametrization, parameter learning is performed using pseudo-likelihood maximization [95]. Maximum likelihood estimation for the GCRF in Eqn. 4.8 needs to be done under the constraints that \mathbf{A} is positive definite. However, in order to do this, we need to calculate the mean parameters $\boldsymbol{\mu} = \mathbf{A}^{-1}\mathbf{b}$ and $\boldsymbol{\Sigma} = \mathbf{A}^{-1} + \mathbf{b}\mathbf{b}^T$. The complexity of a matrix inverse is $O(|V|^3)$, where $|V| \times |V|$ is the size of \mathbf{A} . $|V|$ corresponds to the number of foreground voxels in an image ($\sim 10^7$) and computing the inverse is not feasible. Thus maximum likelihood estimation is not feasible, which is why the authors [94] carried out pseudo-likelihood maximization. Given test data $\hat{\mathbf{x}}$, the matrices $\mathbf{A}(\hat{\mathbf{x}})$ and $\mathbf{b}(\hat{\mathbf{x}})$ are constructed using the different regression trees and the learned parameters in their leaves. Inference is done by solving the linear system in Eqn. 4.26 using a conjugate gradient descent approach.

Our approach builds upon RTFs in a way that makes it applicable for large amounts of high-dimensional data. Each of the pairwise potential trees in RTFs needs millions of $2 \times d$ -sized feature vectors to train, where d is the length of the feature vector $\mathbf{f}_i(\mathbf{x})$ extracted at each voxel. Instead of encoding the unary and multiple pairwise potentials in different trees, we encode all the potentials in a single tree trained with millions of d -sized feature vectors. In Section 4.2, we describe our parametrization, parameter learning, and inference strategies. Parameter learning for our model is also based on pseudo-likelihood maximization. The inference procedure is similar to the one described in RTFs except, since we only have a single tree, the construction of $\mathbf{A}(\hat{\mathbf{x}})$ and $\mathbf{b}(\hat{\mathbf{x}})$ matrices is much faster and consumes less memory. We refer to our

method as Synthesis with Conditional Random Field Tree or SynthCRAFT.

4.2 Method

4.2.1 Problem Setup

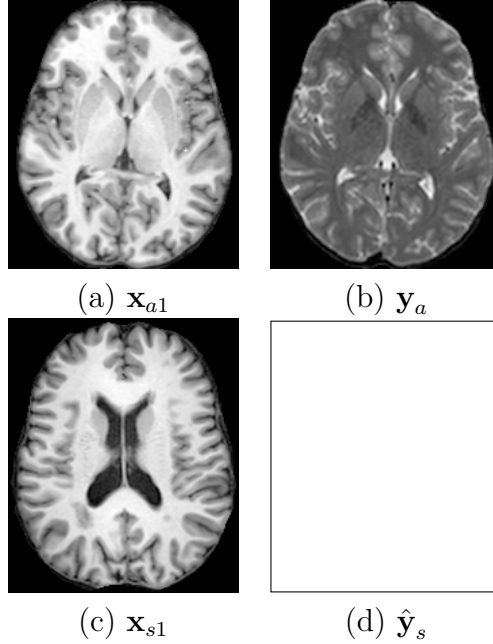


Figure 4.1: A typical synthesis problem setup. In the first row, we show the atlas images \mathbf{x}_a and \mathbf{y}_a , used for training. The second row shows a given test subject image \mathbf{x}_s . The synthesis task involves estimating the unknown image, $\hat{\mathbf{y}}_s$.

Before we describe our approach, we restate the synthesis problem and provide the notation we use. Observed data is denoted by \mathbf{x} . Hidden values are referred to by \mathbf{y} . In the image synthesis scenario, we are given a collection of observed, co-registered, subject images $\mathbf{x}_s = \{\mathbf{x}_{s1}, \mathbf{x}_{s2}, \dots, \mathbf{x}_{sm}\}$, generated by modalities Φ_1, \dots, Φ_m , respectively. The image synthesis task entails predicting the unknown image $\hat{\mathbf{y}}$ of a target modality Φ_t .

CHAPTER 4. SYNTHCRAFT

Given a test image set $\hat{\mathbf{x}}$, our strategy is to determine the conditional distribution $p(\mathbf{y}|\hat{\mathbf{x}})$ and output the MAP estimate $\hat{\mathbf{y}}$ as the expected synthetic image. In Fig. 4.1, we show an image synthesis task of generating T_2w images from input T_1w images. The atlas image collection in Figs. 4.1(a) and 4.1(b) just has two images, $\mathbf{x}_a = \{\mathbf{x}_{a1}\}$ and \mathbf{y}_a . The given subject image is shown Figs. 4.1(c), while the goal is to predict the unknown subject T_2w image denoted by the empty box in Fig. 4.1(d). Next, we describe how to build a regression tree from the training data and encode the CRF parameters in it.

4.2.2 Unary Regression Tree Construction

Given training data comprising images from input and target modalities, as shown in Fig. 4.2, we extract features $\mathbf{f}_i(\mathbf{x}_a)$ from the input modalities and pair them with corresponding target modality intensities y_i , where i is the voxel location. The feature $\mathbf{f}_i(\mathbf{x}_a)$ consists of two parts, 1) a small, local $3 \times 3 \times 3$ -sized patch $\mathbf{p}_i(\mathbf{x}_a)$, and 2) a context feature vector $\mathbf{c}_i(\mathbf{x}_a)$, as described in Chapter 2. A small 3D patch, denoted by $\mathbf{p}_i(\mathbf{x}) = [\mathbf{p}_i(\mathbf{x}_1), \dots, \mathbf{p}_i(\mathbf{x}_m)]$, where the size of the patch is typically $3 \times 3 \times 3$, provides us with local intensity information from all the m input modalities. We construct the context descriptor $\mathbf{c}_i(\mathbf{x})$ as described in Section 2.2.1. The final feature vector is thus $\mathbf{f}_i(\mathbf{x}) = [\mathbf{p}_i(\mathbf{x}), \mathbf{c}_i(\mathbf{x})]$. $\mathbf{f}_i(\mathbf{x})$ is paired with the voxel intensity y_i at i in the target modality image \mathbf{y} to create training data pairs $(\mathbf{f}_i(\mathbf{x}), y_i)$. We train the regression tree Ψ on this training data using the algorithm described by Breiman [42].

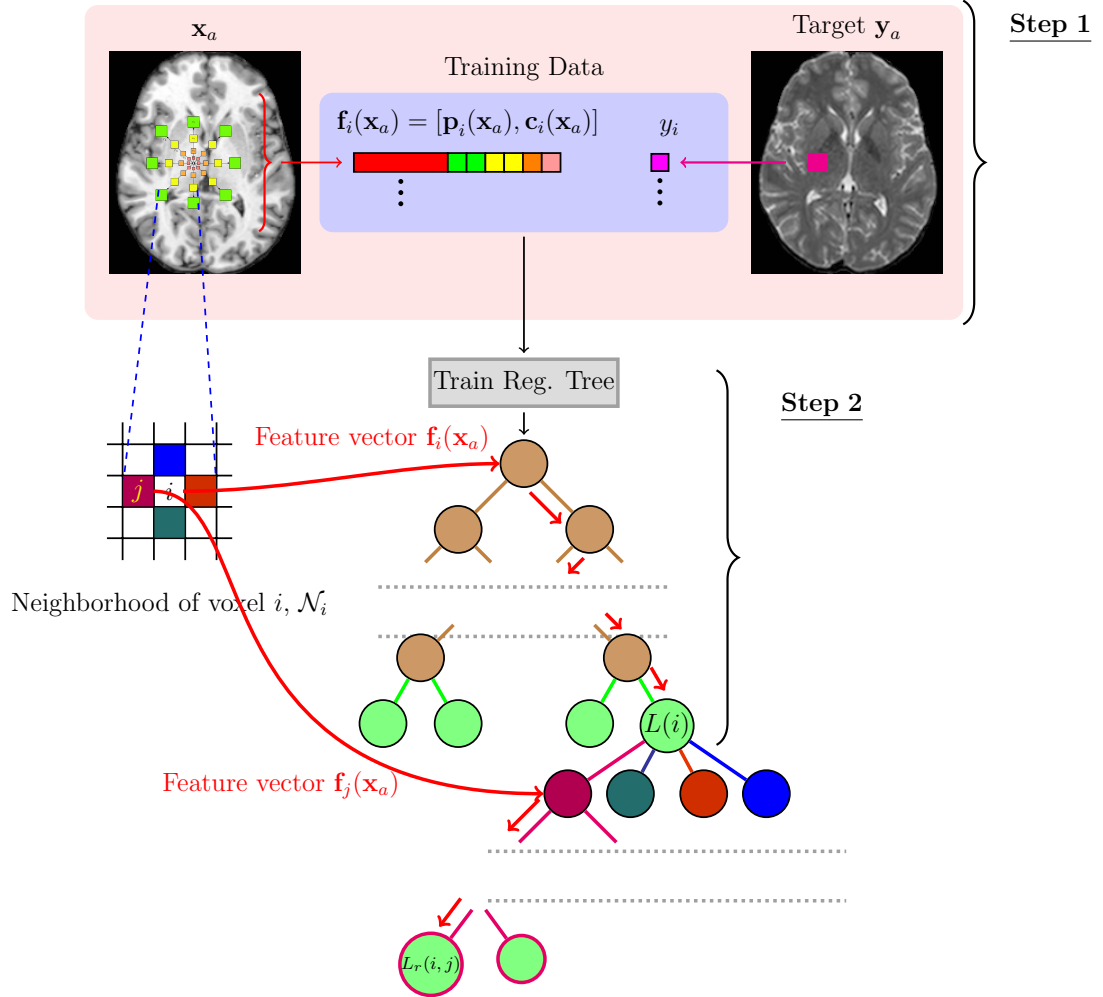


Figure 4.2: Construction and encoding of CRF parameters within a regression tree.

CHAPTER 4. SYNTHCRAFT

Once the tree is constructed, we initialize two parameters a_l and b_l at each of the leaves $l \in \mathcal{L}_\Psi$. These parameters will be used to define the unary potential as we explain in Section 4.2.4.

Algorithm 4 summarizes the unary regression tree construction.

Algorithm 4 SynthCRAFT Training: Learning Unary Regression Tree

- 1: Data: Rigidly co-registered atlas images \mathbf{x}_a and \mathbf{y}_a in the MNI space
 - 2: Extract $3 \times 3 \times 3$ -sized patches from \mathbf{x}_a
 - 3: Extract context descriptors from corresponding voxels
 - 4: Concatenate to create joint feature vector set
 - 5: Extract corresponding voxel intensities from \mathbf{y}_a to complete training data
 - 6: Train a regression tree Ψ on this training data
 - 7: For each leaf node l_u in Ψ , initialize parameters $\{a_{l_u}, b_{l_u}\}$
-

4.2.3 Construction of Pairwise Regression Subtrees

Once we have constructed the unary regression tree, we construct pairwise regression sub-trees at each of the leaves of the unary regression tree. Consider a neighborhood system with $|\mathcal{N}_i|$ number of neighbors. In a typical 3D image, we consider $|\mathcal{N}_i| = 26$ for all voxels i . We depict a simpler scenario in Fig. 4.2. Consider a neighborhood system with 4 neighbors, north, east, south, and west, shown in different colors on the image grid. The unary regression tree results in formation of leaf nodes, shown in green in Fig. 4.2. Let the feature vector $\mathbf{f}_i(\mathbf{x}_a)$ fall in a leaf node, which we denote by $L(i)$. Note that feature vectors calculated at many other voxels might also end up in $L(i)$. Next we consider all the same type of neighboring voxels j of the voxels i whose feature vectors ended up at $L(i)$. A neighbor type $r \in \{1, \dots, |\mathcal{N}_i|\}$ for a voxel

CHAPTER 4. SYNTHCRAFT

i implies one of its n ($= |\mathcal{N}_i| = |\mathcal{N}|$) neighbors. For example, a neighborhood system with four neighbors (north, south, west, east) has four types of neighbors, and hence four types of edges. In Fig. 4.2, we consider the western neighbor j (in magenta) of voxel i . We accumulate feature vectors $\mathbf{f}_j(\mathbf{x}_a)$ of all j , where the feature vector $\mathbf{f}_i(\mathbf{x}_a)$ has landed in the leaf $L(i)$. These $\mathbf{f}_j(\mathbf{x}_a)$ are paired with the target modality value y_j . Using these as the training data, we train a regression tree $\Psi_r(L(i))$ (at leaf $L(i)$, for each type of neighbor r). Thus with four neighbors, we have four regression sub-trees at $L(i)$ in Fig. 4.2. In a 3D scenario, with 26 neighbors we would have 26 sub-trees. These sub-trees are designed to be ‘shallow’, which means their depth is limited to be 3–5. They are also very quick to train as the number of training $\mathbf{f}_j(\mathbf{x}_a)$ are equal to the number of $\mathbf{f}_i(\mathbf{x}_a)$ at $L(i)$ and these can be specified via a parameter while constructing the unary regression tree. After constructing the pairwise sub-trees for all types of neighbors, we note the leaf $L_r(i, j)$, where the feature vector of neighbor j of type r , $\mathbf{f}_j(\mathbf{x}_a)$ ends up. At $L_r(i, j)$, we initialize the parameters which will help in defining the pairwise potential, namely, $\alpha_{L_r(i, j)}$, $\beta_{L_r(i, j)}$, $\gamma_{L_r(i, j)}$, $\omega_{L_{r1}(i, j)}$, and $\omega_{L_{r2}(i, j)}$.

Algorithm 5 summarizes the procedure for constructing pairwise regression sub-trees.

Algorithm 5 SynthCRAFT Training: Learning Pairwise Regression Sub-Trees

```

1: Data: Unary regression tree  $\Psi$  learned from atlas images
2: for neighbor type  $r = 1:|\mathcal{N}|$  do
3:   for each leaf node  $l_u \in \Psi$  do
4:     Determine the voxel locations  $i$  of all the training samples in  $l_u$ 
5:     Extract feature vectors  $\mathbf{f}_j(\mathbf{x}_a)$  from  $j$ , where  $j$  the  $r^{th}$  neighbor of  $i$ 
6:     Train a shallow tree  $\Psi_r(l_u)$  with all extracted  $\mathbf{f}_j(\mathbf{x}_a)$ 
7:     for each leaf node  $l_p \in \Psi_r(l_u)$  do
8:       Initialize parameters  $\{\alpha_{l_p}, \beta_{l_p}, \gamma_{l_p}, \omega_{l_p1}, \text{ and } \omega_{l_p2}\}$ 
9:     end for
10:   end for
11: end for

```

4.2.4 Parametrization

Using the unary regression tree and its pairwise regression sub-trees we can now define the unary and pairwise potentials. The unary potential is defined as follows.

$$E_{\mathcal{U}}(y_i, \mathbf{x}; \theta) = \frac{1}{2}(a_{L(i)}y_i^2) - b_{L(i)}y_i, \quad (4.14)$$

where $\{a_{L(i)}, b_{L(i)}\} \in \theta$ are the parameters defined at the leaf $L(i)$. We can compare Eqn. 4.14 with the generic definition of a unary potential in Eqn. 4.12. The right hand side term $p(y_i|\mathbf{f}_i(\mathbf{x}))$ in Eqn. 4.12 corresponds to the quadratic term in the right hand side of Eqn. 4.14. Thus we are modeling the distribution of y_i 's in the leaf $L(i)$ as a Gaussian distribution with the parameters $a_{L(i)}$ and $b_{L(i)}$. This is a reasonable model for the samples in the leaves of a regression tree as the splitting process in a regression tree tries to ensure that the sample distribution variance inside a node reduces as we go deeper in the tree. If the features are sufficiently disambiguating,

CHAPTER 4. SYNTHCRAFT

then the distribution of samples in the leaf nodes is close to being a mono-modal Gaussian and hence can be modeled as such.

The pairwise potential for neighbors of type r is also defined in a quadratic form as follows,

$$E_{\mathcal{P}_r}(y_i, y_j, \mathbf{x}; \theta) = \frac{1}{2}(\alpha_{L_r(i,j)}y_i^2 + \beta_{L_r(i,j)}y_iy_j + \gamma_{L_r(i,j)}y_j^2) - \omega_{L_{r1}(i,j)}y_i - \omega_{L_{r2}(i,j)}y_j, \quad (4.15)$$

where $\{\alpha_{L_r(i,j)}, \beta_{L_r(i,j)}, \gamma_{L_r(i,j)}, \omega_{L_{r1}(i,j)}, \omega_{L_{r2}(i,j)}\} \in \theta$ are the parameters defined at the leaf $L_r(i, j)$. The pairwise potential usually acts as a smoothing term, but can also be designed in a more general manner. For instance, $E_{\mathcal{P}}(y_i, y_j, \mathbf{x}; \theta) = (y_i - y_j)^2$ would imply that neighboring intensities y_i and y_j will be forced to be close to each other when estimating \mathbf{y} that maximizes the posterior distribution $p(\mathbf{y}|\mathbf{x})$. Thus, if there is an edge present at that location in the input \mathbf{x} , it will be oversmoothed in the synthetic result $\hat{\mathbf{y}}$, which is undesirable. In our model, the parameters of this quadratic potential are dependent on the leaves $L_r(i, j)$ where $\mathbf{f}_j(\mathbf{x}_a)$ falls, given the leaf $L(i)$ where $\mathbf{f}_i(\mathbf{x}_a)$ has fallen. Neighboring feature vectors $\mathbf{f}_j(\mathbf{x}_a)$ that are similar to $\mathbf{f}_i(\mathbf{x}_a)$ will tend to fall in a certain pairwise sub-tree leaf while those that differ by a lot will tend to fall in a different pairwise sub-tree leaf. Thus, if there is an edge in the input data \mathbf{x} , the parameters for the quadratic potential will be different compared to when there is no edge, and hence will not result in oversmoothing. We define pairwise potentials for each type of neighbor.

CHAPTER 4. SYNTHCRAFT

To summarize, for each leaf l_u in the unary regression tree, we have $\{a_{l_u}, b_{l_u}\}$ that belong to the parameter set θ . For each pairwise regression sub-tree leaf l_p , we have $\{\alpha_{l_p}, \beta_{l_p}, \gamma_{l_p}, \omega_{l_{p1}}, \omega_{l_{p2}}\}$ that also belong to θ . These account for all the parameters that can be used to generate a GCRF distribution. Next, we describe a parameter learning procedure that estimates these parameters in all leaves using a maximum pseudo-likelihood procedure.

4.2.5 Parameter Learning

An ideal approach to learn parameters would be to perform maximum likelihood estimation using the distribution in Eq. 4.8. However as mentioned in [94], estimation of the mean parameters Σ and μ , requires calculation of \mathbf{A}^{-1} (see Eq. 4.8). The size of \mathbf{A} is $|V| \times |V|$ where $|V|$ is the number of voxels in \mathbf{y} and for large 3D images, $|V|$ is of the order of $\sim 10^6$, which makes the computation practically infeasible. We follow [94] and implement a pseudo-likelihood maximization-based parameter learning.

Pseudo-likelihood is defined as the product of local conditional likelihoods,

$$\hat{\theta}_{\text{MPLE}} = \operatorname{argmax}_{\theta} \prod_{i \in V} p(y_i \mid \mathbf{y}_{\mathcal{N}_i}, \mathbf{x}; \theta). \quad (4.16)$$

CHAPTER 4. SYNTHCRAFT

The local conditional likelihood can be expanded as

$$\begin{aligned} p(y_i | \mathbf{y}_{\mathcal{N}_i}, \mathbf{x}; \theta) &= \frac{p(y_i, \mathbf{y}_{\mathcal{N}_i}, \mathbf{x}; \theta)}{\int_{\mathbb{R}} p(y_i, \mathbf{y}_{\mathcal{N}_i}, \mathbf{x}; \theta) dy_i}, \\ -\log p(y_i | \mathbf{y}_{\mathcal{N}_i}, \mathbf{x}; \theta) &= -\log p(y_i, \mathbf{y}_{\mathcal{N}_i}, \mathbf{x}; \theta) + \log Z_i, \end{aligned} \quad (4.17)$$

where $Z_i = \int_{\mathbb{R}} p(y_i, \mathbf{y}_{\mathcal{N}_i}, \mathbf{x}; \theta) dy_i$. Using the CRF definition in Eq. 4.6, we can write $-\log p(y_i, \mathbf{y}_{\mathcal{N}_i}, \mathbf{x}; \theta)$ as

$$\begin{aligned} -\log p(y_i, \mathbf{y}_{\mathcal{N}_i}, \mathbf{x}; \theta) &= E_{\mathcal{U}}(y_i, \mathbf{x}; \theta) + \sum_{j \in \mathcal{N}_i} E_{\mathcal{P}}(y_i, y_j, \mathbf{x}; \theta) \\ &= \frac{1}{2} a_{Ci} y_i^2 - b_{Ci} y_i, \end{aligned} \quad (4.18)$$

where we can find a_{Ci} (Eq. 4.19) and b_{Ci} (Eq. 4.20) by matching quadratic and linear terms. Equations 4.19 and 4.20 show the contribution of interaction potentials induced by the neighbors of voxel i . The \tilde{r} denotes the type of edge which is symmetric to type r . For example, if edges of type r are between voxel i and its right neighbor, then \tilde{r} denotes the type that is between a voxel and its left neighbor.

$$a_{Ci} = a_{L(i)} + \left(\sum_{j | (i,j) \in E_r} \alpha_{L_r(i,j)} + \sum_{h | (h,i) \in E_{\tilde{r}}} \gamma_{L_{\tilde{r}}(h,i)} \right) \quad (4.19)$$

CHAPTER 4. SYNTHCRAFT

$$\begin{aligned}
b_{Ci} = & b_{L(i)} + \left(\sum_{j|(i,j) \in E_r} \omega_{L_{r1}(i,j)} + \sum_{h|(h,i) \in E_{\tilde{r}}} \omega_{L_{\tilde{r}2}(h,i)} \right. \\
& \left. - \frac{1}{2} \sum_{j|(i,j) \in E_r} \beta_{L_r(i,j)} y_j - \frac{1}{2} \sum_{h|(h,i) \in E_{\tilde{r}}} \beta_{L_{\tilde{r}}(h,i)} y_h \right).
\end{aligned} \tag{4.20}$$

The integral of exponential terms Z_i in Eq. 4.17, is also known as the log partition term. To optimize objective functions with log partition terms, we express Z_i in its variational representation using the mean parameters $\boldsymbol{\mu}_i = [\mu_i, \sigma_i]$ [92]. The parameter set $\boldsymbol{\theta}_{Ci} = \{b_{Ci}, a_{Ci}\}$ that defines the exponential distribution is known as the canonical parameter set. The conjugate dual function of Z_i is defined as follows,

$$Z_i^*(\mu_i, \sigma_i) = \sup_{\boldsymbol{\theta}_{Ci}} \langle \boldsymbol{\theta}_{Ci}, \boldsymbol{\mu}_i \rangle - Z_i(\boldsymbol{\theta}_{Ci}), \tag{4.21}$$

where $\langle \rangle$ denotes inner product. Substituting $\boldsymbol{\theta}_{Ci}$ and the expression for $-\log p(y_i, \mathbf{y}_{\mathcal{N}_i}, \mathbf{x}; \boldsymbol{\theta})$ from Eq. 4.18, we get the negative pseudo-likelihood (NPL) contributed by voxel i to be,

$$\text{NPL}_i(\boldsymbol{\theta}) = b_{Ci}(\mu_i - y_i) + \frac{1}{2} a_{Ci}(y_i^2 - \sigma_i) + \log(\sigma_i - \mu_i^2) + \log(2\pi e), \tag{4.22}$$

where the mean parameters are given by $\mu_i = \frac{b_{Ci}}{a_{Ci}}$ and $\sigma_i = \frac{1}{a_{Ci}} + \mu_i^2$.

Equation 4.22 is similar to the one in [94], as the overall model is a Gaussian CRF. The objective function is linearly related to $\boldsymbol{\theta}$ and has convex constraints [92, 94].

CHAPTER 4. SYNTHCRAFT

We minimize $\sum_{i \in V} \text{NPL}_i(\theta)$ using constrained, projected gradient descent on the parameters to ensure positive definiteness of the final precision matrix $\mathbf{A}(\mathbf{x})$ for any input \mathbf{x} , similar to the regression tree fields concept [94]. Training in our experiments takes about 20–30 minutes with $\sim 10^6$ samples of dimensionality of the order of $\sim 10^2$ and neighborhood size of 26, on a 12 core 3.42 GHz machine. Algorithm 6 describes the parameter learning process.

Algorithm 6 SynthCRAFT Training: Parameter Learning

- 1: Data: Unary regression tree Ψ and pairwise regression sub-trees $\Psi_r(l_u)$ for each leaf node $l_u \in \Psi$ and each neighbor type $r \in \{1, \dots, |\mathcal{N}|\}$
 - 2: Initialize all the parameters present in the leaf nodes of the unary regression tree and the pairwise regression sub-trees
 - 3: Minimize the objective function in the form of negative pseudo-likelihood from Eqn. 4.22 using projected gradient descent until convergence or 100 iterations, whichever is earlier
-

4.2.6 Inference

Given a test image set $\hat{\mathbf{x}} = \{\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_m\}$, which are co-registered, we first extract features $\mathbf{f}_i(\hat{\mathbf{x}})$ from all voxel locations i . Next, we apply the learned regression tree Ψ to each of $\mathbf{f}_i(\hat{\mathbf{x}})$ to determine the leaf node $L(i)$ in Ψ . Knowledge of $L(i)$ tells us the values of the unary potential parameters $a_{L(i)}$ and $b_{L(i)}$ that we need to use while constructing the entries of $\mathbf{A}(\hat{\mathbf{x}})$. Next, for each voxel i , we evaluate the feature vectors $\mathbf{f}_j(\hat{\mathbf{x}})$ at each neighbor $j \in \mathcal{N}_i$. For a neighboring voxel j of type r , we pass the feature vector $\mathbf{f}_j(\mathbf{x})$ down the shallow, pairwise regression tree $\Psi_r(L(i))$ for neighbor type r in the leaf $L(i)$. This gives us the leaf location $L_r(i, j)$ where the neighboring

CHAPTER 4. SYNTHCRAFT

feature vector lands, and hence, the values of all the pairwise potential parameters;

$$\{\alpha_{L_r(i,j)}, \beta_{L_r(i,j)}, \gamma_{L_r(i,j)}, \omega_{L_{r1}(i,j)}, \omega_{L_{r2}(i,j)}\}.$$

Using these, we can now construct the $\mathbf{A}(\hat{\mathbf{x}})$ matrix element-by-element by matching the quadratic terms of Eq. 4.8. The diagonal terms $\mathbf{A}_{ii}(\hat{\mathbf{x}})$ consist of coefficients of y_i^2 terms in the quadratic expansion, which can be obtained by matching the coefficients of y_i^2 coefficients in our unary and pairwise potentials, as shown in Eqn. 4.23.

$$\mathbf{A}(\hat{\mathbf{x}})_{ii} = a_{L(i)} + \sum_{(i,j)|(i,j) \in E_r} \alpha_{L_r(i,j)} + \sum_{(h,i)|(h,i) \in E_{\bar{r}}} \gamma_{L_{\bar{r}}(h,i)}. \quad (4.23)$$

The off-diagonal terms correspond to the coefficients of the $y_i y_j$ terms Eqn. 4.8.

These can be obtained as follows,

$$\mathbf{A}(\hat{\mathbf{x}})_{ij} = \frac{1}{2} \left(\sum_{(i,j)|(i,j) \in E_r} \beta_{L_r(i,j)} + \sum_{(h,i)|(h,i) \in E_{\bar{r}}} \beta_{L_{\bar{r}}(h,i)} \right). \quad (4.24)$$

Finally, the $\mathbf{b}(\hat{\mathbf{x}})$ vector is generated by matching the linear terms from Eqn. 4.8 as follows,

$$\mathbf{b}(\hat{\mathbf{x}})_i = b_{L(i)} + \sum_{(i,j)|(i,j) \in E_r} \omega_{L_{r1}(i,j)} + \sum_{(h,i)|(h,i) \in E_{\bar{r}}} \omega_{L_{\bar{r}2}(h,i)}. \quad (4.25)$$

The MAP estimate for $p(\mathbf{y}|\hat{\mathbf{x}})$ as well as the conditional expectation $E[\mathbf{y}|\hat{\mathbf{x}}]$ is the mean of the multivariate Gaussian described in Eq. 4.8. The expression for the mean

CHAPTER 4. SYNTHCRAFT

and hence the estimate $\hat{\mathbf{y}}$ is given by,

$$\hat{\mathbf{y}} = \mathbf{A}(\hat{\mathbf{x}})^{-1}\mathbf{b}(\hat{\mathbf{x}}). \quad (4.26)$$

$\mathbf{A}(\hat{\mathbf{x}})$ is a large ($\sim 10^6 \times 10^6$), sparse ($\sim 27 \times 10^6$ non-zero entries), symmetric positive definite matrix. Thus, we use an iterative preconditioned conjugate gradient descent method to solve the linear system in Eq. 4.26. The estimate $\hat{\mathbf{y}}$ is our synthetic image.

The inference process is summarized in Algorithm 7.

Algorithm 7 SynthCRAFT Prediction: Inference

- 1: Data: Subject image, $\hat{\mathbf{x}}$, unary regression tree Ψ and pairwise regression sub-trees $\Psi_r(l_u)$ for each leaf node $l_u \in \Psi$ and each neighbor type $r \in \{1, \dots, |\mathcal{N}|\}$ with learned parameters
 - 2: Extract feature vectors $\mathbf{f}_i(\hat{\mathbf{x}})$ from subject image $\hat{\mathbf{x}}$
 - 3: Apply unary regression tree Ψ to $\mathbf{f}_i(\hat{\mathbf{x}})$ and for all voxels i , determine the leaf nodes $L(i)$ where $\mathbf{f}_i(\hat{\mathbf{x}})$ land
 - 4: **for** each leaf l_u in Ψ **do**
 - 5: Determine all voxels i that have feature vectors $\mathbf{f}_i(\hat{\mathbf{x}})$ that have landed at l_u
 - 6: **for** $r = 1:|\mathcal{N}|$ **do**
 - 7: Let j be the r^{th} neighbor, extract $\mathbf{f}_j(\hat{\mathbf{x}})$ and apply the corresponding pairwise sub-tree $\Psi_r(l_u)$ to determine the pairwise sub-tree leaf nodes $L_r(i, j)$ for all j
 - 8: **end for**
 - 9: **end for**
 - 10: **for** voxels i in $\hat{\mathbf{x}}$ **do**
 - 11: Construct diagonal entries of $\mathbf{A}(\hat{\mathbf{x}})$ as per Eqn. 4.23
 - 12: **for** $r = 1:|\mathcal{N}|$ **do**
 - 13: Let j be the r^{th} neighbor, construct the off-diagonal entries of $\mathbf{A}(\hat{\mathbf{x}})$ as per Eqn. 4.24
 - 14: Construct entries of $\mathbf{b}(\hat{\mathbf{x}})$ as per Eqn. 4.25
 - 15: **end for**
 - 16: **end for**
 - 17: Estimate $\hat{\mathbf{y}} = \mathbf{A}(\hat{\mathbf{x}})^{-1}\mathbf{b}(\hat{\mathbf{x}})$ using conjugate gradient algorithm
-

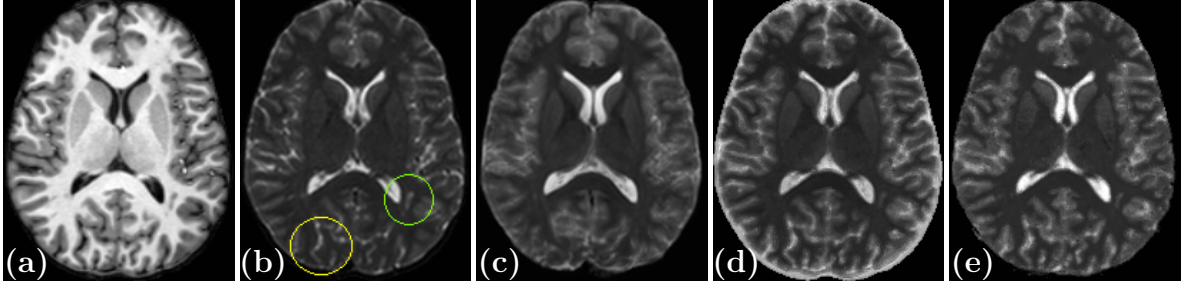


Figure 4.1: Shown are (a) the input MPRAGE image, (b) the true T_2w image, and the synthesis results from the MPRAGE for each of (c) FUSION, (d) MIMECS, and (e) SynthCRAFT (our method). The lesion (green circle) and the cortex (yellow circle) in the true image are synthesized by MIMECS and SynthCRAFT, but not by FUSION.

4.3 Results

4.3.1 Synthesis of T_2w Images from T_1w Images

In this experiment, we used MPRAGE images from the publicly available Multimodal MRI Reproducibility Resource (MMRR) data [50] and synthesized the T_2w images of the DSE sequence. The multimodal reproducibility data consists of 21 subjects, each with two imaging sessions, acquired within an hour of each other. Thus there are 42 MPRAGE images. We excluded data of five subjects (ten images), which were used for training and synthesized the remaining 32. We compared SynthCRAFT to MIMECS [4] and multi-atlas registration and intensity fusion (FUSION) [33]. We used five subjects as the atlases for FUSION with the parameters $\beta = 0.5$ and $\kappa = 4$ (fuse the four best patch matches).

We used PSNR (peak signal to noise ratio), universal quality index (UQI) [53], and structural similarity (SSIM) [54] as metrics. UQI and SSIM take into account image

CHAPTER 4. SYNTHCRAFT

Table 4.1: Mean and standard deviation (Std. Dev.) of the PSNR, UQI, and SSIM values for synthesis of T_2w images from 32 MPRAGE scans.

	PSNR	UQI	SSIM
	Mean (Std)	Mean (Std.)	Mean (Std)
FUSION	52.73 (2.78)*	0.78 (0.02)	0.82 (0.02)
MIMECS	36.13 (2.23)	0.78 (0.02)	0.77 (0.02)
REPLICA	50.73 (2.67)	0.89 (0.02)*	0.87 (0.02)*
SynthCRAFT	49.73 (1.99)	0.86 (0.01)*	0.84 (0.01)*

* Statistically significantly better than FUSION and MIMECS methods (α level of 0.01) using a right-tailed test.

degradation as observed by a human visual system. Both have values that lie between 0 and 1, with 1 implying that the images are equal to each other. SynthCRAFT performs significantly better than FUSION and MIMECS for all metrics except PSNR. Figure 4.1 shows the results for all three methods along with the true T_2w image. FUSION results (Fig. 4.1(b)) have the highest PSNR, but produce anatomically incorrect images, especially in the presence of abnormal tissue anatomy (lesions for example) and the cortex. Overall, SynthCRAFT produces an image that is visually closest to the true T_2w image. SynthCRAFT is slightly worse than REPLICA, but with better parameter selection it can be improved.

4.3.2 Synthesis for FLAIR Images

In this experiment, given atlas P_{Dw} , T_2w , T_1w , and FLAIR images, we trained SynthCRAFT and applied it to subject P_{Dw} , P_{Dw} , and T_1w images, to predict the subject synthetic FLAIR image. We used our in-house multiple sclerosis (MS) patient image dataset with 49 subject images, with four training subjects and testing on the

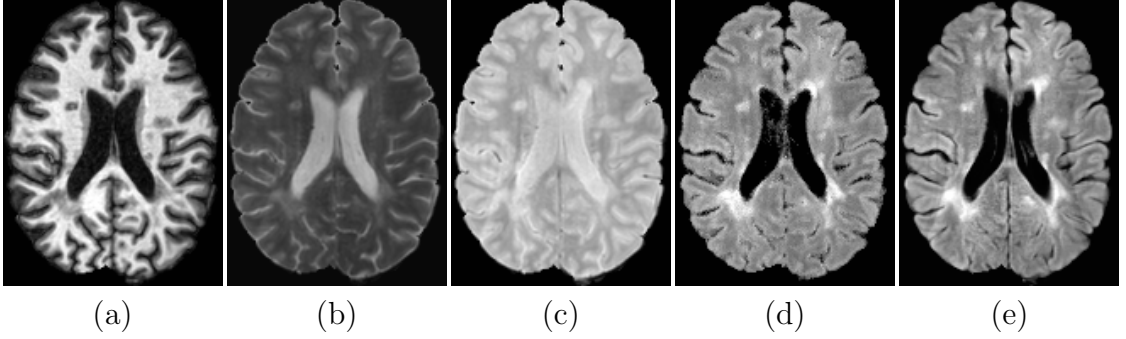


Figure 4.2: (a) Subject T_1w , (b) T_2w , (c) P_Dw images, with (d) SynthCRAFT FLAIR, and (f) Subject Real FLAIR.

remaining 45. We computed average PSNR (20.81, std = 1.19), UQI (0.81, std = 0.03) and SSIM (0.78, std = 0.03), over these 45 subjects. These values indicate that the synthetic FLAIRs are structurally and visually similar to their corresponding real FLAIRs. Figure 4.2 shows the input images and the synthetic FLAIR image along side the real FLAIR image.

Next, we investigated the segmentations acquired from these synthetic FLAIRs. We would like the segmentation algorithm, LesionTOADS [55], to behave identically for real and synthetic images. LesionTOADS uses a T_1w and a corresponding FLAIR to generate a multi-class, topologically correct segmentation in the presence of lesions. We compared the overlap of segmentations obtained using synthetic FLAIRs to those obtained using real FLAIRs in terms of Dice coefficients (averaged over 45 subjects) for white matter (0.97, std = 0.01) (WM), gray matter (0.99, std = 0.01) (GM), cerebrospinal fluid (0.97, std = 0.01) (CSF) and white matter lesions (0.52, std = 0.17) (WML).

Figure 4.3 shows the segmentations by LesionTOADS on real and synthetic FLAIRs.

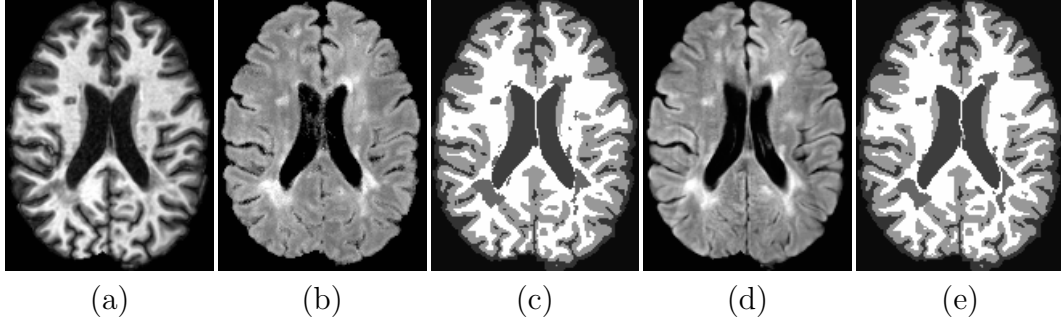


Figure 4.3: (a) Subject T_1w , (b) SynthCRAFT FLAIR, (c) LesionTOADS segmentation of T_1w + SynthCRAFT FLAIR, (d) Subject real FLAIR, and (e) LesionTOADS segmentation of T_1w + real FLAIR.

The overlap is very good for WM, GM, and CSF, however it is low for the WML class. The lesions being small and diffuse, even a small difference in the overlap can cause a low value for the Dice coefficient. So, we looked at the overall lesion volumes as provided by the algorithm for real and synthetic FLAIRs. To understand how different the lesion volumes are for the synthetic images as compared to the real images, we created a Bland-Altman [59] plot shown in Fig. 4.4. Let $RFlv$ be the lesion volumes given by LesionTOADS using real FLAIRs as input. Let $SFlv$ be the lesion volumes using synthetic FLAIRs as input. Bland-Altman plot is a scatter plot of $(RFlv - SFlv)$ (y -axis) versus $(RFlv + SFlv)/2$ (x -axis). The measurements are considered to be interchangeable if 0 lies within $\pm 1.96\sigma$ where σ is the standard deviation of $(RFlv - SFlv)/2$. There is a small bias between $RFlv$ and $SFlv$ (mean = 0.88×10^3) however, 0 does lie between the prescribed limits and hence based on this plot we can say that these two measurements are interchangeable.

Thus, in this experiment we have shown that SynthCRAFT is also capable of

CHAPTER 4. SYNTHCRAFT

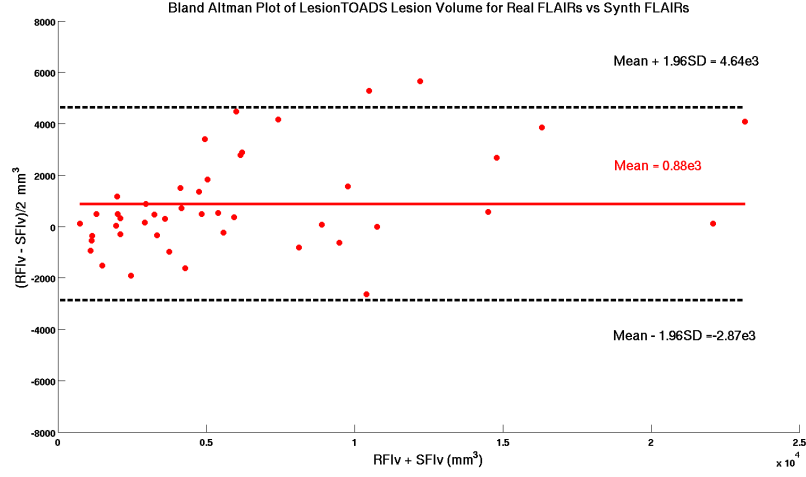


Figure 4.4: A Bland-Altman plot of lesion volumes for synthetic FLAIRs vs real FLAIRs.

FLAIR synthesis, like REPLICA and Ψ -CLONE. The lesion segmentation results are very similar to REPLICA, even though the quality of images is not as good.

4.3.3 Intensity Standardization

Next, we perform an intensity standardization experiment, similar to the one we performed for REPLICA as described in Section 2.3.4. We wish to demonstrate that SynthCRAFT has the same capabilities as REPLICA with regards to intensity standardization. We use the same subset of the BLSA dataset [64] that we used in the REPLICA intensity standardization experiment (Section 2.3.4). This subset consists of 82 scans of 60 subjects, some of which are longitudinal acquisitions. Each scanning session was carried out on a Philips 1.5 T scanner and has an SPGR (see Fig. 2.12(a)) and an MPRAGE (see Fig. 2.12(b)) acquisition from the same session.

CHAPTER 4. SYNTHCRAFT

The goal of this experiment is to show that a segmentation algorithm provides different results for SPGR and MPRAGE, but using synthetic MPRAGE images generated from SPGR images can reduce the difference in the segmentations. We use an in-house implementation of the probabilistic atlas-driven, EM-base segmentation algorithm by [61], which we refer to as AtlasEM. AtlasEM provides us with a 4-class segmentation, the classes being sulcal CSF, GM, WM, and ventricles. In an ideal scenario, where the algorithm is impartial to the underlying T_1w input, the segmentations should be identical. We calculate the tissue volumes provided by AtlasEM on all three sets of images, SPGR, MPRAGE, and synthetic MPRAGE. As stated earlier, if the AtlasEM algorithm were robust to the input modality, the tissue volumes for a particular tissue should be identical for SPGR and MPRAGE. In Fig. 4.5 (a)–(d), we show scatter plots for the tissue volumes obtained on real MPRAGEs (x -axis) and those obtained for SPGRs (blue) and synthetic MPRAGEs (red). We also show the least square line fits to the scatter plots (blue for SPGR, red for synthetic MPRAGE). In the ideal scenario, the least square line fits should be close to the $x = y$ line. We can see that for sulcal CSF (Fig. 4.5(a)), GM (Fig. 4.5(c)), and WM (Fig. 4.5(d)), synthetic MPRAGE is closer to the identity line than SPGR. The ventricle volumes are more or less similar in both the modalities (Fig. 4.5(b)). Comparing these results with the ones shown in Fig. 2.13, we can see that SynthCRAFT can produce intensity standardization results that are similar to REPLICA.

Thus, in this experiment we have shown that SynthCRAFT is capable of intensity

CHAPTER 4. SYNTHCRAFT

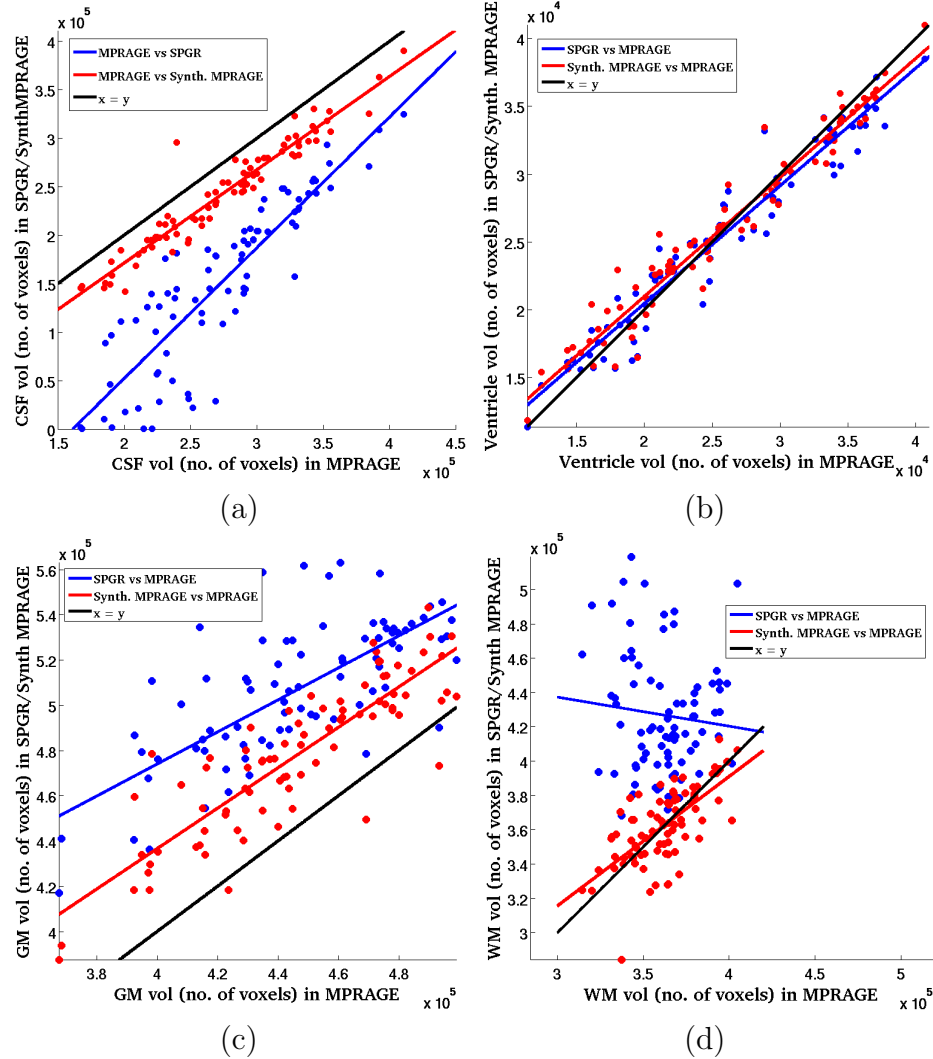


Figure 4.5: (a) CSF AtlasEM volume scatter plot for SPGR vs MPRAGE and synthetic MPRAGE vs MPRAGE, (b) Ventricles AtlasEM volumes, (c) GM AtlasEM volumes, and (d) WM AtlasEM volumes. The blue scatter plots are of volumes observed in SPGR vs those in MPRAGE. The red scatter plots are of volumes observed in synthetic MPRAGE vs those in MPRAGE. The black line indicates the identity transform $x = y$.

standardization tasks like REPLICA and the tissue segmentations results are very similar to tissue segmentation on REPLICA-generated images. SynthCRAFT however offers us the theoretical guarantee of providing the most optimal image based on the

posterior distribution, which REPLICA does not.

4.3.4 Super-resolution of FLAIR

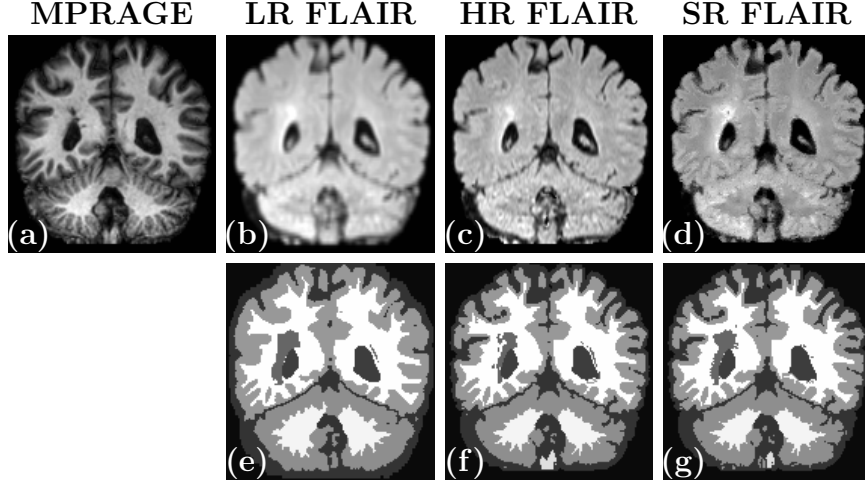


Figure 4.6: Coronal slices of LR, HR, and SR FLAIRs along with their corresponding LesionTOADS segmentation are shown. It is evident that using a LR FLAIR affects the segmentation of the lesions and even the cortex.

Next, we applied SythCRAFT to synthesize super-resolution (SR) FLAIRs using corresponding high resolution (HR) MPRAGE and low resolution (LR) FLAIRs, a similar application to REPLICA’s.

Our approach can be described as an example-based super-resolution [24] technique. Example-based methods leverage the high resolution information extracted from a HR image—an MPRAGE, for example—in conjunction with a LR input image—corresponding FLAIR image—to generate a SR version of the LR image. We used HR ($1 \times 1 \times 1 \text{ mm}^3$) MPRAGE and FLAIR data, and downsampled the HR FLAIR to create a LR ($1 \times 1 \times 4 \text{ mm}^3$) FLAIR. The atlas data included an HR MPRAGE

+ LR FLAIR and we trained SynthCRAFT to predict the HR FLAIR. Given a test HR MPAGE and LR FLAIR, we applied SynthCRAFT to synthesize a SR FLAIR. We ran the LesionTOADS segmentation algorithm [55] on three scenarios for each subject: (a) HR MPAGE + LR FLAIR (b) HR MPAGE + SR FLAIR (c) HR MPAGE + HR FLAIR. The last case acting as the ground truth for how the segmentation algorithm should behave on best case data. We aim to show that the tissue segmentation using SR FLAIR is closer to that achieved using HR FLAIR, than using LR FLAIR. Figure 4.6(d) shows the super-resolution results, the LR FLAIR image is shown in Fig. 4.6(b), and the HR FLAIR image in Fig. 4.6(c). The corresponding LesionTOADS segmentations are shown in Figs. 4.6(e), (f), and (g), respectively. The lesion boundaries as well as the cortex is overestimated when a LR FLAIR is used. Shown in Fig. 4.7 are the lesion volumes on 13 subjects for each of the three scenarios.

Thus, SynthCRAFT can also be used for FLAIR super-resolution. The lesion segmentation results are very similar to REPLICA, even though the quality of images is not as good.

4.4 Summary and Discussion

We have described an image synthesis framework, SynthCRAFT, as an inference problem on a Gaussian CRF. The parameters of the Gaussian CRF are built from

CHAPTER 4. SYNTHCRAFT

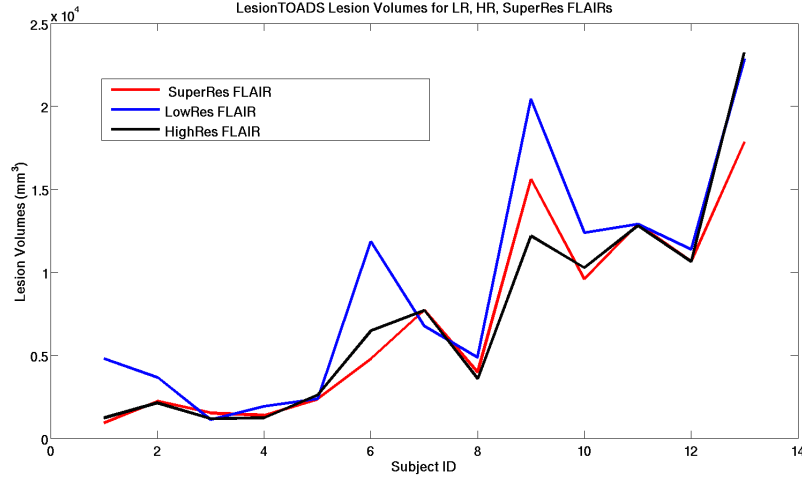


Figure 4.7: Shown are the lesion volumes acquired by LesionTOADS on HR FLAIR+HR MPRAGE (black), LR FLAIR+HR MPRAGE (blue), and SR FLAIR+HR MPRAGE (red). Note that the black plot is closer to the red plot than the blue plot for all but one of the subjects.

parameters stored at the leaves of a single regression tree. Parameter learning is done by maximizing a pseudo-likelihood objective function.

We have shown that SynthCRAFT is competitive with the state-of-the-art image synthesis algorithms. The quality of synthesis is not as good as REPLICa in terms of image quality metrics, but we believe that with the right set of parameters, this gap can be bridged easily. We intend to perform a rigorous parameter selection experiment, similar to REPLICa in the future.

SynthCRAFT was also shown to be effective for FLAIR synthesis and example-based super-resolution of FLAIR images. It can be applied in realistic scenarios, where imaging data is missing and needs to be replaced by a feasible alternative. Finally we also applied our algorithm to enhance the resolution of low resolution FLAIRs and showed improved tissue segmentation as a result. The segmentation results obtained

CHAPTER 4. SYNTHCRAFT

were comparable to REPLICA and we hope to demonstrate more applications with continuing improvements to the algorithm.

SynthCRAFT is flexible in terms of features it can use to create the initial regression tree. It is also general enough to add larger neighborhoods and long-range relationships among voxels. Adding more neighbors leads to additional parameters, but these can be stored in the same initial tree and we do not need to create any more trees. Our approach is also computationally efficient, training from millions of samples in 20–30 minutes. Inference takes less than five minutes. The parameter learning algorithm is a projected gradient descent algorithm that we have parallelized over the training voxels. With clever programming we intend to speed it up even more, the implementation of which is ongoing.

The multi-scale nature of REPLICA can be easily added into SynthCRAFT. This would involve adding more ‘neighbor’ relationships, not across the same image but across scales of a Gaussian pyramid. Therefore a voxel in the highest resolution level will have a neighboring voxel in the level lower to itself. This will add another pairwise potential, which will be a cross-scale potential. It will also mitigate the oversmoothing caused in REPLICA results when information from lower resolution levels dominates that from the higher resolution ones.

Another avenue to extend this work is for longitudinal image processing. MRI data that has been acquired longitudinally can be used for synthesis by considering spatial as well as temporal neighbors of voxels. This can be especially useful for time-series

CHAPTER 4. SYNTHCRAFT

MRI acquisitions like functional MRI (fMRI). None of these extensions is obviously available to REPLICA.

Chapter 5

Discussion, Conclusions, and Future Work

5.1 Summary

In this thesis, we have described three different image synthesis algorithms. In Chapter 2, we explored the feature design and the strengths and limitations of random forest regression to come up with a multi-resolution image synthesis algorithm that we called REPLICA. In Chapter 3 we outlined Ψ -CLONE, which estimated the pulse sequence parameters from a given image and used them to augment the atlas dataset, which allowed the subsequent random forest regression to produce a synthetic image. Finally in Chapter 4 we formulated a probabilistic, discriminative framework which framed the image synthesis problem as inference on a Gaussian CRF. Here, we

summarize what we have learned and what we need to do to improve in the future.

5.2 REPLICA

5.2.1 Key Points and Results

- REPLICA is multi-resolution, random forest regression algorithm for image synthesis. REPLICA assumes that the subject image is easily standardized to the atlas data. It uses innovative features that combine local and global information to provide an unambiguous intensity mapping from input modalities to the output modality.
- REPLICA was shown to be significantly better than state-of-the-art algorithms on the T_2w synthesis task. For that particular task and that particular dataset, REPLICA provides better results than the published version Ψ -CLONE. However, Ψ -CLONE updated with REPLICA as a component would perform even better.
- Using REPLICA we performed full-head T_2w image synthesis given a T_1w image as input. Full-head MR image synthesis is a challenging image synthesis problem due to high variability in anatomy and the tissue MR intensities and was not performed successfully by other competing algorithms.
- We used REPLICA for FLAIR synthesis and went ahead and performed lesion segmentation to demonstrate that synthetic FLAIR images provided a similar

CHAPTER 5. CONCLUSIONS

segmentation as real FLAIR images, which no previous algorithm had shown.

- REPLICa was also shown to perform example-based super-resolution of T_2 w and FLAIR images and we also showed that the super-resolution FLAIR images improved the lesion segmentation, which also was the first such demonstration for super-resolution approaches in MRI.
- Intensity standardization between SPGR and MPRAGE was also performed using REPLICa and further tissue segmentation results confirmed that synthesis can be used to reconcile to diverse datasets.
- REPLICa has also been shown to improve cross-modality registration [44], where instead of registering a T_1 w image to a T_2 image, it is beneficial to register it to a synthetic T_1 w image generated from the target T_2 w image.
- Like Ψ -CLONE, REPLICa is also computationally fast. Training can take up to 20 minutes, but is a one-time task. Actual synthesis can be easily parallelized over multiple cores, and can be performed in under a minute. Given the typical times of neuroimaging pipelines (usually many hours), this makes the use of REPLICa as a preprocessing step quite feasible.

5.2.2 Future Work

- REPLICa uses random forest regression, which predicts by averaging the values in leaf node. This leads to REPLICa results appearing slightly smoother and

CHAPTER 5. CONCLUSIONS

with less noise than in real images. Though this characteristic of synthetic images is not always bad, it can lead to algorithms behaving differently for synthetic images than for real images, which is not desirable. This was one of the motivations for development of SynthCRAFT, later in our research.

- The features designed for REPLICA are useful in most image synthesis tasks, but have been hand-designed with the human expertise, and hence may not be general for all possible tasks. In the future we envision a feature discovery module added to REPLICA that will automatically discover features that are necessary for that particular synthesis task and only use those. Recent advances in deep learning in the machine learning literature are encouraging [96], but these methods are not yet powerful enough to handle the high-dimensional medical imaging data.
- One of the issues with REPLICA is that it is an ad-hoc approach towards image synthesis. It works, but it is not clear how to interpret it in a probabilistic setting. Additionally, as we mentioned earlier, REPLICA generates an image voxel-by-voxel, each independent of the rest, when it is known that these voxels, especially neighboring ones are not independent of each other. The drive to formulate a solution to this problem led to the creation of SynthCRAFT. However, the wisdom gained on the way to creating REPLICA proved invaluable for SynthCRAFT.

5.3 Ψ -CLONE

5.3.1 Key Points and Results

- Ψ -CLONE estimates the acquisition parameters of a given image to generate a synthesis that respects the MR image formation process. Previous work in MR image synthesis had so far ignored the image acquisition process. Our process of using the underlying NMR tissue parameters to construct additional atlas images that are adaptable to the input subject image is unique.
- Ψ -CLONE was validated on phantom data and was shown to perform intensity standardization of T_1 w images across time and on large datasets. It was also used to synthesize T_2 w and FLAIR pulse sequences and shown to be superior to other competing approaches. We also showed it capable of performing example-based super-resolution of low resolution T_2 w images.
- Ψ -CLONE is computationally very fast and can be used as preprocessing tool to reconcile disparate datasets, replace a missing or artifact-affected dataset or improve the quality of existing datasets for the purpose of improving further image analysis.
- Ψ -CLONE can be thought of as a very general MRI image synthesis solution, where we do not need training data to be similar to the test data. Using MRI imaging equations, we can always create the appropriate training data. In

neuroimaging research where appropriate data is indispensable and more often than not, unavailable, Ψ -CLONE offers a solution to bypass this need.

5.3.2 Future Work

- Estimation of pulse sequence parameters is an important step and requires an initial segmentation of the input subject image. So far, our input images have been T_1w images, which are not difficult to segment quickly. However, for different pulse sequence inputs we would need an estimate of the tissue segmentation. One solution is to use other supervised methods to get a crude estimate of the tissue segmentation from the input image.
- The pulse sequence parameters estimated by our method are not equal to the true parameters due to various approximation errors. However, even these “wrong” parameter values when used in a pulse sequence equation produce an image that is very close to the ground truth, as we showed in Chapter 3. Even though the metrics we use to measure the similarity of the images are commonplace; they may have a subtle deficiency that could only be revealed by a far larger and more rigorous study.
- Finally, in the published version of Ψ -CLONE the random forest regression used very simple features like local patches and sub-optimal parameters for the random forest. Further research in REPLICA has given us a much better

understanding of what parameters we need to use. We intend to include these improvements in the next version of Ψ -CLONE.

5.4 SynthCRAFT

5.4.1 Key Points and Results

- In SynthCRAFT, we have formulated image synthesis as an inference problem in a probabilistic graphical model framework. SynthCRAFT models the conditional distribution of the desired synthetic image given the available input images as a Gaussian conditional random field.
- The parameters of the GCRF are stored in leaves of a regression tree, which is built using available training data. Parameter learning is achieved by pseudo-likelihood maximization.
- Given a subject image, the synthetic image is predicted by building the conditional distribution and carrying out MAP estimation using conjugate gradient descent.
- SynthCRAFT was shown to be capable of performing T_2w synthesis and FLAIR synthesis with similar segmentations reported for real and synthetic FLAIRs. The quality of synthesis is not as high as REPLICa, but with better parameter tuning, can improve considerably.

CHAPTER 5. CONCLUSIONS

- SynthCRAFT was also used for example-based super-resolution of low resolution FLAIR images and segmentation results of the super-resolution images were better than the low resolution inputs, again similar to REPLICA.
- The probabilistic nature of SynthCRAFT enables us to determine a synthetic image that is optimal for a well-defined objective, unlike REPLICA, where the output is largely the result of intuitive engineering choices.
- SynthCRAFT is also computationally efficient, training from millions of samples in 20–30 minutes. Inference takes less than five minutes.

5.4.2 Future Work

- SynthCRAFT is an attractive theoretical concept, but implementation-wise it includes many parameters that need to be tuned for improving the final result. REPLICA also has small set of such parameters that we tested empirically. Such an analysis is essential for SynthCRAFT.
- Since SynthCRAFT also derives information from a regression tree learned from training data, like REPLICA, we need to explore better feature design and automated feature selection/generation.
- The parameter learning process of SynthCRAFT is based on gradient descent and requires the gradient to be calculated at each iteration. It is already parallelized

across voxels, but can be made faster with a serial algorithm, which will use more memory but reduce computing time.

- SynthCRAFT is easily extendible to multi-resolution frameworks like REPLICA by simply adding more neighbors that are between different scales. It is also very easy to add neighbors across time to provide a longitudinal processing algorithm that will synthesize a whole chain of longitudinally acquired images.

5.5 Concluding Thoughts

We have presented three different methods for image synthesis in brain MRI. The goal of this work was to develop an image synthesis tool that will help improve subsequent image analysis. We have shown that our methods help improve segmentation [45,46,97], and registration [44,98]. Our methods are fast, computationally cheap and do not have strict requirements on input. We can apply these methods to enhance quality of available datasets, reconcile differences between them, and make possible consistent image analysis. Our hope is that researchers find these tools worth using to solve those problems in their data that otherwise would not have been possible to tackle. We have shown our validation for a certain set of algorithms and on certain sets of data. We would like more validation to happen on the data “in the wild”, and for that to happen we need to make these methods freely available and user-friendly. We intend for this work to culminate into an image synthesis toolkit. Such a toolkit

CHAPTER 5. CONCLUSIONS

would enable reproducible research. We also intend for it to be useful to researchers in the community and is of wide applicability in the field. Our vision for the long term future for these methods is to include them in the MRI scanner firmware. Since these methods are fairly light, implementing them on the scanner is not difficult. Once in the scanner, with access to imaging parameters and the acquired image, our methods can be immediately used to improve image resolution, fix image artifacts, standardize intensities, and generate images that were not included in the imaging protocol, on the fly. A patent describing these ideas has been filed [99]. Such a future would be exciting not only for the tremendous possibilities involved, but also for the dramatic reduction of scanning expenses and increase in patient comfort.

Bibliography

- [1] J. V. Manjón, P. Coupé, A. Buades, , D. L. Collins, and M. Robles, “MRI superresolution using self-similarity and image priors,” *International Journal of Biomedical Imaging*, vol. 425891, p. 11, 2010.
- [2] L. G. Nyúl, J. K. Udupa, and X. Zhang, “New variants of a method of MRI scale standardization,” *IEEE Trans. Med. Imag.*, vol. 19, no. 2, pp. 143–150, 2000.
- [3] N. Burgos, M. Cardoso, K. Thielemans, M. Modat, S. Pedemonte, J. Dickson, A. Barnes, R. Ahmed, C. Mahoney, J. Schott, J. Duncan, D. Atkinson, S. Arridge, B. Hutton, and S. Ourselin, “Attenuation correction synthesis for hybrid PET-MR Scanners: Application to brain studies,” *Medical Imaging, IEEE Transactions on*, vol. 33, no. 12, pp. 2332–2341, Dec 2014.
- [4] S. Roy, A. Carass, and J. L. Prince, “Magnetic resonance image example-based contrast synthesis,” *Medical Imaging, IEEE Transactions on*, vol. 32, no. 12, pp. 2348–2363, Dec 2013.
- [5] R. Deichmann, C. D. Good, O. Josephs, J. Ashburner, and R. Turner, “Opti-

BIBLIOGRAPHY

- mization of 3-D MP-RAGE sequences for structural brain imaging,” *NeuroImage*, vol. 12, no. 1, pp. 112–127, 2000.
- [6] G. H. Glover, *Handbook of MRI pulse sequences*. Elsevier Academic Press, 2004, vol. 18.
- [7] M. Styner, C. Brechbuhler, G. Szekely, and G. Gerig, “Parametric estimate of intensity inhomogeneities applied to MRI,” *IEEE Trans. Med. Imag.*, vol. 19, no. 3, pp. 153–165, 2000.
- [8] M. Prastawa, J. Gilmore, W. Lin, and G. Gerig, “Automatic segmentation of neonatal brain MRI,” in *7th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2004)*, 2004, pp. 10–17.
- [9] S. Roy, A. Carass, P. L. Bazin, S. M. Resnick, and J. L. Prince, “Consistent segmentation using a Rician classifier,” *Medical Image Analysis*, vol. 16, no. 2, pp. 524–535, 2012.
- [10] A. M. Dale and B. Fischl, “Cortical surface-based analysis: Segmentation and surface reconstruction,” *NeuroImage*, vol. 9, no. 2, pp. 179–194, 1999.
- [11] X. Han, D. L. Pham, D. Tosun, R. E. Rettman, C. Xu, and J. L. Prince, “CRUISE: Cortical reconstruction using implicit surface evolution,” *NeuroImage*, vol. 23, no. 3, pp. 997–1012, 2004.
- [12] N. Shiee, P.-L. Bazin, J. L. Cuzzocreo, C. Ye, B. Kishore, A. Carass, P. A.

BIBLIOGRAPHY

- Calabresi, D. S. Reich, J. L. Prince, and D. L. Pham, “Robust reconstruction of the human brain cortex in the presence of the WM lesions: Method and validation,” *Human Brain Mapping*, vol. 35, no. 7, pp. 3385–3401, 2014.
- [13] M. A. Lee, S. Smith, J. Palace, and P. M. Matthews, “Defining multiple sclerosis disease activity using MRI T2-weighted difference imaging,” *Brain*, vol. 121, no. 11, pp. 2095–2102, 1998.
- [14] J. Simon, D. Li, A. Traboulsee, P. Coyle, D. Arnold, F. Barkhof, J. Frank, R. Grossman, D. Paty, E. Radue, and J. Wolinsky, “Standardized MR imaging protocol for Multiple Sclerosis: Consortium of MS centers consensus guidelines,” *American Journal of Neuroradiology*, vol. 27, no. 2, pp. 455–461, 2006.
- [15] C. H. Polman, S. C. Reingold, B. Banwell, M. Clanet, J. A. Cohen, M. Filippi, K. Fujihara, E. Havrdova, M. Hutchinson, L. Kappos, F. D. Lublin, X. Montalban, P. O’Connor, M. Sandberg-Wollheim, A. J. Thompson, E. Waubant, B. Weinshenker, and J. S. Wolinsky, “Diagnostic criteria for multiple sclerosis: 2010 Revisions to the McDonald criteria,” *Annals of Neurology*, vol. 69, no. 2, pp. 292–302, 2011.
- [16] S. Stuckey, T.-D. Goh, T. Heffernan, and D. Rowan, “Hyperintensity in the subarachnoid space on FLAIR MRI,” *American Journal of Roentgenology*, vol. 189, no. 4, pp. 913–921, 2007.
- [17] S. Roy, A. Carass, J. L. Prince, and D. L. Pham, “Longitudinal patch-based

BIBLIOGRAPHY

- segmentation of multiple sclerosis white matter lesions,” in *Machine Learning in Medical Imaging*, ser. Lecture Notes in Computer Science, L. Zhou, L. Wang, Q. Wang, and Y. Shi, Eds. Springer International Publishing, 2015, vol. 9352, pp. 194–202. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-24888-2_24
- [18] K. A. Clark, R. P. Woods, D. A. Rottenberg, A. W. Toga, and J. C. Mazziotta, “Impact of acquisition protocols and processing streams on tissue segmentation of T1 weighted MR images,” *NeuroImage*, vol. 29, no. 1, pp. 185–202, 2006.
- [19] X. Han and B. Fischl, “Atlas renormalization for improved brain MR image segmentation across scanner platforms,” *IEEE Trans. Med. Imag.*, vol. 26, no. 4, pp. 479–486, 2007.
- [20] A. Madabhushi and J. K. Udupa, “New methods of MR image intensity standardization via generalized scale,” *Med. Phys.*, vol. 33, no. 9, pp. 3426–3434, 2006.
- [21] F. Jäger, L. Nyúl, B. Frericks, F. Wacker, and J. Hornegger, *Whole body MRI intensity standardization*. Springer, 2007.
- [22] F. Jager and J. Hornegger, “Nonrigid registration of joint histograms for intensity standardization in magnetic resonance imaging,” *IEEE Trans. Med. Imag.*, vol. 28, no. 1, pp. 137–150, 2009.
- [23] N. I. Weisenfeld and S. K. Warfield, “Normalization of joint image-intensity

BIBLIOGRAPHY

- statistics in MRI using the Kullback-Leibler divergence,” in *IEEE Intl. Symp. on Biomed. Imaging*, 2004, pp. 101–104.
- [24] F. Rousseau, “Brain hallucination,” in *2008 European Conference on Computer Vision (ECCV 2008)*, 2008, pp. 497–508.
- [25] S. Roy, A. Carass, and J. L. Prince, “Synthesizing MR contrast and resolution through a patch matching technique,” in *Proceedings of SPIE Medical Imaging (SPIE-MI 2010), San Diego, CA, February 14-17, 2010*, 2010, p. 7623.
- [26] S. Roy, A. Carass, N. Shiee, D. L. Pham, and J. L. Prince, “MR contrast synthesis for lesion segmentation,” in *7th International Symposium on Biomedical Imaging (ISBI 2010)*, 2010, pp. 932–935.
- [27] F. Rousseau, “A non-local approach for image super-resolution using intermodality priors,” *Medical Image Analysis*, vol. 14, no. 4, pp. 594–605, 2010.
- [28] S. Roy, A. Carass, and J. L. Prince, “A compressed sensing approach for MR tissue contrast synthesis,” in *22nd Inf. Proc. in Med. Imaging (IPMI 2011)*, 2011, pp. 371–383.
- [29] S. Roohi, J. Zamani, M. Noorhosseini, and M. Rahmati, “Super-resolution MRI images using Compressive Sensing,” in *Electrical Engineering (ICEE), 2012 20th Iranian Conference on*, 2012, pp. 1618–1622.

BIBLIOGRAPHY

- [30] A. Jog, S. Roy, A. Carass, and J. L. Prince, “Magnetic resonance image synthesis through patch regression,” in *10th International Symposium on Biomedical Imaging (ISBI 2013)*, 2013, pp. 350–353.
- [31] F. Rousseau and C. Studholme, “A supervised patch-based image reconstruction technique: Application to brain MRI super-resolution,” in *10th International Symposium on Biomedical Imaging (ISBI 2013)*, 2013, pp. 346–349.
- [32] E. Konukoglu, A. van der Kouwe, M. R. Sabuncu, and B. Fischl, “Example-based restoration of high-resolution magnetic resonance image acquisitions,” in *16th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2013)*, vol. 8149, 2013, pp. 131–138.
- [33] N. Burgos, M. Cardoso, M. Modat, S. Pedemonte, J. Dickson, A. Barnes, J. Duncan, D. Atkinson, S. Arridge, B. Hutton, and S. Ourselin, “Attenuation correction synthesis for hybrid PET-MR scanners,” in *Medical Image Computing and Computer-Assisted Intervention MICCAI 2013*, ser. Lecture Notes in Computer Science, K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, Eds., 2013, vol. 8149, pp. 147–154.
- [34] D. H. Ye, D. Zikic, B. Glocker, A. Criminisi, and E. Konukoglu, “Modality propagation: Coherent synthesis of subject-specific scans with data-driven regularization,” in *16th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2013)*, 2013, pp. 606–613.

BIBLIOGRAPHY

- [35] J. E. Iglesias, E. Konukoglu, D. Zikic, B. Glocker, K. Van Leemput, and B. Fischl, “Is synthesizing MRI contrast useful for inter-modality analysis?” in *16th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2013)*, 2013, pp. 631–638.
- [36] S. Roy, A. Jog, A. Carass, and J. L. Prince, “Atlas based intensity transformation of brain MR images,” in *3rd International Workshop on Multimodal Brain Image Analysis at the 16th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2013)*, 2013, pp. 51–62.
- [37] S. Roy, W. T. Wang, A. Carass, J. L. Prince, J. A. Butman, and D. L. Pham, “PET attenuation correction using synthetic CT from Ultrashort Echo-time MR,” *Journal of Nuclear Medicine*, vol. 55, no. 12, pp. 2071–2077, 2014.
- [38] R. Vemulapalli, H. Van Nguyen, and S. Kevin Zhou, “Unsupervised cross-modal synthesis of subject-specific scans,” in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [39] M. I. Miller, G. E. Christensen, Y. Amit, and U. Grenander, “Mathematical textbook of deformable neuroanatomies,” *Proc. Natl. Acad. Sci.*, vol. 90, no. 24, pp. 11 944–11 948, 1993.
- [40] M. Cardoso, C. Sudre, M. Modat, and S. Ourselin, “Template-based multimodal joint generative model of brain data,” in *Information Processing in Medical Imaging*, ser. Lecture Notes in Computer Science, S. Ourselin, D. C. Alexander,

BIBLIOGRAPHY

- C.-F. Westin, and M. J. Cardoso, Eds. Springer International Publishing, 2015, vol. 9123, pp. 17–29.
- [41] T. Hastie, R. Tibshirani, and J. Friedman, “The elements of statistical learning: Data mining, inference and prediction.” 2009.
- [42] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. U.S.A.: Wadsworth Publishing Company, 1984.
- [43] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, “Image analogies,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH ’01, 2001, pp. 327–340.
- [44] M. Chen, A. Carass, A. Jog, J. Lee, S. Roy, and J. L. Prince, “Multi-modal image registration using image synthesis with multi- channel frameworks,” *Medical Image Analysis*, vol. Submitted.
- [45] A. Jog, A. Carass, S. Roy, D. L. Pham, and J. L. Prince, “{MR} image synthesis by contrast learning on neighborhood ensembles,” *Medical Image Analysis*, vol. 24, no. 1, pp. 63 – 76, 2015.
- [46] A. Jog, A. Carass, D. L. Pham, and J. L. Prince, “Tree-encoded conditional random fields for image synthesis,” in *Information Processing in Medical Imaging*, ser. Lecture Notes in Computer Science, S. Ourselin, D. C. Alexander, C.-F.

BIBLIOGRAPHY

- Westin, and M. J. Cardoso, Eds. Springer International Publishing, 2015, vol. 9123, pp. 733–745.
- [47] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [48] A. Jog, A. Carass, and J. L. Prince, “Improving magnetic resonance resolution with supervised learning,” in *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, April 2014, pp. 987–990.
- [49] V. Fonov, A. C. Evans, K. Botteron, C. R. Almli, R. C. McKinsty, and D. L. Collins, “Unbiased nonlinear average age-appropriate brain templates from birth to adulthood,” *NeuroImage*, vol. 47, Supplement 1, 2009, organization for Human Brain Mapping 2009 Annual Meeting.
- [50] B. A. Landman, A. J. Huang, A. Gifford, D. S. Vikram, I. A. L. Lim, J. A. D. Farrell, J. A. Bogovic, J. Hua, M. Chen, S. Jarso, S. A. Smith, S. Joel, S. Mori, J. J. Pekar, P. B. Barker, J. L. Prince, and P. van Zijl, “Multi-parametric neuroimaging reproducibility: A 3-T resource study,” *NeuroImage*, vol. 54, no. 4, pp. 2854–2866, 2011.
- [51] S. Ourselin, A. Roche, G. Subsol, X. Pennec, and N. Ayache, “Reconstructing a 3D structure from serial histological sections,” *Image and Vision Computing*, vol. 19, no. 1, pp. 25 – 31, 2001.

BIBLIOGRAPHY

- [52] M. Modat, G. R. Ridgway, Z. A. Taylor, M. Lehmann, J. Barnes, D. J. Hawkes, N. C. Fox, and S. Ourselin, “Fast free-form deformation using graphics processing units,” *Computer Methods and Programs in Biomedicine*, vol. 98, no. 3, pp. 278 – 284, 2010, hP-MICCAI 2008.
- [53] Z. Wang and A. C. Bovik, “A universal image quality index,” *IEEE Signal Proc. Letters*, vol. 9, no. 3, pp. 81–84, 2002.
- [54] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Proc.*, vol. 13, pp. 600–612, 2004.
- [55] N. Shiee, P. L. Bazin, A. Ozturk, D. S. Reich, P. A. Calabresi, and D. L. Pham, “A topology-preserving approach to the segmentation of brain images with Multiple Sclerosis lesions,” *NeuroImage*, vol. 49, no. 2, pp. 1524–1535, 2010.
- [56] E. Geremia, O. Clatz, B. H. Menze, E. Konukoglu, A. Criminisi, and N. Ayache, “Spatial decision forests for MS lesion segmentation in multi-channel magnetic resonance images,” *NeuroImage*, vol. 57, no. 2, pp. 378–390, 2011.
- [57] X. Llado, A. Oliver, M. Cabezas, J. Freixenet, J. C. Vilanova, A. Quiles, L. Valls, L. Ramio-Torrent, and A. Rovira, “Segmentation of multiple sclerosis lesions in brain MRI: A review of automated approaches,” *Information Sciences*, vol. 186, no. 1, pp. 164 – 185, 2012.

BIBLIOGRAPHY

- [58] A. Carass, J. Cuzzocreo, M. B. Wheeler, P. L. Bazin, S. M. Resnick, and J. L. Prince, “Simple paradigm for extra-cerebral tissue removal: Algorithm and analysis,” *NeuroImage*, vol. 56, no. 4, pp. 1982–1992, 2011.
- [59] J. M. Bland and D. G. Altman, “Statistical methods for assessing agreement between two methods of clinical measurement,” *The Lancet*, vol. 327, no. 8476, pp. 307 – 310, 1986.
- [60] D. L. Pham, C. Xu, and J. L. Prince, “Current methods in medical image segmentation,” *Annual Review of Biomedical Engineering*, vol. 2, no. 1, pp. 315–337, 2000.
- [61] K. Van Leemput, F. Maes, D. Vandermeulen, and P. Suetens, “Automated model-based tissue classification of MR images of the brain,” *Medical Imaging, IEEE Transactions on*, vol. 18, no. 10, pp. 897–908, Oct 1999.
- [62] L. G. Nyúl and J. K. Udupa, “On standardizing the MR image intensity scale,” *Mag. Reson. Med.*, vol. 42, no. 6, pp. 1072–1081, 1999.
- [63] A. Jog, S. Roy, A. Carass, and J. L. Prince, “Pulse sequence based multi-acquisition MR intensity normalization,” in *Proceedings of SPIE Medical Imaging (SPIE-MI 2013), Orlando, FL, February 9-14, 2013*, vol. 8669, 2013, pp. 86 692H–86 692H–8.
- [64] M. Thambisetty, J. Wan, A. Carass, Y. An, J. L. Prince, and S. M. Resnick, “Lon-

BIBLIOGRAPHY

- gitudinal changes in cortical thickness associated with normal aging,” *NeuroImage*, vol. 52, no. 4, pp. 1215–1223, 2010.
- [65] H. Greenspan, G. Oz, N. Kiryati, and S. Peled, “MRI inter-slice reconstruction using super-resolution,” *Mag. Reson. Med.*, vol. 20, no. 5, pp. 437–446, 2002.
- [66] R. R. Peeters, P. Kornprobst, M. Nikolova, S. Sunaert, T. Vieville, G. Malandain, R. Deriche, O. Faugeras, M. Ng, and P. Van Hecke, “The use of super-resolution techniques to reduce slice thickness in functional MRI,” *Int. J. Imaging Syst. Technol.*, vol. 14, no. 3, pp. 131–138, 2004.
- [67] R. C. Hardie, K. J. Barnard, J. G. Bogner, and E. E. Armstrong, “Joint MAP registration and high-resolution image estimation using a sequence of undersampled images,” *IEEE Trans. Image Proc.*, vol. 6, no. 2, pp. 1621–1633, 1997.
- [68] J. G. Tamez-Pena, S. Totterman, and K. J. Parker, “MRI isotropic resolution reconstruction from two orthogonal scans,” in *Proceedings of SPIE-MI 2001*, 2001, pp. 87–97.
- [69] Y. Bai, X. Han, and J. L. Prince, “Super-resolution reconstruction of MR brain images,” in *Proceedings of the Conference on Information Sciences and Systems (CISS 2004)*, 2004.
- [70] J. V. Manjón, P. Coupé, A. Buades, V. Fonov, D. L. Collins, and M. Robles,

BIBLIOGRAPHY

- “Non-local MRI upsampling,” *Medical Image Analysis*, vol. 14, no. 6, pp. 784–792, 2010.
- [71] B. Fischl, D. H. Salat, A. J. W. van der Kouwe, N. Makris, F. Segonne, B. T. Quinn, and A. M. Dale, “Sequence-independent segmentation of magnetic resonance images,” *NeuroImage*, vol. 23, no. S1, pp. S69–S84, 2004.
- [72] C. A. Cocosco, V. Kollokian, R. K. S. Kwan, and A. C. Evans, “BrainWeb: Online interface to a 3D MRI simulated brain database,” *NeuroImage*, vol. 5, no. 4, p. S425, 1997.
- [73] R. K. S. Kwan, A. C. Evans, and G. B. Pike, “MRI simulation-based evaluation of image-processing and classification methods,” *IEEE Trans. Med. Imag.*, vol. 18, no. 11, pp. 1085–1097, 1999.
- [74] J. P. Wansapura, S. K. Holland, R. S. Dunn, and W. S. Ball, “NMR relaxation times in the human brain at 3.0 Tesla,” *Jrnl of Mag. Reson. Imaging*, vol. 9, no. 4, pp. 531–538, 1999.
- [75] J. C. Bezdek, “A convergence theorem for the fuzzy ISO-DATA clustering algorithms,” *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 20, no. 1, pp. 1–8, 1980.
- [76] P. Bottomley and R. Ouwerkerk, “The dual-angle method for fast, sensitive

BIBLIOGRAPHY

- T1 measurement in vivo with low-angle adiabatic pulses,” *Journal of Magnetic Resonance, Series B*, vol. 104, no. 2, pp. 159–167, 1994.
- [77] P. L. Bazin and D. L. Pham, “Topology-preserving tissue classification of magnetic resonance brain images,” *IEEE Trans. Med. Imag.*, vol. 26, no. 4, pp. 487–498, 2007.
- [78] J. G. Sled, A. P. Zijdenbos, and A. C. Evans, “A nonparametric method for automatic correction of intensity nonuniformity in MRI data,” *IEEE Trans. Med. Imag.*, vol. 17, no. 1, pp. 87–97, Feb. 1998.
- [79] B. Avants, C. Epstein, M. Grossman, and J. Gee, “Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain,” *Medical Image Analysis*, vol. 12, no. 1, pp. 26–41, 2008.
- [80] J. Lecoer, J. Ferré, and C. Barillot, “Optimized supervised segmentation of MS sessions from multispectral MRIs,” in *MICCAI workshop on Med. Image Anal. on Multiple Sclerosis*, 2009, pp. 1–11.
- [81] F. Forbes, S. Doyle, D. García-Lorenzo, C. Barillot, and M. Dojat, “Adaptive weighted fusion of multiple MR sequences for brain lesion segmentation,” in *7th International Symposium on Biomedical Imaging (ISBI 2010)*, 2010, pp. 69–72.
- [82] A. Jog, A. Carass, D. L. Pham, and J. L. Prince, “Random forest FLAIR

BIBLIOGRAPHY

- reconstruction from T1, T2, and PD-weighted MRI,” in *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, April 2014, pp. 1079–1082.
- [83] S. Z. Li and S. Singh, *Markov Random Field Modeling in Image Analysis*. Springer, 2009, vol. 26.
- [84] A. Blake, “Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 1, pp. 2–12, Jan 1989.
- [85] D. Geman and G. Reynolds, “Constrained restoration and the recovery of discontinuities,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 3, pp. 367–383, Mar 1992.
- [86] S. Kumar and M. Hebert, “Discriminative random fields,” *International Journal of Computer Vision*, vol. 68, no. 2, pp. 179–201, 2006.
- [87] X. Feng, C. Williams, and S. Felderhof, “Combining belief networks and neural networks for scene segmentation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 4, pp. 467–483, Apr 2002.
- [88] V. Vapnik, “Statistical learning theory. 1998,” 1998.
- [89] A. Y. Ng and M. I. Jordan, “On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes,” *Advances in neural information processing systems*, vol. 14, p. 841, 2002.

BIBLIOGRAPHY

- [90] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” ser. ICML ’01, 2001, pp. 282–289.
- [91] D. Lowe, “Object recognition from local scale-invariant features,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, 1999, pp. 1150–1157 vol.2.
- [92] M. J. Wainwright and M. I. Jordan, “Graphical models, exponential families, and variational inference,” *Found. Trends Mach. Learn.*, vol. 1, no. 1-2, pp. 1–305, Jan. 2008.
- [93] M. Tappen, C. Liu, E. Adelson, and W. Freeman, “Learning Gaussian conditional random fields for low-level vision,” in *CVPR 2007*, June 2007, pp. 1–8.
- [94] J. Jancsary, S. Nowozin, T. Sharp, and C. Rother, “Regression tree fields; An efficient, non-parametric approach to image labeling problems,” in *CVPR 2012*, June 2012, pp. 2376–2383.
- [95] J. Besag, “Statistical analysis of non-Lattice data,” *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 24, no. 3, pp. pp. 179–195, 1975.
- [96] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

BIBLIOGRAPHY

- [97] A. Jog, A. Carass, S. Roy, D. L. Pham, and J. L. Prince, “Random forest regression for magnetic resonance image synthesis,” *Medical Image Analysis*, vol. Submitted.
- [98] M. Chen, A. Jog, A. Carass, and J. L. Prince, “Using image synthesis for multi-channel registration of different image modalities,” pp. 94 131Q–94 131Q–7, 2015.
- [99] A. Jog, S. Roy, A. Carass, and J. L. Prince, “Pulse sequence-based intensity normalization and contrast synthesis for magnetic resonance imaging,” July 2013, US Patent App. 13/940,578.

Vita

Amod Jog earned his B. Tech degree in Computer Science and Engineering from the Indian Institute of Technology, Bombay in 2009. He received his M.S.E. degree in Computer Science from the Johns Hopkins University in 2011. He was awarded the Outstanding Teaching Award by the Computer Science department in 2012. Currently he is a PhD candidate in Computer Science at the Johns Hopkins University, and is advised by Dr. Jerry L. Prince. His research focuses on image synthesis methods in magnetic resonance imaging. His general interests include medical image analysis, computer vision, and machine learning.