

Automated Walks using Machine Learning for Segmentation

S. Vyas*, P. Burlina, D. Kleissas, and R. Mukherjee*

The Johns Hopkins University, Applied Physics Laboratory
Laurel, Maryland, 20723, United States of America

**These authors contributed equally to this work*

Abstract. This paper describes an automated algorithm for segmentation of brain structures (CSF, white matter, and gray matter) in MR images. We employ machine learning (using k -Nearest Neighbors) of features derived from k -means, Canny edge detection, and Tourist Walks to fully automate the seeding process of the Random Walker algorithm. We test our methods on the MRBrainS13 dataset, which consists of imagery from diabetes patients with atrophy and varying degrees of white matter lesions, and find encouraging segmentation performance.

Keywords: Random Walker, Tourist Walks, Machine Learning, k -NN, k -Means, Edge Detection

1 Introduction

Magnetic Resonance Imaging (MRI) is a key modality for 3D neuro-imaging. It has unique benefits when compared to other 3D imaging modalities: Unlike CT, PET, or SPECT, which are also used for neuro-imaging, MRI is non-ionizing. 3D ultrasound, while used extensively for other organs, is not adequate for neural imaging because of the shadowing effect of the skull.

3D segmentation of the brain is motivated by a number of applications, including: for performing diagnostics, for assessing the effects of therapy (e.g., medication or radiation), for generating personalized anatomical models in pre-operative planning (e.g., gamma knife therapy or brain surgery), and for clinical or scientific studies (e.g., brain development). In particular, neuro-imaging is used for assessing the nature and extent of various conditions, including tumors, malformations, multiple sclerosis, traumatic brain injury, and recently, tracking the progression of certain types of dementia.

Accurate semi- and fully- automatic 3D segmentation approaches are desirable because manual segmentation is a prohibitively laborious task. Automated segmentation is also important because of the need for objective and repeatable delineation in longitudinal patient studies, given the considerable variations within and across experts resulting from manual delineation.

Work in automated MRI segmentation and brain delineation using MRI has drawn substantial interest. A general taxonomy of methods includes: intensity

based methods using either supervised or unsupervised approaches (e.g., fuzzy c-means [13]); methods using dynamic contours, deformable models or active appearance models [11]; and methods using priors or atlases [2, 10]. While each methodology offers unique advantages and disadvantages, no single method outperforms all others in every area (computational burden, accuracy, and reliance on prior segmentations/ground truth). A comprehensive review of segmentation methods used across 3D MRI is provided in a review by Balafar et al. [5].

2 Method

As discussed, there exists a vast body of literature focused on developing novel methods for the task of accurate brain segmentation. Our goals in this study are two-fold: (1) we don't wish to re-invent the wheel; we take a popular state-of-the-practice semi-automated segmentation algorithm, i.e. Random Walker, and attempt to make it fully-automated. (2) We want to develop a completely generalizable algorithm that is not limited to segmenting MR or even medical imagery. As such, we employ machine learning in lieu of, for example, atlas-based methods for automating Random Walker. This enables the algorithm to perform similarly on any imagery for which example segmentations are available. An overview of our segmentation pipeline can be seen in Figure 2.

2.1 Pre-processing

In this paper, we are not interested in segmenting the skull, or finding a method to automatically identify skull features. As such, in order to remove the skull and other unwanted tissue that surrounds the brain in each MRI image set, we rely on the Brain Surface Extractor (BSE) tool within BrainSuite [16, 15], which is made available by the UCLA Laboratory of Neuro Imaging (LONI) framework [14].

To remove the skull and scalp tissue, BSE performs a three-step procedure as described in [16]. First, anisotropic diffusion is applied to smooth the image while preserving edges. Second, a Marr-Hildreth edge detector is applied to identify the boundaries between the dura and the skull. Finally, the largest connected component is used to generate a mask of brain tissue. This final step is preceded and followed by morphological operations that remove extraneous objects and smooth the mask boundaries.

In addition to being used within BSE, anisotropic diffusion is also applied as a pre-processing step to the MRI imagery prior to segmentation. Edge preservation is an important consideration in our approach as edge information is relied upon heavily to seed Random Walker.

Anisotropic diffusion was first proposed for image filtering by Perona and Malik [12]. As the name suggests, this filtering process is performed by applying 2D anisotropic diffusion, essentially resulting in Gaussian filtering, to each slice. Anisotropic diffusion is implemented by using an iterative process. The

anisotropic variant adds a locally adaptive behavior near edges, where the behavior approximates a pass-through filter. As such, the anisotropic diffusion process results in a non-linear filtering of the images. The diffusion parameter values used in this study were: iterations = 3, $\delta_t = \frac{1}{7}$, $\kappa = 20$. No attempt was made at optimizing the parameters for each contrast level. It is possible that better results could therefore be obtained if parameter optimization is performed on either a slice-by-slice or contrast-level basis. However, this is speculative, and is left as future work.

2.2 Random Walker segmentation

Random Walker (RW) segmentation is a popular algorithm first proposed by Leo Grady [9], which has recently been used in a number of applications. A complete description of the algorithm is beyond the scope of this paper, however, a brief review is included in this section in order to provide a high level understanding of the method.

The RW algorithm is a semi-automated method that requires the user to label pixels in an image (known as seeds), where the seeds correspond to the foreground (i.e., object to be segmented) and the background. This is true of the binary case. In our application, seeds are provided for four classes: background, CSF, GM, and WM. However, this is a trivial extension of the binary case; as such, to motivate understanding of the method, we proceed with the binary case. Once the user has seeded the original image, the pixels that are not selected as seeds are assumed to release a random walker. The algorithm calculates the probability of each random walker arriving at a seed bearing each of the labels. This allows the algorithm to label the pixel as either foreground or background based on whichever seed has the largest probability of receiving the random walker.

The image is modeled as a graph or a network of nodes and edges. The pixels are modeled as the nodes and the edges represent the connections between pixels and their neighbors, i.e. the edges. The edge weights are based on the similarity between neighboring pixels; a common measure of similarity is often the pixel intensity. The probability of a random walker arriving at a particular seed can be computed using the graph Laplacian matrix, given by L . Nodes are labeled as V_i and edges joining nodes i and j are labeled as E_{ij} . Edge weights are given by w_{ij} , and the intensity of a particular node (pixel) is given by g_i . Therefore, w_{ij} can be computed as follows [9]:

$$w_{ij} = \exp(-\beta(g_i - g_j)^2) \quad (1)$$

where β is a constant. The goal is to optimize the energy function given by:

$$Q(x) = x^T L x = \sum_{E_{ij}} w_{ij} (x_i - x_j)^2 \quad (2)$$

where,

$$x_i = \begin{cases} 1, & \text{if } v_i \in \text{Foreground} \\ 0, & \text{if } v_i \in \text{Background} \end{cases} \quad (3)$$

Let S represent the set of all nodes that are seeded, and let \bar{S} represent the set of nodes that are not seeded, then the optimum of the energy function Q can be shown to be the solution to the following:

$$L_{\bar{S},\bar{S}}x_{\bar{S}} = -L_{\bar{S},S}x_S \quad (4)$$

The graph Laplacian matrix is composed of the corresponding sets S and \bar{S} . In other words, the subscripts of the Laplacian matrix denote what portion of the matrix is indexed by the respective set. A complete review of the RW algorithm, as well as foundational mathematics for walks on graphs is provided in [9].

2.3 Seed candidates

While the RW algorithm has been shown to perform well on a wide variety of imagery, the results are largely sensitive to the quality (spatial location, and accuracy) of the seeds. For example, in this application, there are areas that are “easy” to segment (e.g., the center of the ventricles), whereas there are other areas that are much more “challenging” (e.g., boundaries separating two classes). As such, we want to seed the algorithm in a way that maximizes segmentation performance.

We motivate the seeding of RW using Figure 1. As an example, we start by looking at slice 22 (center of the cube looking axially) of training example 1. Because we have the ground truth available for this cube, we can test a number of hypotheses on it. As such, in Figure 1(a) we uniformly sample 25% of each of the four classes we seek to segment (background, CSF, GM, and WM). Therefore, we now have 25% of the pixels from each class accurately labeled. These pixels are provided as seeds to RW and the resulting segmentation is provided in Figure 1(b). This segmentation (only for slice 22) has a Dice coefficient of 96.4%. This is clearly a strong result. This experiment demonstrates that if we provide RW with enough correctly labeled seeds, it will segment the image with high accuracy.

While this is a convincing result, it isn’t practical to use such a uniform sampling methodology. In practice we will not have the ground truth, so we will not be able to sample each class uniformly. Therefore, a method is required to identify candidate locations and properly seed them to better approximate the performance achievable with uniform sampling of the ground truth. The following sections discuss how we attempt to approximate this performance, and we use the other panels of Figure 1 as visual aids.

***k*-means** One of the simplest algorithms for clustering is known as *k*-means. The idea is simply to cluster points based on their values. The algorithm iteratively partitions the observations into a user-defined number of clusters by using the mean value of each cluster, where the mean is updated with each iteration.

After skull stripping, it is possible to segment CSF, GM, and WM relatively easily for large portions of bias-corrected MRI brain imagery using intensity alone. The majority of intensities in each class differ enough for *k*-means to provide robust segmentation results for the body of each segmentation target.

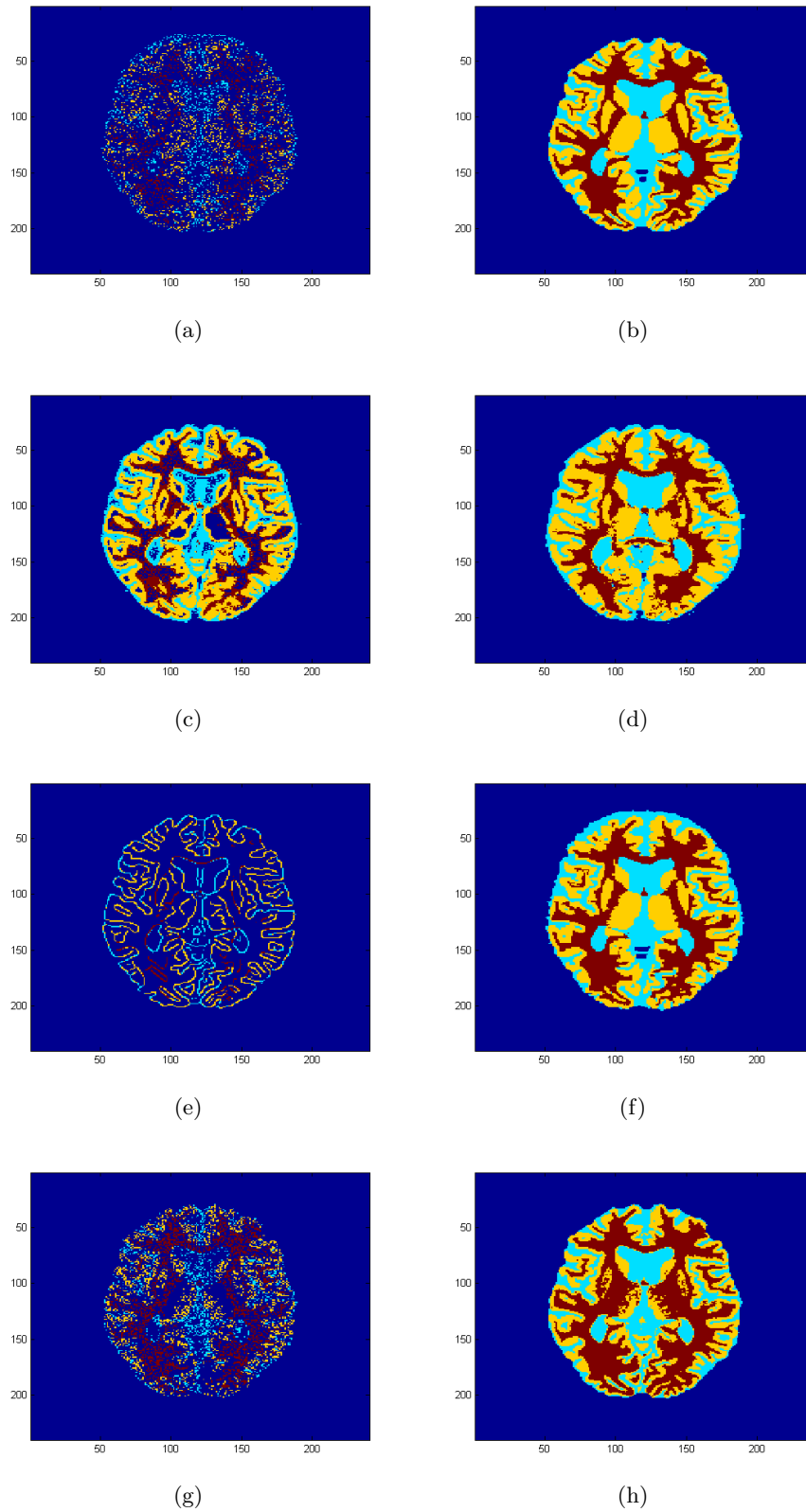


Fig. 1. The figures in the left column are the seeds to RW (for axial slice 22), and the figures on the right are the resulting segmentations. (a) - (b) denotes the 25% uniform sampling with a Dice of 0.9639, (c) - (d) denotes 40% of the (correct) k -means with a Dice of 0.8015, (e) - (f) denotes the ground truth Canny with a Dice of 0.9315, and finally (g) - (h) is 40% k -means + Canny from this work with a Dice of 0.8762.

We exploit the k -means segmentation results by only using the most robust portions of them as seed points to later algorithms. To do this, we first run k -means, providing it with an initial guess for the means of each cluster. These initial means are set so that each class is labeled consistently across all patients. After k -means finishes, each pixel is assigned a probability of belonging to each of the k classes based on the relative distances to the means of each class. These probabilities are then thresholded, and only the labels that have been assigned with high probability (experimentally determined to be 94% and higher) are used as seed points for RW segmentation.

The idea of using k -means stems from the desire to approximate the RW performance when uniform sampling pixels from the ground truth. We observed that seeding RW with 25% of the pixels was enough to produce a highly accurate segmentation. As such, we can hypothesize that if we seed it with points where k -means is confident, we can approximate that behavior. Figure 1(c) shows all the points where k -means is at least 94% confident in correctly labeling pixels. We take 40% of the confident points (which are correct based on the ground truth) as seeds and provide them to RW. The segmentation result is shown in Figure 1(d), and the Dice coefficient is 80.2%. While the result looks visually acceptable, the Dice coefficient is far from what is possible using uniform sampling of the ground truth. This result is not unexpected. We posit that there are areas which are “hard” to segment (i.e., class boundaries). At these areas, k -means is not confident in its segmentation because pixel intensities blur between multiple classes and are most distant from the means of each cluster. This experiment shows that RW requires at least some of these “hard” areas as seeds in order to achieve better segmentation performance.

Canny edge detection We hypothesize that some of these hard-to-segment areas include the edges and boundaries that separate classes. Therefore, we seek to test if providing RW with these edges will help it achieve a superior segmentation performance.

Edge detection has been studied in great detail by the image processing community. In this work, we employ a widely used method first proposed by John Canny, known as Canny edge detection. The method has widely been accepted as finding optimal edges compared to other methods. A review of the method is provided in literature [7].

We again use Figure 1 as a motivating example. We start by finding all the pixels that correspond to the Canny edges on slice 22. For testing our hypothesis, we use the ground truth to label each of these edge pixels correctly, and then treat these pixels as our seeds to RW. The seeds are shown in Figure 1(e). The resulting segmentation is provided in Figure 1(f), and the Dice coefficient is 93.2%. This is once again a strong result, and while it is a bit shy of the uniformly sampled result, it can be explained by the fact that we only use approximately 7% of the total pixels compared to 25% before.

Therefore, we have determined that in order to achieve good segmentation performance from RW, we need to correctly seed it with labels near the class

boundaries. Given the resolution and quality of the imagery, we found that sometimes the edge is composed of more than one pixel. As such, we dilate the edge using a 4×4 morphological kernel.

These results must be interpreted with caution. In other words, these results were obtained under the assumption that all the Canny edges are labeled correctly (using the ground truth). In practice, this cannot always be achieved. As such, we saw it fit that, in addition to using the Canny edges, we also use the confident k -means points from non-edge locations. This served two purposes: (1) it helps mitigate Canny seeds that aren't labeled correctly, and (2) it increases the total number of points provided as seeds. For comparison purposes, Figure 1(g) - (h) shows the seeds for Canny with k -means and the segmentation with a Dice of 87.6%. Note that this result is obtained using our machine learning method, which will be discussed in Section 2.5, i.e. no ground truth is used. This analysis presents encouraging evidence that using Canny and the non-edge k -means points is a viable way to seed the RW algorithm and achieve reasonable performance.

2.4 Feature extraction

The goal of this paper is to treat segmentation as a machine learning (supervised classification) problem. As such, we need to automate the seeding step of the RW method. We showed an optimal way to select the seeds that will be used for the RW method, i.e. we found that the optimal seeds were from (1) the Canny edges, and (2) the k -means high-confidence points. By performing a k -means segmentation, we have access to the labels for all high-confidence, non-edge k -means seeds. However, we do not know the labels for the Canny edge seeds. Therefore, we use feature extraction and machine learning in order to ascertain those labels.

Following a machine learning paradigm, for each pixel that will act as a seed for the RW method, we need to instantiate a feature vector that will help determine that pixel's correct class label. This feature vector is used by the machine learning algorithm along with examples of expert-segmented ground truth. In this work, ground truth comes in the form of five expert-segmented cubes, which are provided as part of a training dataset.

Gray levels As in the k -means results, the gray value is often enough to coarsely segment the image into the four classes. In the case of the Canny edge pixels, we cannot rely entirely on the gray value, but we can use it, in part, for classification. As such, one dimension of our feature vector is the physical gray value corresponding to the pixel being classified.

Tourist Walk As an additional feature to discriminate between segmentation classes, we use the average intensity over a simplified deterministic tourist walk. Deterministic tourist walks have been used recently as a relatively robust method for encoding and classifying texture [3, 6, 4, 8]. One issue with intensity-based

segmentation techniques is that they tend to perform poorly when there are spatial intensity variations between classes, or when the intensity boundaries between classes overlap. In these scenarios, spatial information is required to accurately determine which class a particular pixel belongs to.

In the case of MR imagery, deterministic tourist walks allow for a more robust estimate of each pixel’s intensity by incorporating spatial intensity variations. Deterministic tourist walks are performed by starting at each pixel and iteratively stepping to unvisited pixels with the smallest intensity difference, as compared with the current pixel, in an 8-connected neighborhood. The intensity values of each of pixel visited during the walk are averaged together and used as one dimension of a feature vector for classification. For all of our experiments, a walk distance of five steps was used.

Multi-contrast segmentation In this work, we have access to three different contrast MR images: T1, T1-IR, and T2-FLAIR. Each of these images has a reasonably unique distribution of gray levels associated with it. We originally computed the gray level feature, and the Tourist Walk feature for only the T1 images. We can now concatenate to the 2D feature vector four additional dimensions composed of the gray level and average Tourist Walk intensity as computed on the T1-IR and T2-FLAIR images. This results in a 6D feature vector. Note that the Canny edges are not re-computed for the other two contrast levels. Instead, we use the Canny edge locations from the T1 imagery, and extract features for the corresponding locations in the IR and FLAIR images. This is possible because the images are registered.

Multi-view segmentation The MR imagery, as well as other 3D imagery, is represented on a digital system as a stack of 2D slices. In neuro-imaging, this stack has three principal planes: (1) axial, (2) coronal, and (3) sagittal. Segmentation (2D) is typically performed with respect to one of these principal planes. For example, the images in Figure 1 are axial slices. There is, however, no restriction from doing segmentation along the other planes. In this paper, we perform segmentation along all three planes. The exact same seed selection and feature extraction methods from above are used for the sagittal and coronal views.

When segmenting along each of these planes, we end up with three segmentations; these need to be combined into a single one. The RW algorithm, for each pixel, returns a probability of belonging to each of the four classes. In this work, we found two ways of combining the segmentations: (1) we can average the probabilities for each pixel from each of the three views, or (2) take the maximum probability for each pixel from each of the three views. In this paper, we average the probabilities, but the second method is also a viable option.

2.5 Supervised classification using k -Nearest Neighbors

After feature vectors have been generated for all of the Canny edge points in each 2D image slice, a classification algorithm is used to determine each point’s

label. Decent segmentation performance was obtained using a simple k -Nearest Neighbors (k -NN) classifier, with k set to 80 in all experiments. However, it may be possible to increase performance using another classification method.

Regardless of the classification method used, some training is required. The training data that we provide to k -NN is derived from the training dataset by running Canny edge detector and extracting 6D feature vectors in an identical fashion as described previously. To improve classification performance, we exploit the relatively similar spatial location of brain data in both the testing and training image sets. We do this by only providing k -NN with training feature vectors extracted from the training imagery between the two slices above and two slices below the current slice being segmented. This step may not be necessary if a more robust classification method is used. However, because of the spatial similarity of image content across the testing and training datasets, restricting the training data used by k -NN for each slice resulted in a slight performance increase.

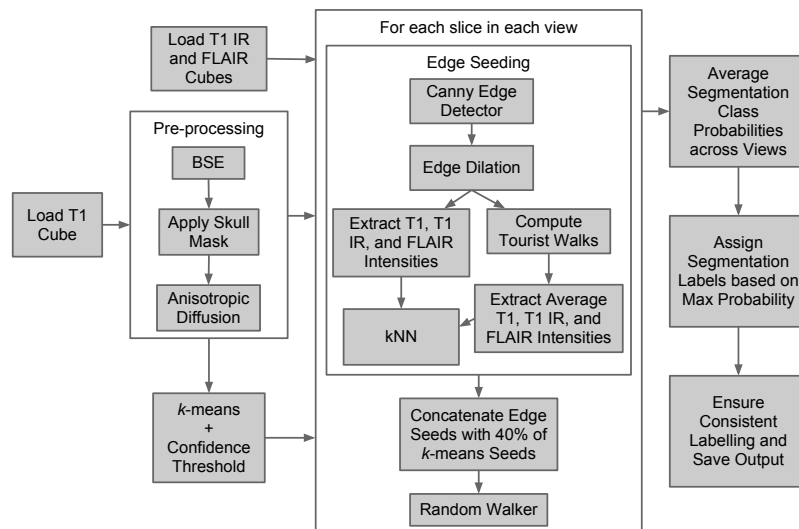


Fig. 2. Block diagram detailing how each cube of MR images is processed.

3 Experiments

3.1 Data

Data used for training and testing our segmentation algorithm consisted of registered and bias-corrected thick-slice T1-weighted scan, T1-weighted inversion recovery scan, and T2-weighted FLAIR scan imagery, each with a spatial resolution of $0.958 \times 0.958 \times 3.0\text{mm}$ for $240 \times 240 \times 48$ voxels. Imagery was collected using a 3T MRI machine at the UMC Utrecht (the Netherlands) from patients with diabetes and matched controls (with increased cardiovascular risk) with varying degrees of atrophy and white matter lesions (age 50 and above).

3.2 Results

The results for 12 patients included in the MRBrainS13 Testing Dataset are tabulated in Table 1. The measures include the Dice Coefficient (Dice) measured in percentage, the modified Hausdorff distance (MHD) measure, defined as the 95th-percentile Hausdorff distance measured in mm, and the absolute brain (gray matter + white matter) volume difference (ABVD) measured in percentage, are included. Each measure is averaged over all 12 patients.

Table 1. Results for 12 patients included in the MRBrainS13 Testing Dataset.

Structure	Dice (%)		MHD (mm)		ABVD (%)	
	Mean	SD	Mean	SD	Mean	SD
Gray matter	77.09	6.29	4.13	3.10	9.70	8.63
White matter	79.61	10.38	4.89	3.99	14.61	13.05
Cerebrospinal fluid	72.45	2.94	3.79	0.90	18.21	9.65
Brain	93.05	1.78	5.85	5.30	3.92	3.11
All intracranial structures	93.93	1.65	20.50	3.00	7.53	2.73

From a computational perspective, each of the three views for a cube takes approximately 27 seconds to segment on a 2.4GHz Intel Xeon E5-2609 running single-threaded in Matlab R2013a. Of course each view (and multiple other steps) are independent and hence can be parallelized.

4 Discussion

The results in Table 1 show that the methods detailed in this paper provide reasonable segmentation results, especially given pathology. It’s worth noting that the analysis presented in Figure 1 is independent of the results or the k -NN algorithm. As such, if a more sophisticated machine learning algorithm is used, it’s possible that one can achieve the performance seen in Figure 1. Moreover, better feature selection (apart from gray levels and Tourist Walks) can also play a key role in approximating the performance seen there. We believe a key contribution of this paper lies in our investigation of automating the RW method. While our results don’t match the theorized performance seen in Figure 1, we posit that other studies could try other features and learning methods to better achieve such performance.

We compare the results with the recent NeoBrains [1] challenge: the minimum and maximum Dice coefficients were (0.61, 0.77) for CSF, (0.87, 0.91) for white matter, and (0.83, 0.85) for gray matter. While a comparison with this datasets should be considered with caution, given that the above cited efforts were applied on a different dataset, we believe that our results appear promising. This is taking into consideration that we only use gray level information, and

that limited spatial contextual information (such as provided by an atlas, as is used for example in [2, 17]) is exploited.

In this paper, we do not use atlas based methods, but rely on machine learning in order to perform segmentation. This choice is in part due to the fact that atlas based methods are not as generalizable because they (a) require an atlas, (b) require registration of the atlas to the data, (c) are typically much slower than other methods, and (d) are not entirely robust to pathology; atlases that capture shape variation due to pathology are still an active area of investigation. While our results might be shy of the performance of some atlas based methods, we posit that we don't suffer from any of these noted disadvantages. Moreover, machine learning could in part better deal with pathology if examples of said pathology are provided, as has been suggested in recent machine learning in medical imaging literature [19, 18]. None-the-less, we still need to make a number of improvements to our learning and feature extraction steps in order to make the method on-par with atlas-based algorithms.

One of the most sizable sources of error in our segmentation results is derived from the skull stripping masks. We made no effort to fine tune the skull stripping algorithm's parameters, or even test different skull stripping algorithms. A close examination of the skull stripping masks generated for many of the training and testing images reveals that there are frequent and sizable over- and under-estimations of the skull boundary.

When the skull boundary is too strict, it crops away regions around the brain that, according to the ground truth, contain CSF. As a result, our segmentation algorithm may never get the chance to correctly label points that should be CSF as CSF, but instead simply marks them as background, resulting in less than stellar CSF segmentation performance statistics.

When the skull boundary is too tolerant, it passes through skull structures and tissue that should be classified as background. Because the intensities of these spurious structures match the intensities of CSF, gray matter, and white matter, they end up being classified as such, which reduces segmentation performance.

5 Conclusion

We describe an automated algorithm for segmenting brain structures in MR images and show promising performance results on the MRBrainS13 Testing Dataset. Insights are also provided regarding performance expectations when RW is correctly seeded, what correct seeding means, and how one might go about automatically seeding RW. Future work will involve testing alternative feature vectors and classification methods to increase the number of correctly labeled seed points, and therefore also increase segmentation accuracy.

Acknowledgments. We would like to thank the Johns Hopkins University, Applied Physics Laboratory, Research and Exploratory Development Department, for financial support for this work.

References

1. Neobrain 2012: Results of the web-based challenge, http://neobrain12.isi.uu.nl/mainResults_Set1.php
2. Akselrod-Ballin, A., Galun, M., Gomori, J.M., Brandt, A., Basri, R.: Prior knowledge driven multiscale segmentation of brain mri (2007)
3. Backes, A.R., Bruno, O.M., Campitelli, M.G., Martinez, A.S.: Deterministic tourist walks as an image analysis methodology based. In: Progress in Pattern Recognition, Image Analysis and Applications, pp. 784–793. Springer (2006)
4. Backes, A.R., Gonçalves, W.N., Martinez, A.S., Bruno, O.M.: Texture analysis and classification using deterministic tourist walk. Pattern Recognition 43(3), 685–694 (2010)
5. Balafar, M.A., Ramli, A., Saripan, M.I., Mashohor, S.: Review of brain mri image segmentation methods. Artificial Intelligence Review 33(3), 261–274 (2010)
6. Campitelli, M.G., Batista, P.D., Kinouchi, O., Martinez, A.S.: Deterministic walks as an algorithm of pattern recognition. Physical Review E 74(2), 026703 (2006)
7. Canny, J.: A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on (6), 679–698 (1986)
8. Gonçalves, W.N., Bruno, O.M.: Dynamic texture analysis and classification using deterministic partially self-avoiding walks. In: Advances Concepts for Intelligent Vision Systems, pp. 349–359. Springer (2011)
9. Grady, L.: Random walks for image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28(11), 1768–1783 (2006)
10. Lötjönen, J.M., Wolz, R., Koikkalainen, J.R., Thurfjell, L., Waldemar, G., Soininen, H., Rueckert, D.: Fast and robust multi-atlas segmentation of brain magnetic resonance images. NeuroImage 49(3), 2352–2365 (2010)
11. Patenaude, B., Smith, S.M., Kennedy, D.N., Jenkinson, M.: A bayesian model of shape and appearance for subcortical brain segmentation. Neuroimage 56(3), 907–922 (2011)
12. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. Pattern Analysis and Machine Intelligence, IEEE Transactions on 12(7), 629–639 (1990)
13. Pham, D.L., Prince, J.L.: Adaptive fuzzy segmentation of magnetic resonance images. Medical Imaging, IEEE Transactions on 18(9), 737–752 (1999)
14. Rex, D.E., Ma, J.Q., Toga, A.W.: The loni pipeline processing environment. Neuroimage 19(3), 1033–1048 (2003)
15. Shattuck, D.W., Leahy, R.M.: Brainsuite: an automated cortical surface identification tool. Medical image analysis 6(2), 129–142 (2002)
16. Shattuck, D.W., Sandor-Leahy, S.R., Schaper, K.A., Rottenberg, D.A., Leahy, R.M., et al.: Magnetic resonance image tissue classification using a partial volume model. NeuroImage 13(5), 856–876 (2001)
17. Srhoj-Egekher, V., Benders, M., Kersbergen, K.J., Viergever, M.A., Išgum, I.: Automatic segmentation of neonatal brain mri using atlas based segmentation and machine learning approach. MICCAI Grand Challenge: Neonatal Brain Segmentation 2012 (NeoBrainS12) (2012)
18. Vyas, S., Mukherjee, R., Sosa, F., Burlina, P.: Endocardium Segmentation in 3D Transesophageal Echocardiography. IEEE International Symposium on Biomedical Imaging: From Macro to Nano (2013)
19. Vyas, S., Banerjee, A., Burlina, P.: Estimating physiological skin parameters from hyperspectral signatures. Journal of biomedical optics 18(5), 057008–057008 (2013)