

# Detection and Segmentation of Pathological Structures by the Extended Graph-Shifts Algorithm

Jason J. Corso<sup>1</sup>, Alan Yuille<sup>2</sup>, Nancy L. Sicotte<sup>3</sup>, and Arthur Toga<sup>1</sup>

<sup>1</sup> Center for Computational Biology, Laboratory of Neuro Imaging

<sup>2</sup> Department of Statistics

<sup>3</sup> Department of Neurology, Division of Brain Mapping

University of California, Los Angeles, USA

jcorso@ucla.edu

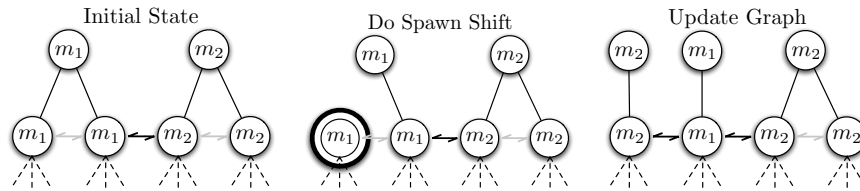
**Abstract.** We propose an extended graph-shifts algorithm for image segmentation and labeling. This algorithm performs energy minimization by manipulating a dynamic hierarchical representation of the image. It consists of a set of moves occurring at different levels of the hierarchy where the types of move, and the level of the hierarchy, are chosen automatically so as to maximally decrease the energy. Extended graph-shifts can be applied to a broad range of problems in medical imaging. In this paper, we apply extended graph-shifts to the detection of pathological brain structures: (i) segmentation of brain tumors, and (ii) detection of multiple sclerosis lesions. The energy terms in these tasks are learned from training data by statistical learning algorithms. We demonstrate accurate results, precision and recall in the order of 93%, and also show that the algorithm is computationally efficient, segmenting a full 3D volume in about one minute.

## 1 Introduction

Automatic detection of pathological brain structures is a problem of great practical clinical importance. From the computer vision perspective, the task is to label regions of an image into pathological and non-pathological components. This is a special case of the well-known image segmentation problem which has a large literature in computer vision [1,2,3] and medical imaging [4,5,6,7,8,9,10]

In previous work [11], we developed a hierarchical algorithm called graph-shifts which we applied to the task of segmenting sub-cortical structures formulated as energy function minimization. The algorithm does energy minimization by iteratively transforming the hierarchical graph representation. A big advantage of graph-shifts is that each iteration can exploit the hierarchical structure and cause a large change in the segmentation, thereby giving rapid convergence while avoiding local minima in the energy function. The algorithm was limited, however, because it required the number of model labels to be fixed and the number of model instances to be known. For example, every brain has a single ventricular system. Nevertheless it was effective for segmenting sub-cortical structures in terms of accuracy and speed. However, such an assumption is not practical in the case of pathological structures, i.e., the number of multiple sclerosis lesions is never known a priori.

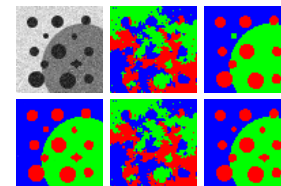
In this paper, we present a generalization which we call the *extended graph-shifts algorithm*. This is able to dynamically create new model instances and hence deal with situations where the number of structures in the image is unknown. Hence we can apply extended graph-shifts to the detection of pathological structures. We formulate these tasks as energy function minimization where statistical learning techniques [12,13] are



**Fig. 1.** An intuitive example of the extended graph-shifts algorithm. It shows a spawn shift being selected (middle panel, double-circle) and then the process of updating the graph hierarchy with the new root-level model node (right panel).

used to learn the components of the energy functions. As we will show, extended graph-shifts is also a computationally efficient algorithm and yields good results on the detection of brain tumors and multiple sclerosis lesions.

The hierarchy is structured as a set of nodes on a series of layers. The nodes at the bottom layer form the image lattice. Each node is constrained to have a single parent. All nodes are assigned a model label which is required to be the same as its parent's label. There is a neighborhood structure defined at all layers of the graph. A *graph shift* is a transformation of the hierarchical structure and thus, the model labeling on the image lattice. There are two types of graph-shifts: (1) changing the parent of a node to the parent of a neighbor with a different model label thus altering the model label of the node and its descendants, and (2) spawning a new sub-graph from a node to the root-level that creates a new model instance. We refer the reader to [11] for a discussion of the first type of shift and restrict the discussion in this paper to the new spawn shift. The spawn shift is illustrated in figure 1, which shows a node being selected to spawn a new instance of model  $m_2$  and then the creation of the new root-level model node. Figure 2 shows a synthetic example comparing the original graph-shifts with the extended algorithm to demonstrate the importance of the spawn-shift to detect small, detached structures. In this case, without spawning, only one of four small structures is detected properly. The extended graph-shifts algorithm minimizes a global energy function and at each iteration selects the shift that maximally decreases the energy.



**Fig. 2.** Extended graph-shifts can detect small structures. Left-col.: image and labels. Middle-col.: initialization. Top-right: no spawning (graph-shifts), bottom-right: with spawning.

We apply this algorithm to brain tumor (*glioblastoma multiforme*, GBM) and multiple sclerosis detection and segmentation. Due to the clinical importance of automatic detection for both diagnosis and treatment, each of these applications has received much attention. Clark et al. [4] integrate knowledge-based techniques and multi-spectral histogram analysis to segment GBM tumors in a multichannel feature space. Corso et al. [5] extend the Segmentation by Weighted Aggregation (SWA) algorithm [3] to integrate Bayesian model classification into the bottom-up aggregation process to rapidly detect GBM tumors. Fletcher-Heath et al. [6] take a fuzzy clustering approach to the segmentation followed by 3D connected components to build the tumor shape. Prastawa et al. [7] present a detection/segmentation algorithm based on learning voxel-intensity distributions for normal brain matter and detecting outlier voxels, which are considered tumor.

Akselrod-Ballin et al. [8] present a sequential approach to segmentation and classification by using the aggregates from the SWA algorithm as features in a decision tree-based classification for multiple sclerosis. Van Leemput et al. [9] and Dugas-Phocion

et al. [10] each use a probabilistic model outlier detection algorithm using a generative i.i.d. model of the normal brain for multiple sclerosis analysis. We next describe the extended graph shifts algorithm, and then we report the experimental results in section (3).

## 2 Extended Graph-Shifts

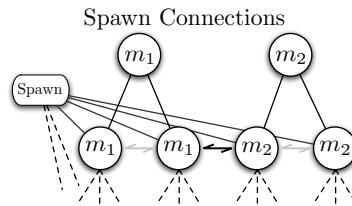
First, we discuss the hierarchical graph structure in section (2.1). Then in section (2.2), we review the recursive energy formulation that makes it possible to evaluate graph shifts at any level in the hierarchy. Finally, we present the extended graph-shifts algorithm in section (2.3), the pseudo-code for which is in figure 4.

### 2.1 The Hierarchical Graph Structure

We define a graph  $G$  to be a set of nodes  $\mu \in \mathcal{U}$  and a set of edges. The graph is hierarchical and composed of multiple layers. The nodes at the lowest layer are the elements of the lattice  $D$  and the edges are defined to link neighbors on the lattice. The coarser layers are computed recursively, as will be described in section (2.3). Two nodes at a coarse layer are joined by an edge if any of their children are joined by an edge.

The nodes are constrained to have a single parent (except for the nodes at the top layer) and every node has at least one child (except for nodes at the bottom layer). We use the notation  $C(\mu)$  for the children of  $\mu$ , and  $A(\mu)$  for the parent. A node  $\mu$  on the bottom layer (i.e. on the lattice) has no children, and hence  $C(\mu) = \emptyset$ . We use the notation  $N(\mu, \nu) = 1$  to indicate that nodes  $\mu, \nu$  on the same layer (or lattice  $D$ ) are neighbors, with  $N(\mu, \nu) = 0$  otherwise.

At the top of the hierarchy, we define a special *root* layer of nodes comprised of a single node for each of the  $K$  model labels. We write  $\bar{\mu}_k$  for these root nodes and use the notation  $m_k$  to denote the model variable associated with it. Each node is assigned a label that is constrained to be the label of its parent. Since, by construction, all non-root nodes can trace their ancestry back to a single root node, an instance of the graph  $G$  is equivalent to a labeled segmentation  $\{m_\mu : \mu \in D\}$  of the image. Finally, we add a *spawning node* which is the neighbor of all nodes in the forest except the root layer nodes. The spawning node can take any model label. It is used to enable a node to switch to any model, and does not make a direct contribution to the energy function. Such a construct is used to simplify the representation of potential shifts: both types of shifts are now simply edges in the graph.



**Fig. 3.** Example of graph-structure including the connections to the spawn node.

### 2.2 The Energy Models

The input image  $\mathbf{I}$  is defined on a lattice  $D$  of pixels/voxels. For the medical image applications being studied this is a three-dimensional lattice. The lattice has the standard 6-neighborhood structure. The task is to assign each voxel  $\mu \in D$  to one of a fixed set of  $K$  models  $m_\mu \in \{1, \dots, K\}$ . This assignment corresponds to a segmentation of the image into  $K$ , or more, connected regions.

We want the segmentation to minimize an energy function criterion:

$$E[\{m_\omega : \omega \in D\}] = \sum_{\nu \in D} E_1(\phi(\mathbf{I})(\nu), m_\nu) + \frac{1}{2} \sum_{\substack{\nu \in D, \mu \in D: \\ N(\nu, \mu)=1}} E_2(\mathbf{I}(\nu), \mathbf{I}(\mu), m_\nu, m_\mu). \quad (1)$$

This energy represents is a hybrid discriminative-generative model and is related to the popular conditional random fields models [14]. The first term  $E_1$  is a *unary term* which gives local evidence that the pixel  $\mu$  takes model  $m_\mu$ , where  $\phi(\mathbf{I})(\mu)$  denotes a nonlinear filter of the image evaluated at  $\mu$ . In this paper, we set  $E_1(\mu, m_\mu) = -\log P(m_\mu | \phi(\mathbf{I})(\mu))$  where  $P(m_\mu | \phi(\mathbf{I})(\mu))$  is the probability distribution for the label  $m_\mu$  at voxel  $\mu \in D$  conditioned on the response of a nonlinear filter  $\phi(\mathbf{I})(\mu)$ . This filter  $\phi(\mathbf{I})(\mu)$  depends on voxels within a local neighborhood of  $\mu$ , and hence takes local image context into account.  $\phi(\mathbf{I})$  is learned from training data from a set of features using boosting techniques [12,13,15]. We discuss the feature-set and learning in more detail in section 3. The second term  $E_2$  is a *pairwise term* which penalizes the length of the segmentation boundaries. It is written as, where  $\delta$  is the standard delta function:  $E_2(\mathbf{I}(\nu), \mathbf{I}(\mu), m_\nu, m_\mu) = 1 - \delta_{m_\nu, m_\mu}$ . We don't use the intensities  $\mathbf{I}(\cdot)$  in the binary term in this formulation but suggest potential variations in [11].

We now recursively assign an energy to all nodes, and neighboring node pairs in the hierarchy. This enables us to rapidly compute the changes in energy caused by extended graph-shifts at any level of the hierarchy, and will be used in the definition of the extended graph-shifts algorithm.

The unary term for assigning a model  $m_\mu$  to a node  $\mu$  is defined recursively by:

$$E_1(\mu, m_\mu) = \begin{cases} E_1(\phi(\mathbf{I})(\mu), m_\mu) & \text{if } C(\mu) = \emptyset \\ \sum_{\nu \in C(\mu)} E_1(\nu, m_\mu) & \text{otherwise} \end{cases} \quad (2)$$

The pairwise energy term  $E_2$  between nodes  $\mu_1$  and  $\mu_2$ , with models  $m_{\mu_1}$  and  $m_{\mu_2}$  is defined recursively by:

$$E_2(\mu_1, \mu_2, m_{\mu_1}, m_{\mu_2}) = \begin{cases} E_2(\mathbf{I}(\mu_1), \mathbf{I}(\mu_2), m_{\mu_1}, m_{\mu_2}) & \text{if } C(\mu_1) = C(\mu_2) = \emptyset \\ \sum_{\substack{\nu_1 \in C(\mu_1), \nu_2 \in C(\mu_2): \\ N(\nu_1, \nu_2)=1}} E_2(\nu_1, \nu_2, m_{\mu_1}, m_{\mu_2}) & \text{otherwise} \end{cases} \quad (3)$$

where  $E_2(\mathbf{I}(\mu_1), \mathbf{I}(\mu_2), m_{\mu_1}, m_{\mu_2})$  is the edge energy for pixels/voxels in equation (1).

### 2.3 Extended Graph-Shifts

We first initialize the graph hierarchy using a stochastic algorithm which recursively coarsens the graph. The coarsening proceeds in three stages. First, we sample a binary edge activation variable  $e_{\mu\nu} \sim \gamma \mathcal{U}(\{0, 1\}) + (1 - \gamma) \exp[-\alpha |\mathbf{I}(\mu) - \mathbf{I}(\nu)|]$ , for all neighbors  $\mu, \nu$  s.t.  $N(\mu, \nu) = 1$  in the current graph layer  $G^t$ . ( $\mathcal{U}$  is the uniform distribution and  $\gamma$  is a fixed weight). Second, we compute connected components to form node-groups. The size of a connected component is constrained by a threshold  $\tau$ , which governs the relative degree of coarsening between two graph layers. Third, we create a

node at the next graph layer for each component. Nodes in this new layer are connected if any two of their children are connected. The algorithm executes this coarsening procedure until the number of node in the layer  $G^T$  falls below a threshold  $\beta$  (related to the number of models). Finally we add a model layer  $G^M$  directly above layer  $G^T$  so that each node in  $G^T$  is the child of its best fit node in  $G^M$  (using the recursive unary energy  $E_1(\mu, m_\mu)$  from (2)). We do not need to enforce the constraint that each node in  $G^M$  has at least one child in  $G^T$  because the new spawn shift makes it possible to create new (connected) nodes on  $G^M$  for any model type.

The extended graph-shifts algorithm enables a node  $\mu$  to change its label to any model. This is an extension of the graph-shifts algorithm [11], which allowed a node to change its model only to the model of one of its neighboring nodes (the children of the node must also change to the same model). This original type of shift is straightforward in the hierarchical structure since it corresponds to allowing node  $\mu$  to change its parent to the parent  $A(\nu)$  of a neighboring node  $\nu$ . In the original graph-shifts algorithm, a list of potential shifts (between nodes having different models only) is maintained. After each shift is taken, this list is quickly updated (the number of updates is logarithmic in the size of the voxel lattice). Thus, graph-shifts can rapidly do the minimization.

In this extension, we permit a node to switch to any other model while maintaining this ability to rapidly and deterministically manage the potential shifts by modifying the graph structure. We create a new *spawning node* to which all nodes in the graph are connected (see figure 3). This spawning node can take the label of any model in the set, and when evaluating the shift on the edge connecting a node to the spawning node, all possible models are evaluated. But, the spawn node contributes nothing to the energy function. A node  $\mu$  taking a spawning graph shift causes a new sub-graph to be dynamically created. The new sub-graph is a chain of nodes from  $\mu$  to the top of the hierarchy, i.e., a new ancestry chain  $A(\mu)$  is created that terminates at a new model level node on  $G^M$ .

Each change of node model will correspond to a change of energy (because the descendant nodes, including the lattice nodes, are also required to make the same change). We need to efficiently compute the change of energy for all nodes and for all changes of models in order to select the best shift to make in the hierarchy. Fortunately, these *shift-gradients* can be computed efficiently using the recursive formulae given above in equations (2) and (3). For example, the change in energy due to node  $\mu$  changing from model  $m_\mu$  to model  $\hat{m}_\mu$  is given by:

$$\Delta E(m_\mu \rightarrow \hat{m}_\mu) = E_1(\mu, \hat{m}_\mu) - E_1(\mu, m_\mu) + \sum_{\eta: N(\mu, \eta)=1} [E_2(\mu, \eta, \hat{m}_\mu, m_\eta) - E_2(\mu, \eta, m_\mu, m_\eta)] . \quad (4)$$

We maintain both the original and the spawning shifts in a single list. Computing the shift gradient is equivalent for both types, but effecting a spawn shift, while still logarithmic in order, has a higher computational cost in creating the new sub-graph. When computing a potential graph-shift, we first evaluate the shift-gradient for all of the node's neighbors. Next, we evaluate the shift-gradient to the spawn node, only considering those models for which there was no neighbor. We store those shifts which have negative gradients in an unsorted list (only the best potential shift is stored per node). The size of this list is generally small (empirically about 2% of those possible), very few neighbors in the graph have different models and a spawn shift more often increases the energy than decreases due to the additional boundary energy cost.

Extended graph-shifts proceeds by selecting the steepest shift-gradient in the list and makes the corresponding shift in the hierarchy. This changes the labels in the part of the hierarchy where the shift occurs, but leaves the remainder of the hierarchy unchanged. If the shift is a spawn, then a new subgraph is dynamically generated. Neighbor connectivity for a newly generated node up the graph is inherited from the node that initiated the spawn. The algo-

rithm recomputes the shift-gradients in the changed part of the hierarchy and updates the weight list. We repeat the process until convergence, when the shift-gradient list is empty (i.e. all shift-gradients are positive or zero). The algorithm tends to initially prefer shifts at coarse levels since those typically alter the labels of many nodes on the lattice and cause large changes in energy. As the algorithm converges, it tends to select shifts at finer levels, but this trend is not monotonic, see examples in [11].

### EXTENDED GRAPH-SHIFTS

**Input:** Volume  $\mathbf{I}$  on lattice  $D$ .  
**Output:** Label volume  $\mathbf{L}$  on lattice  $D$ .  
0 Initialize graph hierarchy.  
1 Compute exhaustive set of potential shifts  $S$ .  
2 **while**  $S$  is not empty  
3      $s \leftarrow$  the shift in  $S$  that best reduces the energy.  
4     Apply shift  $s$  to the graph.  
5     **if** ( $s$  is a spawn)  
6         Build new subgraph and create model.  
7     Recompute affected shifts and update  $S$ .  
8 Compute label volume  $\mathbf{L}$  from final hierarchy.

**Fig. 4.** Extended graph-shifts pseudo-code.

## 3 Detection of Pathological Brain Structures

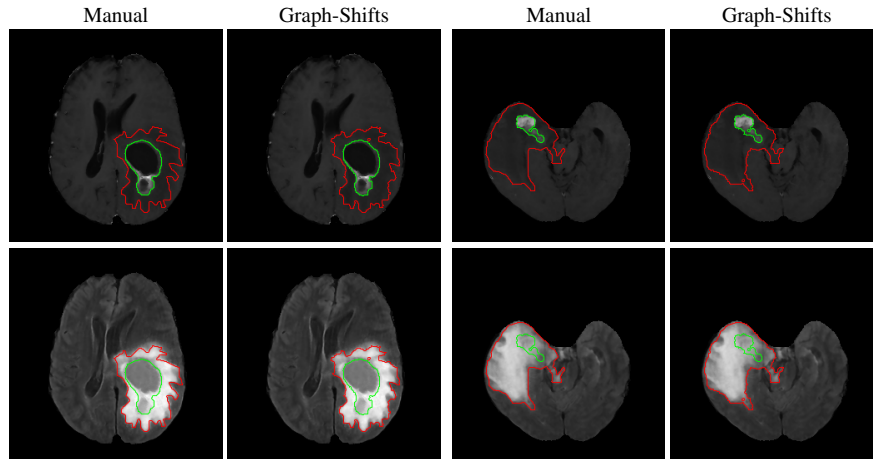
We apply extended graph-shifts to detecting and segmenting brain tumors and multiple sclerosis lesions. For the task of modeling the different pathologies, we build a cascade of boosted discriminative classifiers [12,13,15]. Each classifier in the cascade is trained using a set of about 3000 features including the standard location and Harr-based filters but also some novel features including inter-channel box-filters and gradients, local intensity curve-fitting, and morphology filters. The filters are combined to define the unary term in equation (1). As we demonstrate below, the combination of the boosting-based discriminative modeling and the extended graph-shifts algorithm provides a powerful, general approach to pathology detection and segmentation.

### 3.1 Segmenting Tumors

We work with a set of 20 expert annotated GBM scans. Each scan is comprised of T1 (with and without contrast), Flair and T2 weighted low-resolution MR scans (about  $1 \times 1 \times 10$  on average); this is a common instance for diagnostic brain tumor imaging. All sequences from each patient are co-registered (to the T1 with contrast sequence), skull-stripped, and intensity standardized using the standard FSL tools [16]. We learn models of three separate classes: brain-and-background, tumor (including enhancing and necrotic regions), and edema; half of the dataset was arbitrarily selected for training and the other half for testing.

In table 1(a), we give quantified volume scores for the segmentation accuracy on the dataset. Let  $T$  be the true positive,  $F_p$  be the false positive, and  $F_n$  be the false negative. The Jaccard score is  $T/(T + F_p + F_n)$ , the precision is  $T/(T + F_p)$ , and the recall is  $T/(T + F_n)$ . To the best of our knowledge, these scores are superior to the current state-of-the-art in tumor and edema segmentation. However, we note that a direct comparison is difficult due to different data, manual raters, and others. The Jaccard scores for Clark et al. [4] are about 70%, for Prastawa et al. [7] are 80% (both on very limited datasets with seven and three patients respectively), and Corso et al. [5] is 85% (on training

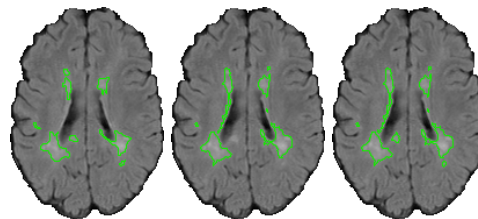
data with of five cases) The extended graph-shifts algorithm is also the fastest among these taking about a minute to perform the segmentation on each of these scans (pre-processing takes about five minutes). We show some examples in figure 5.



**Fig. 5.** An example of the brain tumor segmentation. Left case is from the training set and right case is from the testing set. Edema is outlined in red and tumor in green. Top row is the contrast-enhanced T1 weighted sequence, and bottom row is the flair sequence. The results are obtained automatically. Please view in color.

### 3.2 Detecting MS Lesions

We work with a set of 12 high-lesion-load multi-sequence MR scans. The voxel resolution of these scans is  $1 \times 1 \times 3$ . In this case, we train a two-class discriminative probability model for lesion/not-lesion using the manually annotated dataset. Again, the dataset was split in half for training and testing. Table 1(b) gives the detection rate for the lesion. The detection rate (recall) stresses the importance of picking up most of the lesion mass. Due to its diffuse nature, there is high-variability in expert raters, and detecting each lesion “kernel” is most important. These scores are comparable to the state-of-the-art in automatic lesion detection (#Hit is 80%-85% in [8] and [9] give graphs with varying thresholds showing scores in the 70%*s* through 90%*s*). In figure 6 we show some qualitative results comparing the original graph-shifts to the extended algorithm; the benefit of the spawning functionality is clear from these images. Many smaller lesions are missed by the original algorithm.



**Fig. 6.** Extended graph-shifts detects small lesions. Left: manual, middle: no spawning, right: with spawning. Please view in color.

**Table 1.** Quantified accuracy of the extended graph-shifts algorithm on two pathologies.

(a) Brain Tumor							(b) Multiple Sclerosis	
	Training Set			Testing Set				Lesion
	Jaccard	Precision	Recall	Jaccard	Precision	Recall		Detection Rate
Tumor	87%	93%	92%	86%	95%	90%	Training Set	86%
Edema	87%	90%	96%	88%	89%	98%	Testing Set	81%

## 4 Conclusion

In this paper, we define the extended graph-shifts algorithm for segmenting and labeling image data. Extended graph-shifts is a hierarchical energy minimization algorithm. It has potential application in a broad range of problems where the components of the energy functions can be learned from labeled training data using techniques from statistical learning. This extension generalizes our recent graph-shifts algorithm so that it can now deal with an unknown number of model instances. Hence, the algorithm can be applied to the task of detecting pathological structures, where the number of regions is unknown in advance. Extended graph-shifts retains the advantages in speed and robustness to local minima which were demonstrated for the graph-shifts algorithm. We applied extended graph-shifts to the tasks of detecting brain tumors and multiple sclerosis lesions. Our results were accurate (precision and recall on the order of 93%) and fast (segmenting an entire 3D volume,  $256 \times 256 \times 50$ , in about a minute).

## References

1. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distributions, and Bayesian Restoration of Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **6** (1984) 721–741
2. Zhu, S.C., Yuille, A.: Region Competition. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **18**(9) (1996) 884–900
3. Sharon, E., Brandt, A., Basri, R.: Fast Multiscale Image Segmentation. In: *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*. Volume I. (2000) 70–77
4. Clark, M.C., Hall, L.O., Goldgof, D.B., Velthuizen, R., Murtagh, R., Silbiger, M.S.: Automatic tumor segmentation using knowledge-based techniques. *IEEE Trans. on Medical Imaging* **17**(2) (1998) 187–201
5. Corso, J.J., Sharon, E., Yuille, A.: Multilevel Segmentation and Integrated Bayesian Model Classification with an Application to Brain Tumor Segmentation. In: *Medical Image Computing and Computer Assisted Intervention*. Volume 2. (2006) 790–798
6. Fletcher-Heath, L.M., Hall, L.O., Goldgof, D.B., Reed Murtagh, F.: Automatic segmentation of non-enhancing brain tumors in magnetic resonance images. *Artificial Intelligence in Medicine* **21** (2001) 43–63
7. Prastawa, M., Bullitt, E., Ho, S., Gerig, G.: A brain tumor segmentation framework based on outlier detection. *Medical Image Analysis Journal* **8**(3) (2004) 275–283
8. Akselrod-Ballin, A., Galun, M., Gomori, M.J., Filippi, M., Valsasina, P., Basri, R., Brandt, A.: Integrated Segmentation and Classification Approach Applied to Multiple Sclerosis Analysis. In: *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*. (2006)
9. Leemput, K.V., Maes, F., Vandermeulen, D., Colchester, A., Suetens, P.: Automated Segmentation of Multiple Sclerosis Lesions by Model Outlier Detection. *IEEE Trans. on Medical Imaging* **20**(8) (2001) 677–688
10. Dugas-Phocion, Gonzalez, M.A., Lebrun, C., Channalet, S., Bensa, C., Malandain, G., Ayache, N.: Hierarchical Segmentation of Multiple Sclerosis Lesions in Multi-Sequence MRI. In: *Proc. of the IEEE Intl. Symposium on Biomedical Imaging*. (2004)
11. Corso, J.J., Tu, Z., Yuille, A., Toga, A.W.: Segmentation of Sub-Cortical Structures by the Graph-Shifts Algorithm. In: *Proc. of Information Processing in Medical Imaging*. (2007) 183–197
12. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. *Journal of Computer and System Science* **55**(1) (1997) 119–139
13. Tu, Z.: Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering. In: *Proc. of International Conference on Computer Vision*. (2005)
14. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: *Proc. of International Conference on Machine Learning*. (2001)
15. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. (2001)
16. Smith, S.M., Jenkinson, M., Woolrich, M.W., Beckmann, C.F., Behrens, T.E.J., Johansen-Berg, H., Bannister, P.R., Luca, M.D., Drobnjak, I., Flitney, D.E., Niazy, R., Saunders, J., Vickers, J., Zhang, Y., Stefano, N.D., Brady, J.M., Matthews, P.M.: Advances in Functional and Structural MR Image Analysis and Implementation as FSL. *NeuroImage* **23**(S1) (2004) 208–219

---

**Acknowledgement:** This work was funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 RR021813 entitled Center for Computational Biology (CCB).