Label fusion based multi-atlas segmentation has proven to be one of the most competitive techniques for medical image segmentation. This technique transfers segmentations from expert-labeled images, called atlases, to a novel image using deformable image registration. Errors produced by label transfer are further reduced by label fusion that combines the results produced by all atlases into a consensus solution. Among the proposed label fusion strategies, weighted voting with spatially varying weight distributions derived from atlas-target intensity similarity is a simple and highly effective label fusion technique. However, one limitation of most weighted voting methods is that the weights are computed independently for each atlas, without taking into account the fact that different atlases may produce similar label errors. To address this problem, we recently developed the joint label fusion technique and the corrective learning technique, which won the first place of the 2012 MICCAI Multi-Atlas Labeling Challenge and was one of the top performers in 2013 MICCAI Segmentation: Algorithms, Theory and Applications (SATA) challenge. To make our techniques more accessible to the scientific research community, we describe an Insight-Toolkit based open source implementation of our label fusion methods. Our implementation extends our methods to work with multi-modality imaging data and is more suitable for segmentation problems with multiple labels. We demonstrate the usage of our tools through applying them to the 2012 MICCAI Multi-Atlas Labeling Challenge brain image dataset and the 2013 SATA challenge canine leg image dataset. We report the best results on these two datasets so far.

Keywords: multi-atlas label fusion, joint label fusion, corrective learning, Insight-Toolkit, open source implementation

1. INTRODUCTION

Image segmentation is often necessary for quantitative medical image analysis. In most applications, manual segmentation labeled by human expert is treated as the gold standard. However, due to the high labor intensive nature of manual segmentation and its poor reproducibility, it is often desirable to have accurate automatic segmentation techniques to replace manual segmentation.

As an intuitive solution for applying manually labeled images to segment novel images, atlas-based segmentation (Rohlfing et al., 2005) has been widely applied in medical image analysis. This technique applies example-based knowledge representation, where the knowledge for segmenting a structure of interest is represented by a pre-labeled image, called an atlas. Through establishing one-to-one correspondence between a target novel image and an atlas image by image-based deformable registration, the segmentation label can be transferred to the target image from the atlas.

Segmentation errors produced by atlas-based segmentation are mostly due to registration errors. One effective way to reduce such errors is through employing multiple atlases. When multiple atlases are available, each atlas produces one candidate segmentation for the target image. Under the assumption that segmentation errors produced by different atlases are not identical, it is often feasible to derive more accurate solutions by label fusion. Since the example-based knowledge representation and registration-based knowledge transfer scheme can be effectively applied in many biomedical imaging problems, label fusion based multi-atlas segmentation has produced impressive automatic segmentation performance for many applications (Rohlfing et al., 2004; Isgum et al., 2009; Collins and Pruessner, 2010; Asman and Landman, 2012; Wang et al., 2013a). For some most studied brain image segmentation problems, such as hippocampus segmentation (Wang et al., 2011) and hippocampal subfield segmentation (Yushkevich et al., 2010), automatic segmentation performance produced by multi-atlas label fusion has reached the level of inter-rater reliability.

Weighted voting with spatially varying weight distributions derived from atlas-target intensity similarity is a simple and highly effective label fusion technique. However, most weighted voting methods compute voting weights independently for each atlas, without taking into account the fact that different atlases may produce similar label errors. To address this problem, we developed the joint label fusion technique (Wang et al., 2013b) and the corrective learning technique (Wang et al., 2011). To make our techniques more accessible to the scientific research community, we describe an Insight-Toolkit based implementation of our label fusion methods. Our work has the following
novel contributions. First, we extend our label fusion techniques to work with multi-modality imaging data and with user designed features. Second, we simplify and improve the efficiency of the corrective learning technique to make it more suitable for segmentation problems with multiple labels. Both theoretical and implementation issues are discussed in detail. We demonstrate the usage of our software through two applications: brain magnetic resonance image (MRI) segmentation using the data from the 2012 MICCAI Multi-Atlas Labeling Challenge (Landman and Warfield, 2012) and canine leg muscle segmentation using the data from 2013 SATA challenge. We report the best segmentation results on these two datasets so far.

2. MATERIALS AND METHODS

2.1. METHOD OVERVIEW

2.1.1. Multi-atlas segmentation with joint label fusion

Let $T_F$ be a target image to be segmented and $A^1 = (A^1_1, A^1_2), \ldots, A^n = (A^n_1, A^n_2)$ be $n$ atlases, warped to the space of the target image by deformable registration. $A^1_i$ and $A^n_i$ denote the $i_{th}$ warped atlas image and manual segmentation. Joint label fusion is a weighted voting based label fusion technique.

Weighted voting is a simple yet highly effective approach for label fusion. For instance, majority voting (Rohlfing et al., 2005; Heckemann et al., 2006) applies equal weights to every atlas and consistently outperforms single atlas-based segmentation. Among weighted voting approaches, similarity-weighted voting strategies with spatially varying weight distributions have been particularly successful (Artaechevarria et al., 2009; Igum et al., 2009; Sabuncu et al., 2010; Yushkevich et al., 2010; Wang et al., 2013b). The consensus votes received by label $l$ are:

$$\hat{p}(l|x, T_F) = \sum_{i=1}^{n} w^i_x p(l|x, A^i) \tag{1}$$

where $\hat{p}(l|x, T_F)$ is the estimated probability of label $l$ for the target image at location $x$. $p(l|x, A^i)$ is the probability that $A^i$ votes for label $l$ at $x$, with $\sum_{l=1}^{L} p(l|x, A^i) = 1$. $L$ is the total number of labels. Note that for determinstic atlases that have one unique label for every location, $p(l|x, A^i)$ degenerates into an indicator function, i.e., $p(l|x, T_F) = I(T_S(x) = l)$ and $p(l|x, A^i) = I(A^i_l(x) = l)$, where $T_S$ is the unknown segmentation for the target image. $w^i_x$ is the voting weight for the $i_{th}$ atlas, with $\sum_{i=1}^{n} w^i_x = 1$.

2.1.1.1. The joint label fusion model Wang et al., 2013b.

For deterministic models, we model segmentation errors produced by each warped atlas as $\delta^i(x) = I(T_S(x) = l) - I(A^i_l(x) = l)$. Hence, $\delta^i(x) \in \{-1, 0, 1\}$ is the observed label difference. The correlation between any two atlases in producing segmentation errors at location $x$ are captured by a dependency matrix $M_x$, with $M_x(i, j) = p(\delta^i(x)\delta^j(x) = 1 | T_F, A^i, A^j)$ measuring the probability that atlas $i$ and $j$ produce the same label error for the target image. The expected label difference between the consensus solution obtained from weighted voting and the target segmentation is:

$$E_{\delta^i(x), \ldots, \delta^j(x)} \left[ \left( I(T_S(x) = l) - \sum_{i=1}^{n} w^i_x I(A^i_l(x) = l) \right)^2 \right] \quad \text{for } T_F, A^1, \ldots, A^n$$

where $t$ stands for transpose. To minimize the expected label difference, the optimal voting weights can be solved by $w^i_x = M_x^{-1}I_{i\cdot}n$, where $I_{i\cdot} = [1; 1; \ldots; 1]$ is a vector of size $n$. To avoid inverting an ill-conditioned matrix, we always add an identity matrix weighted by a small positive number $\alpha$ to $M_x$.

The key difference between joint label fusion and other label fusion methods is that it explicitly considers correlations among atlases, i.e., the dependency matrix, into voting weight assignment to reduce bias in the atlas set. In the extreme example, if one of the atlases in the atlas set is replicated multiple times, the combined weight assigned to all replicates of the atlas would be the same as when the atlas is included only once. This is in contrast to earlier weighted voting label fusion methods (Artaechevarria et al., 2009; Sabuncu et al., 2010), in which the weight assigned to the replicated atlas increases with the number of replicates. More generally, the weights assigned by joint label fusion to anatomical patterns in the atlases are not biased by the prevalence of those patterns in the atlas set.

2.1.1.2. Estimation of the pairwise atlas dependency matrix $M_x$.

Since the segmentation of the target image is unknown, we apply an image similarity based model over local image patches to estimate $M_x$ as follows:

$$M_x(i, j) = \left[ \sum_{d=1}^{D} |A^d_F(N(x)) - T_S^d(N(x))|, |A^d_F(N(x)) - T_F^d(N(x))| \right]^{\beta} \tag{3}$$

where $\beta$ is a model parameter. Note that if the off-diagonal elements in $M_x$ are set to zeros, the voting weights derived from $M_x$ is equivalent to the local weighted voting approach with the inverse distance weighting function as described in Artaechevarria et al. (2009). In this simplified case, $\beta$ has a more straightforward interpretation that controls the distribution of voting weights. Large $\beta$ will produce more sparse voting weights and only the atlases that are most similar to the target image contribute to the consensus solution. Similarly, small $\beta$ will produce more uniform voting weights.

To make the measure more robust to image intensity scale variations across different images, we normalize each image intensity patch to have zero mean and a unit variance before estimating $M_x$.

2.1.1.3. The local search algorithm. To make label fusion more robust against registration errors, we apply a local search...
algorithm to find the patch from each warped atlas within a small neighborhood $N_s(x)$ that is the most similar to the target patch in the target image. Under the assumption that more similar patches are more likely to be correct correspondences, instead of the original corresponding patches in the warped atlases, the searched patches are applied for label fusion.

We determine the local search correspondence map between the atlas $i$ and the target image as follows:

$$\xi_i(x) = \arg\min_{x' \in N_s(x)} \left\| A^i_F (N(x')) - T_F (N(x)) \right\|^2,$$  \hspace{1cm} (4)

Note that the domain of the minimization above is restricted to a neighborhood $N_s(x)$. Given the set of local search correspondence maps $\{\xi_i\}$, we refine the definition of the consensus segmentation as:

$$\hat{\rho}(l|x, T_F) = \sum_{i=1}^{n} w_i(\xi_i(x)) \rho(l|\xi_i(x), A^i),$$ \hspace{1cm} (5)

The local search algorithm compares each target image patch with all patches within the searching neighborhood in each warped atlas. Normalizing image patches within the search neighborhood can be an expensive operation. To make the algorithm more efficient, we make the following observation. Let $X$ and $Y$ be vectors storing the original intensity values for two image patches. Let $x$ and $y$ be the normalized vector for $X$ and $Y$, respectively. Let $\bar{Y} = \sum_{i=1}^{k} Y(i)/k$ and $\sigma(Y) = \left[ \frac{\sum_{i=1}^{k} (Y(i) - \bar{Y})^2}{k} \right]^{0.5}$ be the mean and standard deviation for $Y$, where $k$ is the vector size of $Y$. Hence, $y(i) = \frac{Y(i) - \bar{Y}}{\sigma(Y)}$. To compute the sum of squared distance between $x$ and $y$, we have:

$$\sum_{i=1}^{n} \left[ x(i) - y(i) \right]^2 = \sum_{i=1}^{n} \left[ x(i) - \frac{Y(i) - \bar{Y}}{\sigma(Y)} \right]^2$$

$$= \frac{1}{\sigma(Y)^2} \sum_{i=1}^{k} \left[ Y(i)^2 - \bar{Y}^2 + x(i)^2 \sigma^2(Y) - 2x(i)Y(i) \sigma(Y) + 2x(i) \bar{Y} \sigma(Y) \right]$$

$$= k + 1 - \frac{2}{\sigma(Y)} \sum_{i=1}^{k} x(i)Y(i)$$ \hspace{1cm} (7)

Equation (8) is obtained from the fact that $\sum_{i=1}^{k} \left[ Y(i)^2 - \bar{Y}^2 \right] = k \sigma(Y)^2$, $\sum_{i=1}^{k} x(i)^2 = 1$, and $\sum_{i=1}^{k} x(i) = 0$. Hence, to make the local search algorithm more efficient, we only need to normalize the target image patch and search the patch in the warped atlas that minimizes $-\frac{1}{\sigma(Y)} \sum_{i=1}^{k} x(i)Y(i)$. Efficiency is achieved by avoiding the normalization operation for atlas patches during local search.

Note that, similar to the non-local mean patch based label fusion approach Coupe et al. (2011), employing all patches within the searching neighborhood for estimating the pairwise atlas dependencies produces more accurate estimation.
examined by the classifier. If the classifier believes that a voxel was mislabeled, its label is changed. In case of more than two labels, corrective learning needs to learn additional classifiers, as detailed below.

Note that machine learning is commonly used for image segmentation in computer vision (Kumar and Hebert, 2003; Tu and Bai, 2010) and medical image analysis (Tu et al., 2007; Morra et al., 2009; Tu and Bai, 2010). Typically, classifiers assigning labels to image voxels are trained and applied purely based on features extracted from images. By contrast, corrective learning allows the learning algorithm to benefit from the domain-specific knowledge captured by the host segmentation method. For instance, a host segmentation method may represent domain-specific knowledge in the form of shape priors and priors on spatial relations between anatomical structures. Corrective learning allows such high-level domain-specific knowledge to be incorporated into the learning process efficiently by using the segmentation results produced by the host method as an additional contextual feature (see more details below).

### 2.1.2.1. Implementation.

In (Wang et al., 2011), we developed two corrective learning algorithms: explicit error correction (EEC) and implicit error correction (IEC). First, we define a working region of interest (ROI) to be derived from performing a dilation operation to the set of voxels assigned to non-background labels by the host method. Each voxel in the working ROI of each training image serves as a sample for training the corrective learning classifiers. The motivation for using the working ROI is that when the host method works reasonably well, most voxels labeled as foreground are in the close proximity of the foreground voxels in the manual segmentation. Hence, using a working ROI simplifies the learning problem by excluding most irrelevant background voxels from consideration.

In binary segmentation problems, IEC is equivalent to EEC. In a problem with \( L > 2 \) labels, EEC uses all voxels in the working ROI to train a single “error detection” classifier, whose task is to identify the voxels mislabeled by the host method. EEC then uses the voxels mislabeled by the host method to train \( L \) “error correction” classifiers, whose task is to reassign labels to the voxels identified as mislabeled by error detection. Each error correction classifier is designed to detect voxels that should be assigned each target label. To reassign labels to a voxel, it is evaluated by all \( L \) error correction classifiers, and the label whose classifier gives the highest response is chosen. By contrast, IEC treats all voxels within the working ROI as mislabeled and directly trains \( N \) error correction classifiers to reassign labels. In principle, EEC is more efficient than IEC for multi-label segmentation because IEC trains \( N \) error correction classifiers using all voxels in the working ROI, while EEC only uses a subset of voxels to train those correction classifiers. On the other hand, IEC has the advantage of not affected by incorrect error detection results.

To make corrective learning more efficient and more effective for segmentation problems with multiple labels, we implemented a third hybrid error correction strategy that combines the advantage of both EEC and IEC. This error correction strategy aims at problems with large numbers of labels by incorporating the prior knowledge that when a host method works reasonably well, most voxels assigned by the host method to a foreground label are in the close proximity of the voxels manually assigned that label. To improve the efficiency of IEC, we propose to restrict error correction for any foreground label only within the label’s working ROI, derived by performing dilation to the set of all voxels assigned the label by the host method. To apply these trained classifiers to correct segmentation errors for a testing image, we apply each classifier to evaluate the confidence of assigning the corresponding label to each voxel within the label’s working ROI. If a voxel belongs to the ROI of multiple labels, the label whose classifier gives the maximal response at the voxel is chosen for the voxel. Since error detection is not explicitly performed, our current implementation is simplified compared to the EEC algorithm. Furthermore, the implemented error correction strategy is not affected by incorrect error detection results. Compared with the IEC algorithm, our implementation is more efficient and more effective as only a small portion of the data, which are also more relevant to the problem of classifying the target label, are used to train the classifier for each label.

Note that the above label’s working ROI definition has one limitation. If a host segmentation method fails to produce some segmentation labels, then the algorithm cannot recover the missing labels. To address this problem, we allow a second approach to define a label’s working ROI by using a predefined ROI mask. If a ROI mask is provided for a label, the label’s ROI is obtained from performing a dilation operation to the set of voxels in the mask. In principle, the ROI mask should cover most voxels of the target label. One way to define ROI masks for missing labels produced by the host method is to use the ROI of labels whose working ROIs cover most voxels manually assigned to the missing label. The union of these labels’ working ROIs can be defined as the missing label’s working ROI.

### 2.1.2.2. Features.

Typical features that can be used to describe each voxel for the learning task include spatial, appearance, and contextual features. The spatial features are computed as the relative coordinate of each voxel to the ROI’s center of mass. The appearance and contextual features are directly derived from the voxel’s neighborhood image patch from the training image and the initial segmentation produced by the host method, respectively. To enhance the spatial correlation, the joint spatial-appearance and joint spatial-contextual features are also included by multiplying each spatial feature with each appearance and contextual feature, respectively. To include other feature types, one can compute features for each voxel and store the voxel-wise feature response into a feature image, i.e., the intensity at each voxel in the feature image is the feature value at that voxel. Passing these feature images to the algorithm, as shown below, will allow these features to be used in corrective learning.

Note that the above patch based features are not rotation or scale invariant. Hence, they are only suitable for images that have similar orientations and scales. Since many medical images, e.g., MRI and CT, are acquired under constrained rotations and scales, these features are often adequate in practice. For problems that do have large rotation and scale variations, one should apply more suitable features.

### 2.1.2.3. Subsampling for large training dataset.

For large data sets, it is not always possible to include all voxels within a label’s
working ROI for learning its classifier due to the memory constraint. For such cases, a subsampling strategy can be applied to randomly select a portion of training voxels according to a specified sampling percentage.

2.1.2.4. Parameter summary. The corrective learning technique has three primary parameters:

- $r_d$: the radius for the dilation operation for defining each label’s working ROI.
- $r_f$: the radius defining the image patch for deriving voxel-wise features.
- SampleRatio: the portion of voxels within the label’s working ROI to be used for learning the classifier for the label.

2.1.2.5. Corrective learning user interface. We separately implemented the algorithm for learning corrective classifiers and the algorithm applying these classifiers for making corrections. We name the program for learning corrective classifiers as $bl$, which stands for bias learning as it learns classifiers that capture the systematic errors, or bias, produced by an automatic segmentation algorithm. We name the program for making corrections as $sa$, which stands for segmentation adapter because it adapts the segmentation produced by the host method to be closer to the desired gold standard. These two programs have the following user interface.

Corrective Learning

usage:

```
bl dim [options] AdaBoost_Prefix
```

required options:

- dim Image dimension (2 or 3)
- ms Manual Segmentation
- as Automatic segmentation
- tl The target label
- rd Dilation radius
- rf Feature radius
- rate Training data sampling rate
- i Number of AdaBoost training iterations

other options:

- c Number of feature channels
- f Feature images
- m ROI mask

Segmentation correction

usage:

```
sa input_segmentation AdaBoost_Prefix
```

output_segmentation [options]

options:

- f Feature images
- p Output posterior maps
- m ROI mask

2.2. APPLICATION 1: BRAIN MRI SEGMENTATION

To demonstrate the usage of the joint label fusion and corrective learning software, we provide implementation details for two applications: whole brain parcellation and canine leg muscle segmentation using MR images. In this section, we describe our application for brain segmentation. The software used in our experiments will be distributed through the Advanced Normalization Tools (ANTS) package Avants et al. (2008) and at http://www.nitrc.org/projects/picsl_malf.

2.2.1. Data and manual segmentation

The dataset used in this study includes 35 brain MRI scans obtained from the OASIS project. The manual brain segmentations of these images were produced by Neuromorphometrics, Inc. (http://Neuromorphometrics.com/) using the brain-COLOR labeling protocol. The data were applied in the 2012 MICCAI Multi-Atlas Labeling Challenge and can be downloaded at (https://masi.vuse.vanderbilt.edu/workshop2012/index.php/Main_Page). In the challenge, 15 subjects were used as atlases and the remaining 20 images were used for testing.

2.2.2. Image registration

To apply our algorithms, we need pairwise registered transformations between each atlas and each target image and between each pair of atlas images. To facilitate comparisons with other label fusion algorithms, we applied the standard transformations provided by the challenge organizers. For the brain image data, the standard transformations are produced by the ANTs registration tool and can be downloadable at http://placid.nlm.nih.gov/user/48. To generate warped images from the transformation files, we applied antsApplyTransforms with linear interpolation. To generate warped segmentations, we applied antsApplyTransforms with nearest neighbor interpolation.

2.2.3. Joint Label Fusion

The following command demonstrates how to apply jointfusion to segment one target image, i.e., subject 1003_3.

```
./jointfusion 3 1 -g ./warped/*_to_1003_3_ image.nii.gz \
-1 warped/*_to_1003_3_ seg_NN.nii.gz \
-m Joint[0.1,2] \
-rp 2x2x2 \
-rs 3x3x3 \
-tg ./Testing/1003_3.nii.gz \
-p ./malf/1003_3_Joint_ posterior%04d.nii.gz \
./malf/1003_3_Joint.nii.gz
```

In this application, only one MRI modality is available. Hence, mod=1. The folder warped stores the warped atlases for each target image. We set the following parameters for jointfusion: $a = 0.1, b = 2$ and isotropic neighborhoods with radius two and three for $r_p$ and $r_s$, respectively. These parameters were chosen because they are optimal for segmenting the hippocampus in our previous study (Wang et al., 2013b). In addition to producing the
consensus segmentation for the target subject, we also saved the posterior probabilities produced by label fusion for each anatomical label as images. These posterior images were applied as an additional feature for corrective learning, as described below. Note that we specify the file name of the output posterior images by the C printf format such that one unique posterior image is created for each label. For instance, for label 0 and 4, the generated posterior images are ./malf/1003_3_JointLabel_posterior 0000.nii.gz and ./malf/1003_3_JointLabel_posterior0004.nii.gz, respectively.

To quantify the performance of jointfusion with respect to the four primary parameters, we also conducted the following leave-one-out cross-validation experiments using the training images. To test the impact of the appearance window size \( r_p \), we varied \( r_p \) from 1 to 3 and fixed \( r_s = 3, \beta = 2, \beta = 0.1 \). To test the impact of the local search window size, we varied \( r_l \) from 0 to 4 and fixed \( r_p = 2, \beta = 2, \beta = 0.1 \). We also varied \( \beta \) from 0.5 to 3 with a 0.5 step and fixed \( r_p = 2, r_s = 3, \alpha = 0.1 \). Finally, we fixed \( r_p = 2, r_s = 3, \beta = 2 \) and tested with \( \alpha = 0, 0.01, 0.05, 0.1, 0.2 \). For experiments testing the effects of \( r_p \) and \( r_s \), we report both computational time and segmentation accuracy for each parameter setting. Since varying \( \beta \) and \( \alpha \) does not have significant impact on computational complexity, we only report segmentation accuracy for each parameter setting.

### 2.2.4. Corrective Learning

To apply corrective learning, we first applied joint label fusion with the above chosen parameters, i.e., \((\alpha, \beta, r_p, r_s) = (0.1, 2, 2, 3)\), to segment each atlas image using the remaining atlases. With both manual segmentation and segmentation produced by joint label fusion, the atlases were applied for training the corrective learning classifiers. Recall that one classifier needs to be learned for each anatomical label. The following command trains the classifier for label 0, i.e., the background label.

```
/bl 3 -ms Training/*_glm.nii.gz 
  -as ./malf/1000_3_Joint.nii.gz 
    ... ./malf/1036_3_Joint.nii.gz 
  -tl 0 
  -rd 1 
  -i 500 
  -rate 0.1 
  -rf 2x2x2 
  -c 2 
  -f ./Training/1000_3.nii.gz 
    ./malf/1000_3_Joint_posterior 0000.nii.gz 
    ... ./Training/1003_3.nii.gz 
    ./malf/1003_3_Joint_posterior 0000.nii.gz 
    ./malf/BL/Joint_BL
```

We applied two feature images. In addition to the original intensity image, we also included the label posteriors generated by jointfusion for corrective learning. As we show in (Wang and Yushkevich, 2012), weighted voting based label fusion produces a spatial bias on the generated spatial label posteriors, which can be modeled as applying a spatial convolution on the ground truth label posteriors. Hence, the label posteriors produced by joint label fusion offers meaningful information for correcting such systematic errors. We set the dilation radius to be \( r_d = 1 \), which was shown to be optimal for correcting segmentation errors produced by multi-atlas label fusion for hippocampus segmentation in our previous study (Wang et al., 2011). For this learning task, a 10 percent sampling rate is applied.

We use the following command to apply the learned classifiers to correct segmentation errors for one testing image.

```
./sa ./malf/1003_3_Joint.nii.gz 
  ./malf/BL/Joint_BL 
  ./malf/1003_3_Joint_CL.nii.gz 
  -f ./Testing/1003_3.nii.gz ./malf/1003_3_Joint_posterior%04d.nii.gz
```

Again, we used the C printf format to specify the file name of label posterior images as feature images.

Since we have shown in our previous work (Wang et al., 2011) that corrective learning is not sensitive to the dilation radius parameter. Here, we only conducted experiments to test the effect of the feature patch size \( r_f \) on the performance. We tested using \( r_f = 1 \) and \( r_f = 3 \) with the same dilation radius.

### 2.2.5. Evaluation

To facilitate comparisons with other work, we follow the challenge evaluation criteria and evaluate our results using the Dice Similarity Coefficient (DSC) (Dice, 1945) between manual and automatic segmentation. DSC measures the ratio of the volume of overlap between two segmented regions and their average volume. For the brain image data, the results were evaluated based on 134 labels, including 36 subcortical labels and 98 cortical labels (see https://masi.vuse.vanderbilt.edu/workshop2012/index.php/Challenge_Details for details of the evaluation criterion). We separately report summarized results for all labels, cortical labels and subcortical labels. To give more information, we also report segmentation performances for nine subcortical structures, including accumbens area, amygdala, brain stem, caudate, cerebral white matter, CSF, hippocampus, putamen, and thalamus proper. For the canine leg data, evaluation was performed over all labels.

### 2.2.6. Results

Using 15 atlases, jointfusion segments one image in about 1 h using a single core 2GHZ CPU with the parameter setting, \( r_p = 2, r_s = 3 \). Applying corrective learning to correct segmentation errors for an image can be done within a few minutes. Figure 1 shows some segmentation results produced by each method.

Table 1 reports the segmentation performance for majority voting, joint label fusion, and joint label fusion combined with correction learning. Joint label fusion produced an average DSC 0.757 for all labels, 0.732 for cortical labels, and 0.825
for subcortical labels. Corrective learning improved the results to 0.771, 0.747, and 0.836, respectively.

**Figures 2, 3** show the average processing time and average segmentation accuracy produced by joint label fusion with respect to $r_p$ and $r_s$, respectively. As expected, the processing time grows proportionally with respect to the neighborhood size. The performance of joint label fusion is not sensitive to the size of appearance patch $r_p$, with the best performance produced by $r_p = 2$. In contrast, the local search algorithm produced more prominent improvement. Although applying larger searching neighbor consistently produced higher averaged DSC, applying $r_s = 1$ produced the greatest improvement. Further increasing $r_s$ only slightly improved the segmentation accuracy.

**Figure 4** shows the segmentation accuracy produced by joint label fusion using different $\beta$ values. For this application, the performance of joint label fusion is not sensitive to $\beta$. Among the tested $\beta$ values, $\beta = 1.5$ produced the best segmentation accuracy.

**Figure 5** shows the segmentation accuracy produced by joint label fusion with respect to $\alpha$. Adding the conditioning matrix, i.e., $\alpha > 0$, produced prominent improvement over without adding the conditioning matrix, i.e., $\alpha = 0$. When the conditioning matrix is added, setting $\alpha$ between 0.01 and 0.2 has a slight impact on the performance, with the best performances achieved at $\alpha = 0.05$ or 0.1.

**Figure 6** shows the segmentation performance produced by corrective learning with respect to feature patch radius. Again, we did not observe large performance variation. The performance produced by radius 2 is slightly better than those produced with radius 1 and 3.

### 2.3. APPLICATION 2: CANINE LEG MUSCLE SEGMENTATION

#### 2.3.1. Data and manual segmentation

The dataset used in this study contains 45 canine leg MR scans. For each dog, images were acquired with two MR modalities: a T2-weighted image sequence was acquired using a variable-flip-angle turbo spin echo (TSE) sequence and a T2-weighted fat-suppressed images (T2FS) sequence was then acquired using the same variable-flip-angle TSE sequence with the same scanning parameters except that a fat saturation preparation was applied. Seven proximal pelvic limb muscles were manually segmented: cranial sartorius, rectus femoris, semitendinosus, biceps femoris, gracilis, vastus lateralis and adductor magnus. In the challenge, 22 subjects were used as atlases and the remaining 23 subjects were used for testing. We will use this dataset for validating the multi-modality extension to our joint label fusion algorithm.

#### 2.3.2. Image registration

For this challenge, we produced the standard registration using ANTs, which can be downloaded...
at https://masi.vuse.vanderbilt.edu/workshop2013/index.php/Segmentation_Challenge_Details. Avants et al. (2013) contains details for how the registrations were generated. To quantify the accuracy of the standard transformations, we applied majority voting to generate a baseline segmentation performance.

2.3.3. Joint label fusion

The following command demonstrates how to apply jointfusion to segment one target image, i.e., subject DD_039, using both MR modality channels.

```
./jointfusion 3 2 -g ./canine-lege-warped/DD_040_to_DD_039_T2.nii.gz
```

Note that, for this application, we applied $\beta = 0.5$. This parameter was chosen because it produced the optimal results for the cross-validation experiments on the training images as described below. To compare with the performance produced by

---

**Table 1 | Segmentation performance in Dice Similarity Coefficient (\(\frac{2|A\cap B|}{|A|+|B|}\)).**

| Anatomical region            | Majority voting | Joint label fusion | Joint label fusion+ Corrective learning |
|------------------------------|-----------------|--------------------|----------------------------------------|
| All labels                   | 0.726 ± 0.138   | 0.757 ± 0.133      | 0.771 ± 0.131                          |
| Cortical labels              | 0.701 ± 0.113   | 0.732 ± 0.114      | 0.747 ± 0.115                          |
| Subcortical labels           | 0.796 ± 0.171   | 0.825 ± 0.154      | 0.836 ± 0.149                          |
| Left accumbens area          | 0.776 ± 0.074   | 0.804 ± 0.053      | 0.796 ± 0.048                          |
| Right accumbens area         | 0.759 ± 0.086   | 0.798 ± 0.058      | 0.796 ± 0.048                          |
| Left amygdala                | 0.800 ± 0.040   | 0.812 ± 0.032      | 0.815 ± 0.034                          |
| Right amygdala               | 0.808 ± 0.028   | 0.827 ± 0.025      | 0.830 ± 0.024                          |
| Brain stem                   | 0.940 ± 0.009   | 0.943 ± 0.008      | 0.946 ± 0.008                          |
| Left caudate                 | 0.801 ± 0.134   | 0.870 ± 0.101      | 0.881 ± 0.088                          |
| Right caudate                | 0.788 ± 0.122   | 0.865 ± 0.076      | 0.884 ± 0.070                          |
| Left cerebral white matter   | 0.903 ± 0.018   | 0.925 ± 0.019      | 0.937 ± 0.017                          |
| Right cerebral white matter  | 0.906 ± 0.018   | 0.926 ± 0.018      | 0.935 ± 0.019                          |
| CSF                          | 0.723 ± 0.166   | 0.789 ± 0.092      | 0.820 ± 0.074                          |
| Left hippocampus             | 0.831 ± 0.046   | 0.862 ± 0.031      | 0.872 ± 0.023                          |
| Right hippocampus            | 0.830 ± 0.044   | 0.861 ± 0.027      | 0.871 ± 0.022                          |
| Left putamen                 | 0.911 ± 0.029   | 0.915 ± 0.037      | 0.909 ± 0.042                          |
| Right putamen                | 0.909 ± 0.033   | 0.914 ± 0.040      | 0.907 ± 0.043                          |
| Left thalamus proper         | 0.903 ± 0.030   | 0.920 ± 0.014      | 0.921 ± 0.012                          |
| Right thalamus proper        | 0.903 ± 0.031   | 0.921 ± 0.012      | 0.923 ± 0.009                          |

---
using a single modality and by using two modalities, we also applied jointfusion by only using the T2-weighted image. Since \( r_p \) and \( \beta \) have the most impact on the joint label fusion performance, for this application we only conducted experiments to quantify the performance of jointfusion with respect to these two parameters using leave-one-out cross-validation experiments on the training images. We varied \( \sigma \) from 0.5 to 2.5 with a 0.5 step while fixing \( r_p = 2, r_s = 3, \lambda = 0.1 \). We varied \( r_i \) from 0 to 4 while fixing \( r_p = 2, \beta = 0.5, \lambda = 0.1 \).

2.3.4. Corrective Learning

To apply corrective learning, again we first applied joint label fusion with the above chosen parameters, i.e., \((\alpha, \beta, r_p, r_s) = (0.1, 0.5, 2, 3)\), to segment each atlas image using the remaining atlases. With both manual segmentation and segmentation produced by joint label fusion, the atlases were applied for training the corrective learning classifiers. The following command trains the classifier for the background label.

```
./bl 3 -ms ./canine-legs/training-labels/*_seg.nii.gz \
-as ./canine-legs-malf/DD_040_Joint.posterior0000.nii.gz \
-tl 0 \
-rd 1 \
-i 500 \
-rate 0.05 \
-rf 2x2x2 \
-c 3 \
-f ./canine-legs/training-images/DD_040_T2.nii.gz \
./canine-legs/training-images/DD_040_T2FS.nii.gz \
./canine-legs-malf/DD_040_Joint_posterior0000.nii.gz \
...
```

We use the following command to apply the learned classifiers to correct segmentation errors for one testing image.

```
./sa ./canine-legs-malf/DD_039_Joint.nii.gz \
./canine-legs-malf/BL/Joint_BL \
./canine-legs-malf/DD_039_Joint_CL.nii.gz \
-f ./canine-legs/testing-images/DD_039_T2.nii.gz \
./canine-legs/testing-images/DD_039_T2FS.nii.gz \
./canine-legs-malf/DD_039_Joint_posterior\%04d.nii.gz
```

2.3.5. Results

Using 22 atlases and both imaging modalities, jointfusion segments one image in about 1 h using a single core 2GHZ CPU with the parameter setting, \( r_p = 2, r_s = 3 \). Applying corrective learning to correct segmentation errors for an image can be done within 1 min. Figure 7 shows some segmentation results produced by each method.

Figure 8 shows the segmentation accuracy produced by joint label fusion using different \( \beta \) values. The results produced by using a single modality and by using two modalities are given separately. As expected, multi-modality based label fusion did result in substantial performance improvement over using a single modality. For this application, the performance of joint label fusion is more sensitive to \( \beta \) when only one modality is applied. Among the tested \( \beta \) values, \( \beta = 0.5 \) produced the best segmentation accuracy.
Figure 7 shows the segmentations produced by manual segmentation, majority voting, joint label fusion with one imaging modality (JLF-Mod1), and joint label fusion with two imaging modalities (JLF-Mod2).

Figure 8 shows joint label fusion performance with respect to $\beta$ (error bars at ±0.05 standard deviation). Other parameters are set to $r_p = 2$, $r_s = 3$, $\alpha = 0.1$.

Table 2 reports the segmentation performance produced by leave-one-out cross validation using the canine leg muscle training data.

| Anatomical region   | Majority voting | Joint label fusion mod1 | Joint label fusion mod2 |
|--------------------|-----------------|-------------------------|-------------------------|
| All labels         | $0.411 \pm 0.274$ | $0.690 \pm 0.185$       | $0.722 \pm 0.160$       |

3. DISCUSSION

3.1. COMPARISON TO THE STATE OF THE ART

Our algorithms participated in the MICCAI 2012 and 2013 multi-atlas labeling challenge competition. Our results on the canine leg muscle dataset are the best among all 2013 challenge entries for this dataset (see Asman et al., 2013 for more detail). Our results for brain segmentation produced based on the standard registration transforms are better than what we originally produced during the competition (Landman and Warfield, 2012). In the
with respect to the scale of the estimated dependency matrix. According to our experiments, we found that setting the scale of estimated 

dependencies resulted in a solution close to the global label fusion. To make sure that the added conditioning matrix is important for joint label fusion, we found that adding conditioning matrix is necessary for achieving almost optimal performance and keeping joint label fusion 

3.2. PARAMETER SELECTION

We found that both joint label fusion and corrective learning are not sensitive to the parameter setting in this brain MRI segmentation application. However, using large local appearance neighborhood, e.g., \( r_p > 2 \), and large local search neighborhood, \( r_s > 2 \), significantly increase the computational cost. Hence, when computational cost is a limiting factor, one could achieve a good trade off between computational complexity and segmentation performance by choosing proper values for these two parameters. Based on our experiments, setting \( r_p = 1, 2 \) and \( r_s = 1, 2 \) can produce almost optimal performance and keep joint label fusion using 15 atlases within 30 min for whole brain segmentation. For \( \alpha \), the weight for adding the conditioning matrix, we found that adding conditioning matrix is important for joint label fusion. To make sure that the added conditioning matrix is sufficient to avoid inverting an ill-conditioned matrix and the resulting voting weights also give a solution close to the global minimum of the original objective function, \( \beta \) should be chosen with respect to the scale of the estimated dependency matrix \( M_\delta \). According to our experiments, we found that setting \( \beta \approx 1\% \) of the scale of estimated \( M_\delta \) seems to be a good choice. For the model parameter \( \beta \) used in estimating appearance based pairwise atlas dependencies Equation (8), its selection depends on the registration quality produced for the application at hand. Based on our experiments and our previous study (Wang et al., 2013b), we found that when registration can be done in good quality such as brain MRI registration in this study, setting \( \beta \approx 2 \) is optimal. For mitral valve segmentation in ultrasound images (Wang et al., 2013a) and canine leg muscle segmentation, where good image registration is more difficult to produce due to low image quality and greater deformations, we found that setting \( \beta = 1 \) or 0.5 is optimal. Hence, setting \( \beta \) depends more on the application.

Table 3 | Segmentation performance in Dice Similarity Coefficient (\( DSC = \frac{2|A \cap B|}{|A| + |B|} \)) produce for the canine leg muscle testing data.

| Anatomical region       | Majority voting | Joint label fusion + corrective learning |
|-------------------------|-----------------|-----------------------------------------|
| All Labels              | 0.418 ± 0.108   | 0.762 ± 0.098                           |

2012 challenge, applying joint label fusion alone, our results are 0.750 for all labels, 0.722 for cortical labels, and 0.827 for subcortical labels. Combining joint label fusion and corrective learning, we produced the best results in the challenge competition, with 0.765 for all labels, 0.739 for cortical labels, and 0.838 for subcortical labels. In this study, applying joint label fusion alone, our results are 0.757 for all labels, 0.732 for cortical labels, and 0.825 for subcortical labels. Combining joint label fusion with corrective learning, our results are 0.771 for all labels, 0.747 for cortical labels, and 0.836 for subcortical labels. Note that, most improvements in our current study are for cortical labels. Hence, it is reasonable to expect that the standard registration transforms provided by the challenge organizers have better accuracy for the cortical regions than those produced by us during the challenge competition.

As we have applied in paper, one way to determine the optimal parameter settings is based on a leave one out experiment on the atlas set. That is segmenting each atlas using the remaining atlases with different parameter settings, the setting produced the best overall segmentation for all atlases should be chosen. As training classifiers in corrective learning, parameter selection for joint label fusion can be done offline. Hence, no additional burden is added for online label fusion. Similarly, combining corrective learning with multi-atlas label fusion is a natural choice, as no additional training data is need for corrective learning and no significant additional online computational burden is added by applying corrective learning.

3.3. FUTURE WORK

Note that when the host segmentation method produces more accurate solutions, applying corrective learning further improves the overall accuracy. Hence, efforts on improving label fusion and corrective learning can be conducted in parallel. For improving corrective learning, one direction would be to explore more effective features and more effective learning algorithms. As recent studies (Montillo et al., 2011; Zikic et al., 2012) have shown that random forest (Breiman, 2001) is a highly effective learning algorithm for addressing segmentation problems. Hence, replacing AdaBoost with random forest may result in further improvement.

ACKNOWLEDGMENTS

This work was supported by NIH awards AG037376, EB014346.

REFERENCES

Artachevarria, X., Munoz-Barrantia, A., and de Solorzano, C. O. (2009). Combination strategies in multi-atlas image segmentation: application to brain MR data. IEEE Trans. Med. Imaging 28, 1266–1277. doi: 10.1109/TMI.2009.2014372

Asman, A., Akhondi-Asl, A., Wang, H., Tustison, N., Avants, B., Warfield, S. K., et al. (2013). “Miccail 2013 segmentation algorithms, theory and applications (SATA) challenge results summary,” in MICCAI 2013 Challenge Workshop on Segmentation: Algorithms, Theory and Applications. (Springer).

Asman, A., and Landman, B. (2012). “Non-local staple: an intensity-driven multi-atlas rater model,” in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012. Lecture notes in computer science, Vol. 7512, eds N. Ayache, H. Delingette, P. Golland, and K. Mori (Berlin; Heidelberg: Springer), 426–443.

Avants, B., Epstein, C., Grossman, M., and Gee, J. (2008). Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. Med. Image Analysis 12, 26–41. doi: 10.1016/j.media.2007.06.004

Avants, B. B., Tustison, N. J., Wang, H., Asman, A. J., and Landman, B. A. (2013). “Standardized registration methods for the sata challenge datasets,” in MICCAI Challenge Workshop on Segmentation: Algorithms, Theory and Applications (SATA) (Springer).

Breiman, L. (2001). Random forests. Mach. Learn. 45, 5–32. doi: 10.1023/A:1010933404324

Collins, D., and Pruessner, J. (2010). Towards accurate, automatic segmentation of the hippocampus and amygdala from MRI by augmenting ANIMAL with a template library and label fusion. Neuroimage 52, 1355–1366. doi: 10.1016/j.neuroimage.2010.04.193

Coupe, P., Manjon, J., Fonov, V., Pruessner, J., Robles, N., and Collins, D. (2011). Patch-based segmentation using expert priors: application to hippocampus and ventricle segmentation. Neuroimage 54, 940–954. doi: 10.1016/j.neuroimage.2010.09.018
Dice, L. (1945). Measure of the amount of ecological association between species. Ecology 26, 297–302. doi: 10.2307/1932409

Freund, Y., and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55, 119–139. doi: 10.1006/jcss.1997.1504

Heckemann, R., Hajnal, J., Aljabar, P., Rueckert, D., and Hammers, A. (2006). Automatic anatomical brain MRI segmentation combining label propagation and decision fusion. Neuroimage 33, 115–126. doi: 10.1016/j.neuroimage.2006.05.061

Isgum, I., Staring, M., Rutten, A., Prokop, M., Viergever, M., and van Ginneken, B. (2009). Multi-atlas-based segmentation with local decision fusion—application to cardiac and aortic segmentation in CT scans. IEEE Transactions on Medical Imaging 28, 1000–1010. doi: 10.1109/TMI.2008.211480

Kumar, S., and Hebert, M. (2003). “Discriminative random fields: a discriminative framework for contextual interaction in classification,” in Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, Vol. 2. 1150–1157.

Landman, B., and Warfield, S. (eds.). (2012). MICCAI 2012 Workshop on Multi-Atlas Labeling. Nice: CreateSpace

Montillo, A., Shotton, J., Winn, J., Iglesias, J., Metaxas, D., and Criminisi, A. (2011). “Entangled decision forests and their application for semantic segmentation of ct images,” in Information Processing in Medical Imaging. Lecture notes in computer science, Vol. 6801, eds G. Székely, G., and H. Hahn (Berlin; Heidelberg: Springer): 184–196.

Morra, J., Tu, Z., Apostolova, L., Green, A., Toga, A., and Thompson, P. (2009). Comparison of Adaboost and support vector machines for detecting Alzheimer’s Disease through automated hippocampal segmentation. IEEE Trans. Med. Imaging 29, 30–43. doi: 10.1109/TMI.2009.2021941

Rohlfing, T., Brandt, R., Menzel, R., and Maurer, C. (2004). Evaluation of atlas selection strategies for atlas-based image segmentation with application to confocal microscopy images of bee brains. Neuroimage 21, 1428–1442. doi: 10.1016/j.neuroimage.2003.11.010

Rohlfing, T., Brandt, R., Menzel, R., Russakoff, D., and Maurer, C. R. Jr. (2005). “Quo vadis, atlas-based segmentation?” in Handbook of Biomedical Image Analysis, Topics in Biomedical Engineering International Book Series, eds J. Suri, D. Wilson, and S. Laxminarayan (Springer): 435–486.

Sabuncu, M., Yeo, B., Leemput, K. V., Fischl, B., and Golland, P. (2010). A generative model for image segmentation based on label fusion. IEEE Trans. Med. Imaging 29, 1714–1720. doi: 10.1109/TMI.2010.2050897

Tu, Z., and Bai, X. (2010). Auto-context and its application to high-level vision tasks and 3D brain image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 32, 1744–1757. doi: 10.1109/TPAMI.2009.186

Tu, Z., Zheng, S., Yuille, A., Reiss, A., Dutton, R., Lee, A., et al. (2007). Automated extraction of the cortical sulci based on a supervised learning approach. IEEE Trans. Med. Imaging 26, 541–552. doi: 10.1109/TMI.2007.892506

Wang, H., Das, S., Suh, J. W., Altinay, M., Pluta, J., Craige, C., et al. (2011). A learning-based wrapper method to correct systematic errors in automatic image segmentation: consistently improved performance in hippocampus, cortex and brain segmentation. Neuroimage 53, 968–985. doi: 10.1016/j.neuroimage.2011.01.006

Wang, H., Pouch, A., Takabe, M., Jackson, B., Gorman, J., Gorman, R., et al. (2013a). “Multi-atlas segmentation with robust label transfer and label fusion,” in Information Processing in Medical Imaging. Lecture notes in computer science, Vol. 7917, eds J. C. Gee, S. Joshi, K. M. Pohl, W. M. Wells, and L. Zöllei (Berlin; Heidelberg: Springer): 548–559.

Wang, H., Suh, J. W., Das, S., Pluta, J., Craige, C., and Yushkevich, P. (2013b). Multi-atlas segmentation with joint label fusion. IEEE Trans. Pattern Anal. Mach. Intell. 35, 611–623. doi: 10.1109/TPAMI.2012.143

Wang, H., and Yushkevich, P. A. (2012). “Spatial bias in multi-atlas based segmentation,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Providence, RI: IEEE): 909–916.

Yushkevich, P., Wang, H., Pluta, J., Das, S., Craige, C., Avants, B., et al. (2010). Nearly automatic segmentation of hippocampal subfields in in vivo focal T2-weighted MRI. Neuroimage 53, 1208–1224. doi: 10.1016/j.neuroimage.2010.06.040

Zikic, D., Glocker, B., Konukoglu, E., Criminisi, A., Demiralp, C., Shotton, J., et al. (2012). “Decision forests for tissue-specific segmentation of high-grade gliomas in multi-channel mri,” in Medical Image Computing and Computer-Assisted Intervention - MICCAI 2012. Lecture notes in computer science, Vol. 7512, eds N. Ayache, H. Delingette, P. Golland, and K. Mori (Berlin; Heidelberg: Springer): 369–376.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 30 August 2013; accepted: 24 October 2013; published online: 22 November 2013.

Citation: Wang H and Yushkevich PA (2013) Multi-atlas segmentation with joint label fusion and corrective learning—an open source implementation. Front. Neuroinform. 7:27. doi: 10.3389/fninf.2013.00027

This article was submitted to the journal Frontiers in Neuroinformatics.

Copyright © 2013 Wang and Yushkevich. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.