Published in final edited form as:

Med Image Comput Comput Assist Interv. 2015 October ; 9351: 198–206. doi:
10.1007/978-3-319-24574-4_24.

Multi-atlas Based Segmentation Editing with Interaction-Guided Constraints

Sang Hyun Park¹, Yaozong Gao^{1,2}, and Dinggang Shen¹

¹Department of Radiology and BRIC, UNC at Chapel Hill, NC 27599, USA

²Department of Computer Science, UNC at Chapel Hill, NC 27599, USA

Abstract

We propose a novel multi-atlas based segmentation method to address the editing scenario, when given an incomplete segmentation along with a set of training label images. Unlike previous multi-atlas based methods, which depend solely on appearance features, we incorporate *interaction-guided constraints* to find appropriate training labels and derive their voting weights. Specifically, we divide user interactions, provided on erroneous parts, into multiple local interaction combinations, and then locally search for the training label patches well-matched with each interaction combination and also the previous segmentation. Then, we estimate the new segmentation through the label fusion of selected label patches that have their weights defined with respect to their respective distances to the interactions. Since the label patches are found to be from different combinations in our method, various shape changes can be considered even with limited training labels and few user interactions. Since our method does not need image information or expensive learning steps, it can be conveniently used for most editing problems. To demonstrate the positive performance, we apply our method to editing the segmentation of three challenging data sets: prostate CT, brainstem CT, and hippocampus MR. The results show that our method outperforms the existing editing methods in all three data sets.

1 Introduction

Automatic segmentation methods have been proposed for various applications. However, these methods often generate erroneous results in some areas of an image caused by difficulties such as unclear target boundaries, large appearance variations and shape changes. If errors can be edited with a few user annotations after automated segmentation, the total segmentation time could be significantly reduced.

Many interactive segmentation methods [1, 2] have been proposed to address the editing problem. These methods can generate certain improved results within a few seconds by using distinct user guidance and simple appearance models. However, it is difficult to directly apply these methods to the editing problem, when allowing only limited annotations on a small number of erroneous parts. For example, the appearance model constructed by a few interactions is often limited to obtain the reliable result as shown in Fig. 1(b). Several methods have been proposed to incorporate high-level information from training data into

the editing framework to improve performance. Schwarz *et al.* [3] learned active shape model (ASM) and then incorporated it into an editing framework. When any incorrect landmark point is edited by users, the adjacent landmark points are modified accordingly and regularized by the ASM. However, manual editing of landmarks is intractable in the 3D space, and also the ASM with limited training data often fails to capture local shape variations. Park *et al.* [4] proposed an editing method based on a patch model that includes localized classifiers and spatial relationship between neighboring patches. In this method, the patches are transferred to appropriate places in a target image by considering the spatial relationship and the similarity between labels and interactions. Then the localized classifiers are used for segmentation. The process of simple similarity comparison, however, can cause the patches to be positioned in the wrong places, making the final label prediction inaccurate even if good classifiers are trained with elaborate learning methods.

In this paper, we propose a new editing method, focused on reliable estimation of label information without a complex learning. To achieve this, we refer to multi-atlas based segmentation methods [5, 6] that consist of two steps: (1) search of training label patches and (2) label fusion of the selected label patches with respective voting weights. So far, most multi-atlas based methods have used appearance features to achieve these two steps, with the basic assumption that similar patches have similar labels. However, it is typically easy to find patches with similar appearance, but distinct label patterns, for medical images, often include weak boundaries and regions with inter-subject appearance variations. Unlike these methods, we use constraints from user interactions to address both label patch selection and label fusion steps. Specifically, we divide user interactions into multiple local interaction combinations and then search for label patches that are well-matched with both the interactions and the previous segmentation for each combination. Next, we estimate new segmentation by label fusion of the selected label patches that have their weights computed with respect to distances to interactions. Specially, we introduce a novel label-based similarity to address the issue in step (1) and new definition of distance-based voting weights to address the issue in step (2). Compared to the case of considering all interactions jointly, the label patches, corresponding to multiple combinations, can generate more reliable results as shown in Figs. 1(c) and 1(d). Also, since our method does not need image information or learning procedures, it is convenient for practice.

2 Multi-atlas Based Editing Method

Our proposed editing procedure begins with an initial segmentation obtained from any existing method, training label images, and user interactions with erroneous parts. We provide an interface to receive user interactions on erroneous parts of an incomplete segmentation result. Intuitively, we assume that foreground (FG)/background (BG) dots or scribbles are inserted into the erroneous regions near the true object boundary. Specifically, the editing procedure consists of four steps: 1) all training labels are registered to the previous segmentation L^{t-1} for guiding the segmentation update, where t represents the editing iteration and L^0 is the initial segmentation. For the registration, we respectively extract surface points from the training label images and L^{t-1} , and then conduct the iterative closest point method [7]. 2) Local interaction combinations ($\hat{U}_k^t$ for FG and $\bar{U}_k^t$ for BG) are

extracted from the FG / BG user interactions, respectively, where k is the index of combination. For each combination, a region of interest (ROI) is set as a bounding box to include the interactions with a small margin. 3) For each combination, the appropriate training label patches, well-matched with both the interactions and L^{t-1} in the ROI, are searched in the aligned label images, and then averaged to build a local probabilistic likelihood map ($\hat{P}_k^t$ for FG and $\bar{P}_k^t$ for BG). 4) A global probabilistic likelihood map P^t in the entire image is determined by averaging L^{t-1} and the local probabilistic likelihood maps with their respective distance-based voting weight maps ($\hat{W}_k^t$ for FG and $\bar{W}_k^t$ for BG) (Fig. 2). Finally, the segmentation is determined by thresholding of P^t . These steps are repeated with inclusion of additional user interactions, if provided, until the segmentation is satisfactory. Note that when repeating each editing procedure, the accumulated user interactions are considered to find the label patches and derive their respective voting weights.

2.1 Extraction of Local Interaction Combinations

In our method, the segmentation is edited by using the information of training labels well-matched with user interactions. If there are many training labels well-matched with all interactions, the segmentation can be edited easily by following the guidance. However, there are few well-matched training labels in most cases. Thus, we separately find the training label patches, which are well-matched with various local interaction combinations, and then aggregate them to estimate the voxel likelihood. We extract three types of local combinations for FG and BG interactions, respectively, as follows: 1) *individual interaction* such as a dot or scribble, 2) *pairwise interaction* that includes two individual interactions within a certain distance, and 3) *union interaction* that includes all interactions within the certain distance from each interaction. The combinations are extracted from the interactions provided by the current round of editing, as well as relevant interactions previously. Specifically, if the previous interactions are located within a certain distance from current interactions, the combinations between current and previous interactions are extracted. On the other hand, the previous interactions, far from all the current interactions, will not be used in the current round of editing since the accurate parts of the updated segmentation do not need to be changed. For each k^{th} combination, we set ROI ($\hat{\varphi}_k^t$ for FG and $\bar{\varphi}_k^t$ for BG) as a bounding box, which covers the interaction combination with a small margin to include possible local variations in the ROI.

2.2 Selection of Training Labels *with respect to* User Interactions

For each interaction combination, we find training label patches well-matched with the interactions and previous segmentation L^{t-1} in ROI. Since training labels are aligned to L^{t-1} , without considering interaction information in the initial registration step, the registration might be inaccurate. To address this issue, we search for the best well-matched label patch in a local search neighborhood for each aligned training label map M with a predefined similarity. Recently, Park *et al.* proposed a label-based similarity [4] that measures the similarity between a label and the interactions. However, since they consider all interactions in ROI to be strong constraints, the number of training label patches well-matched with all

interactions is often limited, causing various shape changes to be unable to be well-captured.

To avoid this, we define the similarity S_k^t as:

$$S_k^t = \sum_{\substack{v \in \varphi_k^t, \\ U_k^t(v) \neq 0}} \delta \left(M(v) - U_k^t(v) \right) + \gamma_o \sum_{\substack{v \in \varphi_k^t, \\ U_k^t(v)=0, \\ U_k^t(v) \neq 0}} \delta \left(M(v) - U_k^t(v) \right) + \gamma_U \sum_{\substack{v \in \varphi_k^t, \\ U_k^t(v)=0, \\ U_k^t(v)=0}} \delta \left(M(v) - L^{t-1}(v) \right), \quad (1)$$

where δ is the kronecker delta and U_k^t includes all current and previous user interactions except U_k^t . If v is annotated by users, $U_k^t(v)=1$ for FG, and -1 for BG, while if v is not annotated, $U_k^t(v)=0$. Here, we assume that a good label patch for U_k^t should be strongly well-matched with U_k^t representing the k^{th} interaction combination (1^{st} term), moderately matched with other interactions except U_k^t (2^{nd} term), and also weakly matched with L^{t-1} on the other voxels (3^{rd} term). γ_o and γ_U denote the parameters for balancing these three terms. In our experiments, γ_o is set as 0.05 to distinguish the strong and moderate constraints for annotated voxels, and γ_U is set as 0.005 to represent the weak constraint for L^{t-1} . The best label patches are found from all the training labels and n_p patches with the highest S_k^t are selected. Finally, the selected patches are averaged to build a local probabilistic likelihood map P_k^t . (2^{th} row in Fig. 2.)

2.3 Label Fusion Based on User Interactions

In the fusion step, local likelihood maps are aggregated to build a global likelihood map P^t for the entire image. Since high emphasis is enforced to the first term of Eq. (1), the local likelihood map is more likely to be accurate near the interaction. In contrast, the confidence of estimated likelihood for a voxel becomes low when its distance from interactions increases. In these low-confidence regions, L^{t-1} is more accurate than the likelihood maps.

By considering the distance from interaction and the confidences of $\hat{P}_k^t$ and $\bar{P}_k^t$, we define $P^t(v)$ as:

$$P^t(v) = \frac{W_L^t(v)L^{t-1}(v) + \sum_{k=1}^{n_f} \hat{W}_k^t(v)\hat{P}_k^t(v) + \sum_{k=1}^{n_b} \bar{W}_k^t(v)\bar{P}_k^t(v)}{W_L^t(v) + \sum_{k=1}^{n_f} \hat{W}_k^t(v) + \sum_{k=1}^{n_b} \bar{W}_k^t(v)}. \quad (2)$$

where n_f , n_b are the numbers of FG and BG combinations. The weight of FG combination $\hat{W}_k^t(v)$ is defined as:

$$\hat{W}_k^t(v) = \frac{\alpha_f}{\sqrt{n_f}} \exp\left(-\left(\frac{d_k(v) \cdot d_k(v)}{2\sigma^2}\right)\right), \quad (3)$$

where $d_k(v)$ is the shortest distance between a voxel v and the annotated voxels in U_k^t . σ controls how quickly the emphasis shifts from the probabilistic likelihood maps to L^{t-1} as the voxel is farther from the interactions. If σ is small, only the region close to the user interactions is affected by the probabilistic likelihood maps, while the other regions are affected by L^{t-1} . α_f is the weight for the confidence of voxel likelihood. For example, when many likelihood maps indicate v as BG with low confidence (e.g. $P_k^t(v)=0.2 \sim 0.5$) and less likelihood maps indicate v as FG with high confidence (e.g. $P_k^t(v)=0.8 \sim 1$), we enforce more weight to the latter likelihood maps to follow the high confident likelihood. We set α_f as a value larger than 1, if $P_k^t(v) > 0.8$; otherwise 1. The BG weight $\bar{W}_k^t(v)$ is similarly defined. $W_L^t(v)$ is defined as: $W_L^t(v) = \exp\left(-\left(\frac{\beta \cdot \beta}{2\sigma^2}\right)\right)$, where β controls the importance of L^{t-1} . If β is small, the result is more affected by L^{t-1} compared to P_k^t and vice versa. Finally, we set the label $L^t(v)$ as FG if $P^t(v) > 0.5$ or v is annotated as FG; otherwise, set that as BG.

3 Experimental Result

Our method was evaluated on three data sets: prostate CT, brainstem CT, and hippocampus MR. We first made the automatic segmentation by using the state-of-the-art method [6] and then applied our method to the results with the greatest errors. Specifically, for the prostate set with 73 images, we applied the automatic method using four-fold cross validation and chose 30 images with the lowest DSC. For the brainstem set with 40 images and the hippocampus set with 35 images, we similarly applied the automatic method using the leave-one-out validation and chose 20 and 15 images with the lowest DSC, respectively. For each of these selected prostate, brainstem, and hippocampus images, 12, 30, and 9 dots were inserted on average as the user interactions, depending on the amount of segmentation errors. In our experiments, the ROI margin was set as $13 \times 13 \times 4$ voxels for prostate and brainstem, and $5 \times 5 \times 5$ voxels for hippocampus. Accordingly, the search range was set as $8 \times 8 \times 4$ and $4 \times 4 \times 4$ voxels, respectively. σ and β were set as 7 and 10 for prostate and brainstem, and 3 and 4 for hippocampus. n_p and α_f were set as 7 and 5 for all the experiments.

The proposed method, using the interaction combinations and the weighted voting (ICWV), was compared with 1) initial automatic segmentation, 2) manual editing, 3) a label fusion method using all interactions in the entire image (LFG) for finding training labels, 4) a label fusion method using interactions in local regions (LFL) for finding label patches, 5) a method using the patch model (PM) [4], and 6) a method using the proposed interaction combinations but with the majority voting (ICMV). The performance was measured by DSC between respective results and manual ground-truth labels. The boxplots in Fig. 3 show the distributions of DSC for the three data sets. Fig. 4 shows the qualitative results of LFG, LFL, PM and our proposed methods. In the manual editing method, only the labels of voxels annotated by users were changed. Due to the small amount of interactions, the improvement of manual method was less than DSC of 0.005 – 0.026. On the other hand, the methods using training labels gave significantly improved scores, i.e., more than DSC of 0.01 – 0.13

except the LFG method for hippocampus. Since the hippocampus has a non-convex shape with local variations, the number of training labels well-matched with all interactions was very limited, thus the inaccurate training labels that were selected often made the result worse (Fig. 4(c)). The LFL and PM methods had better performances. However, shape variations near the object boundary could not be well-captured in the LFL method due to the limited number of label patches well-matched with all interactions in ROI. The PM method had similar problem near the object boundary due to the patch localization errors and the low intensity contrast between FG and BG. The ICMV method outperformed these methods in terms of accuracy, by finding the well-matched label patches for multiple combinations. Nonetheless, since some inaccurate label patches derived from individual or pairwise combinations still contribute equally to the final result, the large standard deviations were obtained by comparing the proposed ICWV method. On the other hand, the ICWV method outperformed all other editing methods in most cases for both accuracy and robustness.

The procedure of repetitive editing is shown in Fig. 5. In this example, due to the large errors occurred in L^0 , there was a lack of good label patches that could cover the large shape change. Thus, the intermediate result (Fig. 5(c)) still included errors on the upper part. In the next round of editing, we inserted additional user dots, re-aligned all training labels to the intermediate result, and further refined the result with additional interaction combinations near the dots. Since the aligned labels were more reliable during the second round of editing, possible shape changes for the erroneous part were covered in the search range, allowing the segmentation to be more accurately demonstrated.

These experiments were performed on a PC with a 3.5 GHz Intel quad-core i7 CPU, and 16GB of RAM. The computational time depended on the number of training labels and interactions, the ROI margin size, and the search range. In our setting, the editing took 1.5 – 7 min, 2 – 18 min, and 50 – 80 seconds for prostate, brainstem, and hippocampus, respectively. Since the editing is conducted with all interactions in the image, our method was unable to promptly produce intermediate results. However, the time issue can be significantly reduced with GPU implementation as the same computation is repeated on every voxel in the search space. The time during the second round of editing was greatly reduced when the number of interaction combinations decreased. In the example of Fig. 5, editing was five times faster than that of the first round.

4 Conclusion

We have proposed a novel multi-atlas based segmentation editing method with interaction-guided constraints to find training label patches and derive voting weights. The proposed method can generate robust editing results in even challenging regions without image information and expensive learning steps. It can help produce accurate segmentations for difficult cases that failed in existing automatic methods. In the future, we will incorporate classifier-based methods into our framework to learn about image features from the label information and eventually release our editing method for public use¹.

¹<http://www.unc.edu/~tkdgus/>

References

1. Boykov Y, Funka-Lea G. Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*. 2006; 70(2):109–131.
2. Grady L. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2006; 28(11):1768–1783. [PubMed: 17063682]
3. Schwarz T, Heimann T, Tetzlaff R, Rau AM, Wolf I, Meinzer HP. Interactive surface correction for 3D shape based segmentation. *Proc SPIE*. 2008
4. Park SH, Lee S, Yun ID, Lee SU. Structured patch model for a unified automatic and interactive segmentation framework. *Medical Image Analysis*. 2015
5. Coupe P, Manjon JV, Fonov V, Pruessner J, Robles M, Collins DL. Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation. *NeuroImage*. 2011; 54(2): 940–954. [PubMed: 20851199]
6. Wu G, Wang Q, Zhang D, Nie F, Huang H, Shen D. A generative probability model of joint label fusion for multi-atlas based brain segmentation. *Medical Image Analysis*. 2014; 18(6):881–890. [PubMed: 24315359]
7. Besl PJ, Makay ND. A method of registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1992; 14(2):239–256.

Fig. 1.

Editing results for three methods: (b) graph cut, (c) majority voting, and (d) proposed method. Test image, initial segmentation (green), ground-truth (white line), and FG/BG interactions (blue/red dots) are shown in (a). Likelihood maps and editing results (green) are shown in the top and bottom rows of (b), (c), and (d), respectively.

Fig. 2.

Interaction combinations, their ROIs (top row), local likelihood maps (middle row), and weight maps (bottom row), obtained from the interactions shown in Fig. 1(a). The initial segmentation, ground-truth, FG/BG interactions, and ROI are shown as green, white line, red/blue dots, and red box, respectively. The global likelihood map in Fig. 1(d) is obtained by fusing L^{t-1} and the local likelihood maps with the weights.

Fig. 3.

The distributions of DSC scores for different editing methods. Top, middle, and bottom lines of the box represent upper quartile, median, and lower quartile, respectively. The whiskers represent maximum and minimum values, respectively.

Fig. 4. Segmentation editing results by LFG, LFL, PM, and our proposed methods. Segmentation results, ground-truth, and FG / BG interactions (UI) are shown as green, white lines, and red / blue dots, respectively.

Fig. 5.

The procedure of segmentation update with respect to the repetitive user interactions. (a) initial segmentation with large errors, (b) the segmentation with interactions, (c) intermediate segmentation result based on the interactions shown in (b), (d) the segmentation with additional interactions, (e) updated segmentation result based on the interactions shown in (d), and (f) ground-truth.