# Data-Driven Interactive 3D Medical Image Segmentation Based on Structured Patch Model

Sang Hyun Park[1], Il Dong Yun[2,⋆], and Sang Uk Lee[1]

[1] Electrical Engineering, ASRI, INMC, Seoul National University, Korea
shpark@cvl.snu.ac.kr, sanguk@snu.ac.kr
[2] Digital Information Engineering, Hankuk University of Foreign Studies, Korea
yun@hufs.ac.kr

**Abstract.** In this paper, we present a novel three dimensional interactive medical image segmentation method based on high level knowledge of training set. Since the interactive system should provide intermediate results to an user quickly, insufficient low level models are used for most of previous methods. To exploit the high level knowledge within a short time, we construct a structured patch model that consists of multiple corresponding patch sets. The structured patch model includes the spatial relationships between neighboring patch sets and the prior knowledge of the corresponding patch set on each local region. The spatial relationships accelerate the search of corresponding patch in test time, while the prior knowledge improves the segmentation accuracy. The proposed framework provides not only fast editing tool, but the incremental learning system through adding the segmentation result to the training set. Experiments demonstrate that the proposed method is useful for fast and accurate segmentation of target objects from the multiple medical images.

**Keywords:** interactive segmentation, 3D medical image, structured patch model, localized classifier.

## 1 Introduction

As creating an amount of medical images with the same modalities and properties, segmentation of desired objects from the multiple images becomes an important task for clinical studies and diagnosis of disease progression. Since the shapes and structures of an object in the medical images are generally maintained, many automatic methods [1–3] using prior knowledge of small number of training data have been proposed. However, most of them have struggled to obtain accurate segmentation for various clinical applications, because the medical images usually contain many vague boundaries and local variations of the same object. Therefore, an intelligent interactive segmentation method is inevitably preferable in the medical community [4].

⋆ Corresponding author. This research was supported by the NRF of Korea funded by the Ministry of Education, Science and Technology (2010-0012006).

J.C. Gee et al. (Eds.): IPMI 2013, LNCS 7917, pp. 196–207, 2013.

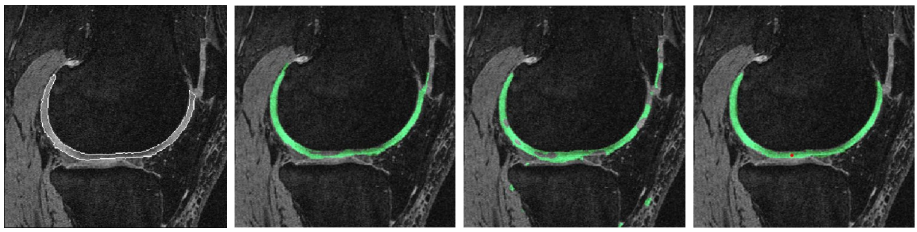
The intelligent interactive method should satisfy two conditions: first, the method has to be fast enough to provide intermediate result to user. Second, the method has to provide flexible and accurate result even though small numbers of user interaction are given. The most interactive methods [5–8] have not satisfied the second condition due to the insufficient low level models depending on user inputs. On the other hand, the methods based on high-level knowledge require heavy computation, making harder to satisfy the first condition, to make and infer the model. Recently, Barnes *et al.* proposed the high level image editing method named PatchMatch [9] for image completion and reshuffling. Although the method satisfies the both conditions by finding dense patch correspondences, it is not applicable to the interactive segmentation because structural spatial relations, important for fast search and accurate segmentation, are not considered.

In this paper, we propose a novel 3D interactive segmentation method satisfying the both conditions by transferring the prior knowledge of similar patches from training sets to a target image with an assumption that the objects in medical images have common structure. Unlike the PatchMatch [9, 10], we construct a structured patch model which includes structural spatial relationships of adjacent patches within an image as well as multiple corresponding patch sets representing properties of local regions across images. The spatial relationships accelerate the patch search speed by constraining the positions of adjacent patches, while the prior knowledge of corresponding patch sets improves the segmentation accuracy by reflecting the properties of each local region. The proposed framework based on the structured patch model provides not only fast editing tool, but the incremental learning system through adding the segmentation result to the training set.

### 1.1 Related Work

**Interactive Segmentation:** There have been many interactive segmentation methods such as graph cut [5, 6, 11], random walk [7] and region growing [8]. The efficiency of methods has been further improved by constructing the fine models [12, 13], or incorporating the active learning scheme [4, 14]. However, high-quality user interactions are still required because the segmentation only depends on the low level cues which are insufficient to reflect various properties of the medical image. That is, most existing methods have focused on the single image segmentation based on the user input without prior knowledge. In this paper, we discuss how to incorporate the high level prior knowledge into interactive segmentation framework within a short time when training sets are given. Unlike the previous methods, the efficiency of proposed method is increased for multiple images segmentation.

**Patch Matching:** The strategy, finding out and using matches of similar patches, is relatively fast and flexible compared with the voxel level matching methods, e.g. non-rigid registration [15, 16]. Therefore, the strategy have been applied to various vision applications [9, 10, 17–19]. Especially, since the corresponding patches are likely to have the same labels, the method is applied to



(a) test (b) 26 sec. (c) 3 sec. (d) 0.01 sec.

**Fig. 1.** Transfer a label of training image to a test image (2D). (a) The test image and its ground truth boundary (white), (b) the transferred label (green) by the non-rigid registration method [15], (c) that by the PatchMatch [9], (d) that by the proposed method with few user input (red dot). The computational times are presented below the results.

semantic segmentation by constructing a graph of corresponding patches across a large image set [19]. In this paper, we expand the strategy to the interactive segmentation. In the proposed model, the corresponding patches are not only found across the training set, but the connection of the patch sets is also considered. Furthermore, multiple cues of each patch are learned for accurate segmentation unlike the previous methods which only transfer the label. Fig. 1 shows the difference of performance between the non-rigid registration method [15], the PatchMatch method [9] and the proposed method for the label transfer.

**Patch Based Segmentation:** There have been many patch based methods such as label fusion [1, 2] and localized classifier [20, 3]. Since the localized classifier can consider local variations by adaptively integrating the multiple cues, we adopt the method with some modifications to enforce the knowledge of user interactions and training patches. Unlike the previous methods which are only focused on automatic manner, effective 3D editing tool is provided in our framework by updating the patch correspondences of local regions. Furthermore, the computational time is accelerated because the search space is limited by an implicit patch structure without complicated alignment steps and all priors of patches are learned in training step.

## 2 Structured Patch Model

Unlike the PatchMatch-based methods which find the similar patches over an whole image without the structure information [18, 19], we find the exact nearest patch correspondence with the assumption that the medical images contain the same structural object. For example, if patches $P(v_j^i)$ and $P(v_{j'}^i)$, centered at voxels $v_j^i$ and $v_{j'}^i$, are neighbor in a volume $V^i$, the corresponding patches $P(v_j^{i'})$ and $P(v_{j'}^{i'})$ of $P(v_j^i)$ and $P(v_{j'}^i)$ should be neighbor in another volume $V^{i'}$. Even though a patch more similar to $P(v_{j'}^i)$ than $P(v_{j'}^{i'})$ exists elsewhere in $V^{i'}$, $P(v_{j'}^{i'})$ next to $P(v_j^{i'})$ is determined as the most informative patch in our model.

### 2.1 Building Structured Patch Model

The structured patch model includes the connection of adjacent patches within an image and patch correspondences across images. To model this, the connection of neighboring patches are firstly constructed from a training set, then the corresponding patches are found from the other training sets. From $m$ training sets $\mathbb{T} = \{T^i = (V^i, L^i) | i = 1, ..., m\}$ including the volume $V^i$ and their manual label $L^i$, an initial set $T^{i'} = (V^{i'}, L^{i'})$ is randomly selected. From the surface $\xi(L^{i'})$ of $L^{i'}$, $n$ voxels $\mathbf{v}^{i'} = \{v_j^{i'} | j = 1, ..., n\}$ are sampled with even distribution and the patches $\mathbf{P}^{i'} = \{P(v_j^{i'}) | j = 1, ..., n\}$ are constructed. Here, $\mathbf{v}^{i'}$ are sampled enough to overlap the adjacent patches (Details are listed in Sec. 5). Each patch has indices of the adjacent patches $\mathbf{N}_j = \{N_j(k) | k = 1, ..., l\}$ and their relative positions $\mathbf{r}_j^{i'} = \{r_j^{i'}(k) = p(v_{N_j(k)}^{i'}) - p(v_j^{i'}) | k = 1, ..., l\}$, where $p(v)$ denotes the position of voxel $v$ and $l$ is the number of adjacent patches.

The corresponding patches from another training set $T^i$ are found by propagating $\mathbf{P}^{i'}$ to the relative positions of $V^i$. The propagation is started on the patch centered at the left-top point of $L^i$. First, we find the most similar patch with the starting patch among $\mathbf{P}^{i'}$. If the index of the most similar patch is $j$, the voxel $v_j^i$, the center of corresponding patch of $P(v_j^{i'})$, is searched near the left-top point within local search space $\Upsilon_j^i$ in $V^i$ as:

$$v_j^i = \underset{v \subset \Upsilon_j^i}{\arg\max}(S(P(v), P(v_j^{i'}))), \tag{1}$$

where $S(\cdot, \cdot)$ denotes the similarity cost. After $P(v_j^i)$ is determined as the corresponding patch of $P(v_j^{i'})$, the adjacent patches of $P(v_j^{i'})$ are propagated to $V^i$. $k^{th}$ adjacent patch $P(v_{N_j(k)}^{i'})$ is searched near the relative position $p(v_j^i) + r_j^{i'}(k)$ by (1). The propagation and the search are repetitively conducted until the all corresponding patches $\mathbf{P}^i$ of $\mathbf{P}^{i'}$ are determined. Since the search space is constrained by $\mathbf{r}_j^{i'}$, the propagation speed is accelerated even if the position of corresponding patch is exhaustively searched. The procedure is shown the upper row in Fig. 2.

The structured patch model $\mathbb{P} = \{\mathbf{P}^i | i = 1, ..., m\}$ is constructed by repeating the patch matching procedure to $m-1$ training sets. Note that $\mathbf{P}^i$ represents the intra patch set from $T^i$, while $\mathbf{P}_j$ represents the corresponding inter patch set of $j^{th}$ local region across the training sets. The corresponding patch set reflects characteristics of the same local region of the training sets.

### 2.2 Adaptive Patch Propagation

Unlike the propagation of $\mathbf{P}^{i'}$ to $V^i$ in the training step, that of the multiple patch sets $\mathbb{P}$ to a target volume $V$ has to be conducted in the test step. Since the appropriate prior according to the local regions is different, we adaptively select the good corresponding patch on $j^{th}$ local region from $\mathbf{P}_j$. The algorithm is started on few user input, e.g. a point near the object boundary. Although

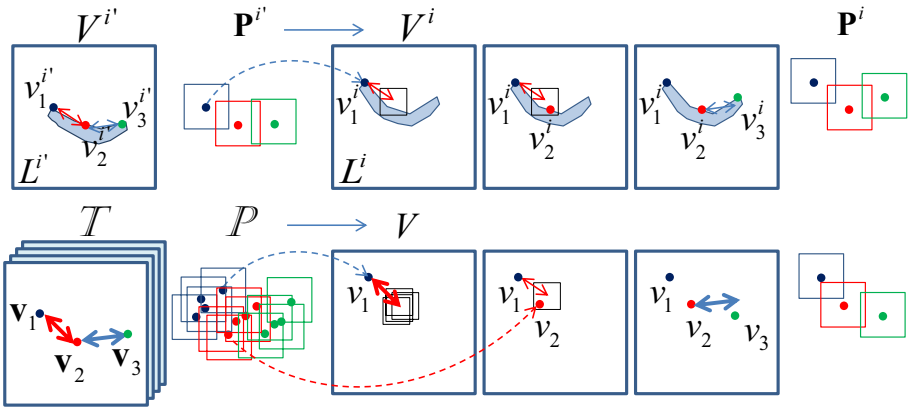


**Fig. 2.** Propagation and local search of the structured patch model. Upper row shows the propagation of $P^{i'}$ to $V^i$ in the training step, while bottom row shows the propagation of $\mathbb{P}$ to $V$ in the test step. The blue regions in upper row show labels and the points represent the center of patches. The neighboring point is searched from the local search space (black windows) near the relative positions (arrows).

the initial input can be automatically estimated by detection methods, we focus on the interactive system in this paper. We construct initial patch including the user input and find the most similar patch from $\mathbb{P}$ and its corresponding position by (1). Then, the adjacent patches are propagated. If the index of adjacent patch is $j$, the index $i_{opt}$ of the best matching patch among $\mathbf{P}_j$ as well as the voxel $v_j$ which is the center of corresponding patch of $P(v_j^{i_{opt}})$ in $V$ is searched as:

$$(i_{opt}, v_j) = \underset{v \subset \Upsilon_j, i}{\arg\max}(S(P(v), P(v_j^i))). \tag{2}$$

That is, the patch $P(v_j)$ centered at $v_j$ is localized in $V$ and $P(v_j^{i_{opt}})$ is determined as the corresponding patch of $P(v_j)$. The adjacent patches are repetitively propagated to $V$ by searching the best correspondences and the positions of adjacent patches near $p(v_j) + r_j^{i_{opt}}(k)$ within $\Upsilon_j$ as (2). The procedure is shown in the bottom row in Fig. 2. Finally, the positions of all patches are localized in $V$ and its corresponding reference patch set $\tilde{\mathbf{P}}$ is obtained from $\mathbb{P}$. Note that the each element of $\tilde{\mathbf{P}}$ is referred by the difference training set.

### 2.3 Similarity Cost Function

The similarity cost $S(\cdot,\cdot)$ between patches can be computed by various functions. Among them, the normalized cross correlation (NCC) $S^{ncc}(\cdot,\cdot)$ and the overlapping cost, computed as $S^{ovl}(P(v_j^i), P(v_{j'}^{i'})) = \frac{|L_j^i \cap L_{j'}^{i'}|}{|L_j^i \cup L_{j'}^{i'}|}$, are used in the proposed method. Since the labels of two patches are required for computing $S^{ovl}(\cdot,\cdot)$, the overlapping cost is used for the training and editing steps. On the other hand, $S^{ncc}(\cdot,\cdot)$ is used for the test step.

### 2.4 Learning Multiple Cues

The patch provides informative cues of the local region. For each patch $P(v_j^i)$ including sub-volume $V_j^i$ and sub-label $L_j^i$, we extract mean and variance of intensities of voxels inside $V_j^i$ for computing NCC and the spatial information of neighboring patches for constraining the search space. Furthermore, we learn additional cues to construct a segmentation framework which is based on the conditional random field (CRF) model consisting of likelihood and smoothness terms. Details are as follow:

**Multiple Cues Based on Localized Classifier.** Inspired by the methods based on the localized classifier [20, 3], the likelihood of a voxel is determined by weighted sum of the probabilities regarding shape and appearance models. We use $L_j^i$ and histograms $H_j^f$, $H_j^b$ of voxel intensities of foreground (FG) and background (BG) labeled region as the shape model and the appearance model, respectively. The weight $w_j(v)$ between the shape and appearance probabilities is computed by the appearance confidence $\sigma_j$ and the distance $d_j(v)$ from $\xi(L_j^i)$ as:

$$w_j(v) = 1 - \exp(-d_j^2(v)/\sigma_j). \tag{3}$$

When the voxel position is close to the surface and the appearances of FG and BG are distinguishable, $w_j(v)$ is decreased (the appearance cue is emphasized) and vice versa. $\sigma_j$ is computed by sum of the difference between $L_j^i$ and the FG probability regarding $H_j^f$ and $H_j^b$. The detailed descriptions of $\sigma_j$ and $w_j(v)$ are referred to [20].

**Ratio between Likelihood and Smoothness.** Since the CRF energy of small patch is sensitive to the ratio $\lambda_j^i$ between the likelihood and smoothness terms, we estimate $\lambda_j^i$ by using the shape of $L_j^i$. With observation that the result is likely to be over-smoothed when the surface is too large compared to the volume, e.g. thin parts, $\lambda_j^i$ is computed by the ratio of FG volume $|L_j^i|$ and surface area $|\xi(L_j^i)|$ of $L_j^i$ as: $\lambda_j^i = \lambda' \cdot \left(\frac{|L_j^i|}{|\xi(L_j^i)|}\right)$, where $\lambda'$ is a parameter.

## 3 Segmentation Based on the Learned Priors

The voxel-labeling problem based on probabilistic models have been formulated as a pairwise CRF. We introduce two kinds of segmentation strategy based on the CRF model according to the learned priors $\tilde{\Theta} = \{\tilde{\theta}_j \supset \{\tilde{L}_j, \tilde{H}_j^f, \tilde{H}_j^b, \tilde{w}_j(v), \tilde{\lambda}_j\} | j = 1, ..., n\}$ of $\tilde{\mathbf{P}}$: one is based on patch-wise manner, while the other one is global manner.

For the patch-wise manner, the voxel-wise segmentation is conducted on each patch by using the learned prior $\tilde{\theta}_j$ and the user scribble $U$. The CRF energy on $j^{th}$ patch is formulated as:

$$E(\mathbf{x}_j|\tilde{\theta}_j, U) = \sum_{v \in V_j} \phi(x_v|\tilde{\theta}_j, U) + \tilde{\lambda}_j \sum_{u,v \in \Gamma_j} |x_u - x_v| \cdot \exp \frac{|I(u) - I(v)|}{2\beta}, \tag{4}$$

where $x_v$ is a random variable representing the label of voxel $v$ in voxel set $V_j$ and $\mathbf{x}_j$ is the label variable set of $j^{th}$ patch. $\phi(x_v|\tilde{\theta}_j, U)$ represents the likelihood term of single variables $x_v$ when $\tilde{\theta}_j$ and $U$ is given, while the exponential function represents the smoothness term between neighboring nodes $(u, v)$ in neighbor set $\Gamma_j$. $\beta$ is the average square-distance of intensities between adjacent voxels in $V_j$ [5]. $\phi(x_v|\tilde{\theta}_j, U)$ is defined by the negative log of likelihood probability $Pr(v|\tilde{\theta}_j, U)$ as: $\phi\left(x_v|\tilde{\theta}_j, U\right) = -\log(Pr(v|\tilde{\theta}_j, U))$. If $U(v) = FG$ or $BG$, $Pr(v|\tilde{\theta}_j, U)$ is set as 1 or 0, respectively. If $U(v)$ is unknown,

$$Pr(v|\tilde{\theta}_j, U) = \tilde{w}_j(v)Pr(v|\tilde{L}_j) + (1 - \tilde{w}_j(v))Pr(v|\tilde{H}_j^f, \tilde{H}_j^b). \tag{5}$$

The probability $Pr(v|\tilde{L}_j)$ based on the reference label is computed as:

$$Pr(v|\tilde{L}_j) = \begin{cases} 1, & \text{if } \tilde{L}_j(v) = FG \\ 0, & \text{if } \tilde{L}_j(v) = BG \end{cases} \tag{6}$$

while $Pr(v|\tilde{H}_j^f, \tilde{H}_j^b)$ based on the appearance model is computed as:

$$Pr(v|\tilde{H}_j^f, \tilde{H}_j^b) = \frac{P(I(v)|\tilde{H}_j^f)}{P(I(v)|\tilde{H}_j^f) + P(I(v)|\tilde{H}_j^b)}, \tag{7}$$

where $I(v)$ denotes the intensity of $v$. Since the sub-modularity condition of (4) is satisfied, the optimal solution is obtained by the graph cut [5]. The patch-wise segmentations are aggregated on overlapping regions by averaging. Then, the voxels which have the higher averaging values than 0.5 are set as the foreground.

The procedure of global manner is similar with that of patch-wise segmentation. However, the likelihood probabilities of all patches are aggregated to a global likelihood by averaging the probabilities on overlapping regions before the patch-wise segmentation. The CRF energy model on the whole volume $V$ is constructed by the global likelihood term and the smoothness term as:

$$E(\mathbf{x}|\tilde{\Theta}) = \sum_{v \in V} \phi(x_v|\tilde{\Theta}) + \lambda \sum_{u,v \in \Gamma} |x_u - x_v| \cdot \exp \frac{|I(u) - I(v)|}{2\beta}, \tag{8}$$

where $\mathbf{x}$ is label variable set in whole volume. Similarly, (8) is optimized by the graph cut [5]. The global method is faster than the patch-wise method because the max-flow algorithm of graph cut is conducted once. On the other hand, the result can be over-smoothed on vague regions or thin parts because $\lambda_j^i$ according to the local shape is not considered.

## 4 Interactive Framework

Although most automatic methods have reduced the user effort, laborious slice-by-slice manual editing or post processing in a whole volume is required; the

**Algorithm 1.** Proposed framework based on the structured patch model

**Input**: the structured patch model $\mathbb{P}$ and a target volume $V$.
1: Adaptive patch propagation with few user input. (Sec. 2.2)
2: Segmentation by optimizing Eq. (8). (Sec. 3)
3: **Iterate** $4 \sim 6$ steps,
4: Input the user scribbles $U$ on false regions.
5: Update the reference patches by Eq. (2) on local regions where the user scribbles are included.
6: Segmentation on the local regions by optimizing Eq. (4). (Sec. 3)
7: **Until** the segmentation is satisfied.
8: (Optional) Add $V$ and the result to $\mathbb{P}$ by the patch propagation (Sec. 2.1, 2.4)

intelligent editing process is not provided. On the other hand, the structured patch model provides the effective local editing framework by using the constructed corresponding patch sets.

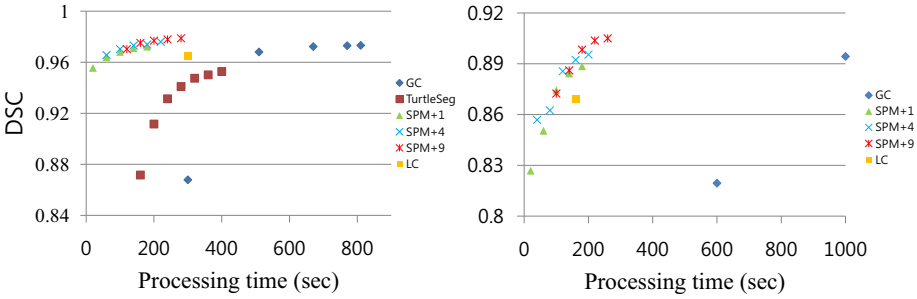
The proposed framework is shown in Algorithm 1. Specifically, the framework is divided into the initial segmentation step and the editing step. The initial step is started from few user input near the boundary of a target object. $\mathbb{P}$ is set to the corresponding positions of the target volume by the adaptive propagation. Then, the initial result is obtained by the global segmentation. The editing step is started from the initial result. FG or BG scribbles are input on false parts by the user and the local regions (patches) including the scribbles are found. On those local regions, the corresponding reference patches and the positions are updated by (2) with the overlapping similarity cost between the labels and the user scribbles. The modified result is obtained by optimizing (4) which is computed by the updated reference patch and the user scribbles. The editing step is repeated until the segmentation is satisfied. After the segmentation is done, the test volume and its segmentation can be incrementally added to the training set by the propagation mentioned in Sec. 2.1 and 2.4.

In the proposed framework, the editing time takes less than few seconds because the editing is conducted on the small number of local regions. In addition, the updated reference patch guides the segmentation to true boundaries even though the user scribbles are roughly input.

## 5 Experiments

The structured patch model (SPM) was evaluated on various organs. First, we present quantitative validation for segmentation of femur and femoral cartilage from five knee MR images[1] with $384 \times 384 \times 160$ voxel dimensions and $0.36 \times 0.36 \times 0.70\,\mathrm{mm}^3$ resolutions. Two organs have very different properties: the femur is thick and cylindrical, while the femoral cartilage is thin and deformable. In

[1] The data were acquired from the Osteoarthritis Initiative (`http://www.oai.ucsf.edu`).



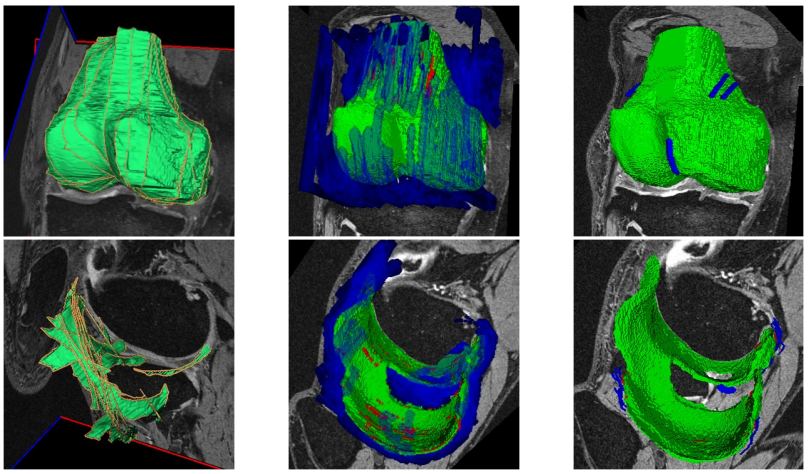

(a) femur (b) femoral cartilage

**Fig. 3.** Comparisons of femur and femoral cartilage segmentation between the graph cut [5] (GC), the method based on active learning [4] (TurtleSeg), the automatic method based on localized classifier [3] (LC), and the proposed method (SPM+the number of training set). The graph shows the average DSC values versus the cumulative processing time (interaction time+computational time) of five test images. The interval of the points represents the processing time between iterations. In the case of cartilage, the result of TurtleSeg is not presented because the method does not work well within the several numbers of user delineations.

the proposed framework, the patch size was varied according to the object size to include the meaningful local region. We empirically set the patch size as $51 \times 51 \times 25$ for the femur and $31 \times 31 \times 15$ for the cartilage. The interval between the sampled points (center of the patches) was set to make the adjacent patches with $60 \sim 70$ percents overlap. $\Upsilon_j^i$ and $\Upsilon_j$ were set to one-third of the patch size. All experiments were conducted by the single core program on a PC with 2.93 GHz Intel Core i7 CPU, and 16GB of RAM.

The performance of SPM was compared with two interactive methods and an automatic method: the method based on 3D graph cut [11, 6] (GC), the method based on active learning [4] (TurtleSeg), and the automatic method based on localized classifier [3] (LC). The GC was conducted on cropping region near the target object because there were many BG regions having the similar appearance of the target object. The SPM was experimented regarding the different size of training set and the LC used nine images as the training sets. The segmentation accuracy was measured by Dice similarity coefficient (DSC) between the segmentation result $S$ and ground truth $R$ as: $\mathbf{dsc}(S, R) = \frac{2 \cdot |S \cap R|}{|S| + |R|}$.

Fig. 3 presents the comparisons of the accuracy versus the cumulative processing time including interaction time and computational time. Fig. 4 shows the difference of user inputs between the methods. The result of GC was converged to accurate result within few iterations. However, the interaction time between the iterations took long time because lots of scribbles should be input to prevent the result passing over true boundaries on vague regions or over-smoothing on thin parts. The TurtleSeg reduced the interaction time by providing the user with uncertain 2D planes. However, the improvement according to the user delineations was converged on rough segmentation. Furthermore, the method



(a) TurtleSeg (b) GC (c) SPM

**Fig. 4.** Comparison of the user inputs between TurtleSeg [4], graph cut [5], and the proposed method. Green shows the segmentation result obtained by the user inputs, yellow lines represent the user delineations, and red and blue represent FG and BG scribbles, repectively.

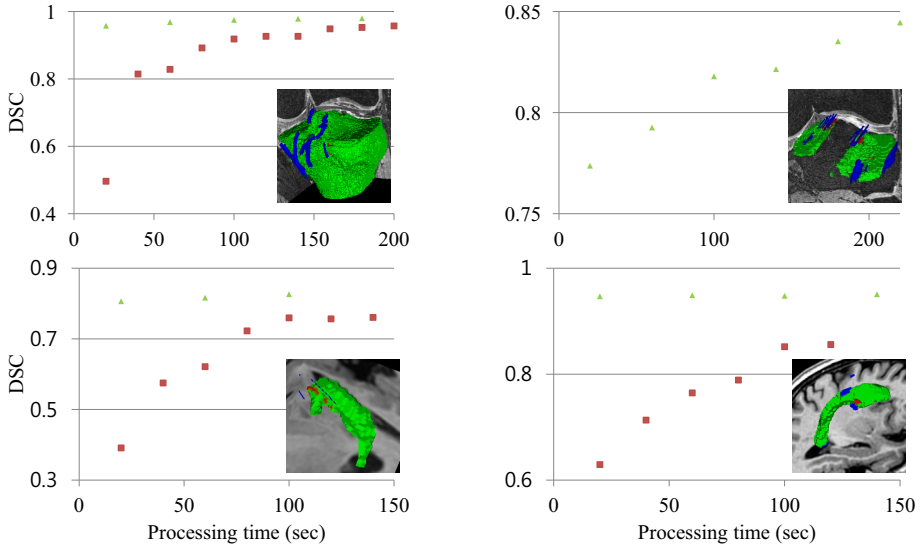



**Fig. 5.** Comparisons between SPM with one training set (green triangles) and TurtleSeg (brown rectangles) for tibia, tibial cartilage, hippocampus, and ventricle. Figures inside the graphs show segmentation results and user scribbles of the proposed method.

required lots of delineations for the cartilage segmentation because its shape is very thin and deformable (Fig 4 (a)). On the other hand, the SPM obtained the accurate result, outperforming that of the LC method, within few processing time on both cases. For most cases, the SPM obtained better DSC than the LC method during the same processing time. As increasing the size of training set, the computational time of the initial step was increased while the performance of initial segmentation was enhanced.

For general purpose, the SPM with one training set was applied to the segmentation of other organs: tibia and tibial cartilage in the knee MRI, hippocampus and ventricle in the brain MRI[2]. The DSC versus the cumulative processing time was compared with TurtleSeg (Fig 5). The results imply that the SPM is applicable to various problems without many training sets and useful to the segmentation of multiple volumes more than two.

## 6 Discussion

Most previous methods usually focus on the automatic or interactive manners. The main contribution of the proposed framework is to provide the connection between the automatic and interactive strategies by using the SPM. The method largely reduces the laborious user efforts by automatically segmenting the most parts, which relatively easy to be inferred by the prior knowledge, in the initial step. The propagated patch structure provides the intelligent editing system by updating the patch models according to the user interaction.

Since the corresponding patch set represents the properties of the same local region, we expect that the SPM will be expanded more effective ways. First, the proposed framework can be combined with the active learning scheme to provide an user with the slices including uncertain local regions. It is useful for our framework because most errors on the editing step occur on small numbers of vague local regions. The uncertainty can be measured through the comparison between the priors such as $L_j$, $\sigma_j$ of the reference patch and the segmentation. Second, local statistics and variations according to the pathological change can be modeled as increasing the size of training data. We have plan to cluster the similar patches and separately model the status of the cluster in each local region for managing the training patches. Third, the method can be expanded to the multi-object segmentation by using the optimization methods dealing with the multi-label problem. Finally, the algorithm can be parallelized because the same processes are repeated for multiple patches. The presented expansions are left to the future work.

## References

1. Coupe, P., Manjon, J.V., Fonov, V., Pruessner, J., Robles, M., Collins, D.L.: Patch-based segmentation using expert priors: application to hippocampus and ventricle segmentation. Neuroimage 54(2), 940–954 (2011)

[2] The data were acquired from the Alzheimer's Disease Neuroimaging Initiative (`http://www.loni.ucla.edu/ADNI`).

2. Rousseau, F., Habas, P.A., Studholme, C.: Human brain labeling using image similarities. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (2011)
3. Lee, S., Park, S.H., Shim, H., Yun, I.D., Lee, S.U.: Optimization of local shape and appearance probabilities for segmentation of knee cartilage in 3-D MR images. Computer Vision and Image Understanding 115(12), 1710–1720 (2011)
4. Top, A., Hamarneh, G., Abugharbieh, R.: Active Learning for Interactive 3D Image Segmentation. In: Fichtinger, G., Martel, A., Peters, T. (eds.) MICCAI 2011, Part III. LNCS, vol. 6893, pp. 603–610. Springer, Heidelberg (2011)
5. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient N-D image segmentation. International Journal of Computer Vision 70(2), 109–131 (2006)
6. Shim, H., Chang, S., Tao, C., Wang, J., Kwoh, C., Bae, K.: Knee cartilage: efficient and reproducible segmentation on high-spatial-resolution MR images with the semiautomated graph-cut algorithm method. Radiology 251(2), 548–556 (2009)
7. Grady, L.: Random walks for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(11), 1768–1783 (2006)
8. Pohle, R., Toennies, K.D.: Segmentation of medical images using adaptive region growing. In: Proc. SPIE M.I. (2001)
9. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: A randomized correspondence algorithm for structural image editing. In: Proc. SIGGRAPH (2009)
10. Barnes, C., Shechtman, E., Goldman, D.B., Finkelstein, A.: The generalized PatchMatch correspondence algorithm. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 29–43. Springer, Heidelberg (2010)
11. Shim, H., Kwoh, C., Yun, I., Lee, S., Bae, K.: Simultaneous 3-d segmentation of three bone compartments on high resolution knee mr images from osteoarthritis initiative (oai) using graph-cuts. In: Proc. SPIE M.I. (2009)
12. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut" - Interactive foreground extraction using iterated graph cuts. In: Proc. SIGGRAPH (2004)
13. Mory, B., Ardon, R.: Non-euclidean image-adaptive radial basis functions for 3D interactive segmentation. In: Proc. International Conference on Computer Vision (2009)
14. Wang, D., Yan, C., Shan, S., Chen, X.: Active Learning for Interactive Segmentation with Expected Confidence Change. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part I. LNCS, vol. 7724, pp. 790–802. Springer, Heidelberg (2013)
15. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: SIFT flow: Dense correspondence across different scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 28–42. Springer, Heidelberg (2008)
16. Glocker, B., Komodakis, N., Tziritas, G., Navab, N., Paragios, N.: Dense image registration through MRFs and efficient linear programming. Medical Image Analysis 12(6), 731–741 (2008)
17. Bleyer, M., Rhemann, C., Rother, C.: PatchMatch stereo - stereo matching with slanted support windows. In: Proc. BMVC (2011)
18. Zhang, H., Fang, T., Chen, X., Zhao, Q., Quan, L.: Partial similarity based nonparametric scene parsing in certain environment. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (2011)
19. Gould, S., Zhang, Y.: PATCHMATCHGRAPH: Building a graph of dense patch correspondences for label transfer. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 439–452. Springer, Heidelberg (2012)
20. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapcut: robust video object cutout using localized classifiers. In: Proc. SIGGRAPH (2009)