Active Learning for Interactive 3D Image Segmentation

Andrew Top[1], Ghassan Hamarneh[1], and Rafeef Abugharbieh[2]

[1] Medical Image Analysis Lab, Simon Fraser University
[2] Biomedical Signal and Image Computing Lab, University of British Columbia
{atop,hamarneh}@sfu.ca, rafeef@ece.ubc.ca

Abstract. We propose a novel method for applying active learning strategies to interactive 3D image segmentation. Active learning has been recently introduced to the field of image segmentation. However, so far discussions have focused on 2D images only. Here, we frame interactive 3D image segmentation as a classification problem and incorporate active learning in order to alleviate the user from choosing where to provide interactive input. Specifically, we evaluate a given segmentation by constructing an "uncertainty field" over the image domain based on boundary, regional, smoothness and entropy terms. We then calculate and highlight the plane of maximal uncertainty in a batch query step. The user can proceed to guide the labeling of the data on the query plane, hence actively providing additional training data where the classifier has the least confidence. We validate our method against random plane selection showing an average DSC improvement of 10% in the first five plane suggestions (batch queries). Furthermore, our user study shows that our method saves the user 64% of their time, on average.

1 Introduction

3D image segmentation is one of, if not the most, important and ubiquitous tasks in medical image analysis. Fully-manual slice by slice segmentation of images is widely recognized as infeasible, being too tedious, time consuming, expensive, and suffering from high inter- and intra-operator variability. Furthermore, manual operators have very limited ability to integrate the information available in 3D data given that almost all displays and data entry mechanisms are 2D. On the other side of the spectrum, fully-automated segmentation techniques have generally struggled to achieve the accuracy and robustness levels needed for clinical practice. Consequently, highly automated interactive image segmentation approaches [8] have recently become the approach of choice in most real life medical applications, and thus are garnering the focus of the medical image analysis community.

Many interaction mechanisms have been previously pursued such as interactive contour delineation [7] or region seeding [3]. Interactive segmentation is often an iterative process. A user provides input that guides computations which return output back to the user so that they may provide additional input. Ideally, the interaction process can occur in real-time so that users receive immediate feedback on their actions. In the case where user interaction takes the form of specifying which voxels belong to the object or region of interest (ROI) and which do not, the interactive segmentation process can be nicely formulated within a supervised machine learning framework. The user input

G. Fichtinger, A. Martel, and T. Peters (Eds.): MICCAI 2011, Part III, LNCS 6893, pp. 603–610, 2011.
© Springer-Verlag Berlin Heidelberg 2011

can thus be seen as labeled training data, which the interactive segmentation algorithm, or classifier, uses to label the remaining data (i.e. to segment the ROI).

Active learning (AL) [11] refers to supervised machine learning where, instead of the user, a component of the algorithm called the 'query strategy' is responsible for choosing the training data to be labeled. The query strategy, having intimate knowledge of the inner workings of the specific classification algorithm deployed, is in an excellent position to determine candidate high-quality training data voxels. By labeling voxels according to AL queries, interactive segmentation algorithms can produce improved segmentations using less, more efficient user input.

Image segmentation has had a number of treatments in the literature that relate to AL. Ma et al. [6] explored image segmentation under an AL framework where they used a support vector machine as the segmentation method and uncertainty sampling as the query strategy. Pavlopoulou et al. [9] discussed the translation of interactive contour delineation to an AL formulation. They used the term "active learning" in reference to the algorithm's support for interactive correction of the contour should it deviate from the ground truth, but this definition differs from that of formal AL. Li et al. [5] explored the segmentation of hyperspectral images by employing AL to query the user for more seedpoints when confidence is lacking. Queries were made in the form of a set of 2D image pixels labeled with low confidence, but it is unclear how to present these queries in a user-friendly fashion. In [12], we presented a new approach to optimize user interaction using a 'Spotlight' feature that was recently implemented into the software package TurtleSeg[1]. That work however was not formulated in an AL framework.

None of the previous methods treated AL in the context of 3D segmentation, which is the focus of our work in this paper. We solve two challenges: 1. The 3D image must be presented to the user on a 2D display in a way that best reveals the object. Once constructed (using some method), a candidate segmentation should be properly overlaid in context. 2. The user must assess the entire 3D segmentation in order to improve it by providing additional input. The simple inspection of the entire image and its segmentation, which is straightforward in 2D, becomes a much more complicated issue in 3D. In order to deal with the first challenge, we adopt the multi-planar reconstruction (MPR) technique where orthogonal or oblique slices of the 3D image are presented to the user. MPR is preferred as it is the most common 3D image display technique in clinical practice. To address the second challenge, we employ an AL uncertainty sampling query strategy, which is designed to highlight to the user the regions in 3D where additional training data will likely improve the segmentation. To the best of our knowledge, we are the first to formulate 3D interactive image segmentation in an AL framework.

2 Methods

For clarity of exposition, we start by defining our notation. Let $\Omega \subset \mathbb{R}^3$ represent the spatial image domain, $y : \Omega \rightarrow \{0, 1\}$ represent the classifier labeling, $I : \Omega \rightarrow \mathbb{R}$ represent the image intensity function, and $U : \Omega \rightarrow \mathbb{R}$ represent the classification uncertainty field. We formulate our 3D image segmentation as a data classification problem. Our feature vector consists of pixel locations $\boldsymbol{x} \in \Omega$ and intensity $I(\boldsymbol{x})$,

[1] http://www.turtleseg.org

but since the intensity depends on the position, we hereafter refer to the feature vector as $\boldsymbol{x}$ for simplicity. The solution to the classification problem is to determine a label $y(\boldsymbol{x}) \in \{0,1\}$ for each $\boldsymbol{x} \in \Omega$. We adopt a supervised learning approach for interactive segmentation. The user provides a set of training data, T, in which each element has the form $(X,Y) \in T$, where $X \in \Omega$, $Y \in \{0,1\}$ and each X is unique, that is $\forall (X_a,Y_a),(X_b,Y_b) \in T, X_a = X_b \implies a = b$. The training data is used to train a classifier and obtain an image of labels i.e. a segmentation, y. The classifier can be any probabilistic or energy minimizing 3D image segmentation algorithm, although probabilistic algorithms (for example [3]) are preferred since they naturally encode uncertainty in the results. Learning occurs as the training data is augmented, improving the ability of the classifier to label unlabeled image pixels.

Since the initial set of training data is typically not sufficient for the classifier to produce accurate labelings, we assess the uncertainty, U, of the classification results. Should the uncertainty remain above a preset threshold or the user deems the segmentation to be of insufficient quality, an AL batch query is calculated from U in order to present to the user regions of maximal uncertainty. Specifically, we find the plane that passes through the most uncertain regions of the classification. The user's resulting input augments the current training data, and the learned classifier is re-executed.

2.1 Uncertainty Field of Automatic Segmentation

The uncertainty value at a point $\boldsymbol{x}$ in the image, $U(\boldsymbol{x})$, reflects the lack of confidence of the classifier in its classification $y(\boldsymbol{x})$ at point $\boldsymbol{x}$. We propose a novel composite uncertainty field U, comprising a weighted sum of four sub fields.

$$U(\boldsymbol{x},y) = \lambda_E U_E(\boldsymbol{x}, p_1(\boldsymbol{x})) + \lambda_B U_B(\boldsymbol{x},y) + \lambda_R U_R(\boldsymbol{x},y) + \lambda_S U_S(\boldsymbol{x},y), \quad (1)$$

where U_E is an entropy energy that depends on the probability $p_1 : \Omega \rightarrow [0,1]$ of point $\boldsymbol{x}$ being labeled 1. U_B is a boundary energy, U_R is a regional energy, and U_S is a smoothness energy. Note that we are not interested in finding a classification $y(\boldsymbol{x})$ that minimizes the sum in (1) over the image domain, but rather how the value of U at one position in the image compares to the value of U at another. For example, a classifier which minimizes the boundary term U_B given a set of constraints might still give a solution with high U_B in regions where there are no hard edges to follow. When using a probabilistic segmentation algorithm as a classifier, a natural choice for assessing uncertainty is the entropy of the segmentation results, giving the entropy energy as

$$U_E(\boldsymbol{x}, p_1(\boldsymbol{x})) = -p_1(\boldsymbol{x}) log_2\, p_1(\boldsymbol{x}) - (1 - p_1(\boldsymbol{x})) log_2 (1 - p_1(\boldsymbol{x})). \quad (2)$$

For the boundary energy, we adopt from [1] a function of the image intensity gradient,

$$U_B(\boldsymbol{x},y) = \delta(D_s(\boldsymbol{x},y)) \frac{1}{1 + |\nabla I(\boldsymbol{x})|^\alpha}, \quad (3)$$

where $D_s(\boldsymbol{x},y)$ gives the distance from $\boldsymbol{x}$ to the classification boundary that divides points where $y(\boldsymbol{x}) = 0$ and $y(\boldsymbol{x}) = 1$. δ is the delta function which allows us to consider the gradient only for points on the segmentation boundary. In this paper, we set $\alpha = 2$.

This term will have values near 0 along a segmentation surface that follows a strong edge in the image data, and values near 1 in homogeneous areas. See Section 3.1 for details on how $\delta(D_s(\boldsymbol{x}, y))$ is implemented. We define the regional energy term as in [2] to measure how well each voxel inside and outside the proposed segmentation conforms to a respective intensity distribution. We derive the intensity distributions for foreground and background voxels by calculating the maximum likelihood Gaussian distribution parameters on the training data. For Gaussian foreground and background intensity distributions given respectively as $p(I(\boldsymbol{x})|Y=1) = \mathcal{N}(\mu_1, \sigma_1)$ and $p(I(\boldsymbol{x})|Y=0) = \mathcal{N}(\mu_0, \sigma_0)$, we can write the regional energy using Baye's theorem as

$$U_R(\boldsymbol{x}, y) = p(Y = y(\boldsymbol{x})|I(\boldsymbol{x})) = \frac{p(I(\boldsymbol{x})|Y=y(\boldsymbol{x}))}{p(I(\boldsymbol{x})|Y=0) + p(I(\boldsymbol{x})|Y=1)} \tag{4}$$

where we assume $p(Y=0) = p(Y=1) = 0.5$. Finally, since smooth surfaces have less area, we define a smoothness energy term based on the surface area of the segmentation around a point $\boldsymbol{x}$. With $N_{\boldsymbol{x}}$ representing a local 3D neighborhood around $\boldsymbol{x}$, we have

$$U_S(\boldsymbol{x}, y) = \iiint_{N_{\boldsymbol{x}}} \delta(D_s(\boldsymbol{z}, y))\, \mathrm{d}V. \tag{5}$$

2.2 Batch Query Active Learning Based on Planar Slices

Given an existing classification, heuristically the user should provide labels in the most uncertain regions in order to best improve the classification. Point by point labeling is too slow, so we instead use MPR to present the user with a plane of uncertain points so that they can be labeled simultaneously. Querying the user for multiple labels at the same time, as we do here, is known as a batch query [11]. Therefore, we seek the plane with maximal uncertainty, which we define as $\mathrm{argmax}_{\mathcal{P}}(U_{\mathcal{P}})$ where $U_{\mathcal{P}}$ is given as

$$U_{\mathcal{P}} = \iint_{\mathcal{P}} U(\boldsymbol{x})\, \mathrm{d}A = \int_{\infty}\int_{\infty} U(f_{\mathcal{P}}(u,v)) \left| \frac{\partial f_{\mathcal{P}}}{\partial u} \times \frac{\partial f_{\mathcal{P}}}{\partial v} \right| \mathrm{d}u\mathrm{d}v. \tag{6}$$

Here, $f_{\mathcal{P}}$ is a function mapping $\mathbb{R}^2$ to $\mathcal{P}$, and $\left| \frac{\partial f_{\mathcal{P}}}{\partial u} \times \frac{\partial f_{\mathcal{P}}}{\partial v} \right|$ is introduced with the change of variables. We evaluate (6) by calculating the uncertainty at sample points on $\mathcal{P}$. Since in general the uncertainty field may be an arbitrary scalar image, (6) may have many local maxima with respect to $\mathcal{P}$. We therefore run multiple iterations of gradient descent with randomized initial parameters (Section 3.1) in order to determine an approximate solution to $\mathrm{argmax}_{\mathcal{P}}\, U_{\mathcal{P}}$.

To that end, we first parameterize $\mathcal{P}$ by its normal $\boldsymbol{n}_{\mathcal{P}}$, and a point on $\mathcal{P}$, $\boldsymbol{p}_{\mathcal{P}}$. Let $f_{\mathcal{P}}(u,v) = \boldsymbol{p}_{\mathcal{P}} + u\boldsymbol{a} + v\boldsymbol{b}$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are orthonormal vectors perpendicular to $\boldsymbol{n}_{\mathcal{P}}$, then $\left| \frac{\partial f_{\mathcal{P}}}{\partial u} \times \frac{\partial f_{\mathcal{P}}}{\partial v} \right| = 1$. Making use of the chain rule, we calculate $\nabla_{\boldsymbol{n}_{\mathcal{P}}} U_{\mathcal{P}}$ as

$$\nabla_{\boldsymbol{n}_{\mathcal{P}}} U_{\mathcal{P}} = \int_{\infty}\int_{\infty} (u J^T_{\boldsymbol{a},\boldsymbol{n}_{\mathcal{P}}} + v J^T_{\boldsymbol{b},\boldsymbol{n}_{\mathcal{P}}}) \nabla_x U(f_{\mathcal{P}}(u,v))\, \mathrm{d}u\mathrm{d}v, \tag{7}$$

where $\nabla_x U(f_{\mathcal{P}}(u,v))$ is the gradient of the uncertainty field with respect to the spatial coordinates $\boldsymbol{x}$. We use the notation $J_{\boldsymbol{t},\boldsymbol{s}}$ to represent the Jacobian of $\boldsymbol{t}$ with respect to $\boldsymbol{s}$. Using the chain rule again, $\nabla_{\boldsymbol{p}_{\mathcal{P}}} U_{\mathcal{P}}$ is easily shown to be

$$\nabla_{\boldsymbol{p}_{\mathcal{P}}} U_{\mathcal{P}} = \int_{\infty}^{} \int_{\infty}^{} \nabla_x U(f_{\mathcal{P}}(u,v)) \, \mathrm{d}u\mathrm{d}v \tag{8}$$

since $J_{f_{\mathcal{P}},\boldsymbol{p}_{\mathcal{P}}} = I$. The maximal plane is presented to the user who will then classify it. This 2D segmentation is easy to do and well-studied in computer vision. There are many methods to choose from: [7,10,3] and others. All these can directly or indirectly provide a labeling of the plane.

3 Results

We experimentally verify our automatic plane suggestion approach by comparing it to randomly chosen, as well as user-chosen, slice planes. Since our approach is general enough to be applicable to a number of classifiers, we first describe our specific implementation in Section 3.1 before presenting qualitative and quantitative results in Section 3.2 and Section 3.3 respectively.

3.1 Implementation Details

We use the Random Walker (RW) [3] algorithm as our classifier (i.e. assigns a class label to each pixel), since it is capable of producing a probabilistic segmentation, it accepts as input a set of pre-labeled feature vectors, and it naturally applies to 3D images. To allow RW to run in a feasible amount of time on 3D images, we use techniques presented in [4] to develop a GPU RW implementation using NVIDIA's CUDA API. In order to keep the classifier consistent for our tests, we set the RW algorithm's only parameter to $\beta = 10$, a value that was found to produce good segmentations. The $D_s(x,y)$ distance values seen in (3) and (5) are implemented using distance maps. The $\delta(x)$ functions are approximated by $e^{-\frac{x^2}{2\sigma}}$, with $\sigma = 1$, giving a differentiable "soft delta" function. In our experimental software, users provide training data along slices using 2D Livewire, which are rasterized to seedpoints in the 3D image and then passed to RW as input. At any time after a segmentation has been computed the user can click a button to have a batch query slice presented to them. For the following studies, we set $\lambda_E = 0.80$, $\lambda_B = 0.05$, $\lambda_R = 0.15$ and $\lambda_S = 0$, which we found empirically to give good results and were justified because U_E exactly captures entropy while others less so. The plane that maximizes (6) is found by initializing a plane 36 times to a random orientation and ensuring the plane always passes through the the existing segmentation. For each plane initialization, gradient descent is iterated 250 times, and the optimal plane over the 36 runs is returned as the query plane.

3.2 Qualitative Results

We evaluate our method qualitatively by observing that it makes intelligent decisions for each slice suggestion. We validate by segmenting two different medical images in

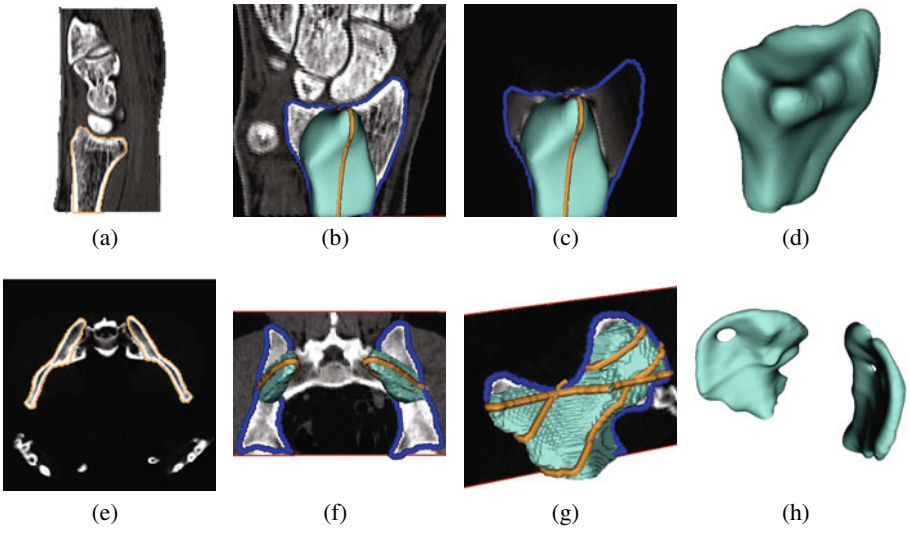


Fig. 1. Example snapshots during the AL interactive segmentation process. The first row shows the radius bone in a CT image. (a) A slice of the image, and a single initial contour in orange. (b) The results of the initial segmentation surface, the next AL suggested plane, and its user-provided classification (blue). (c) A slice through the uncertainty field. (d) The final 3D segmentation after 3 more active queries were labeled. The second row shows the segmentation of the iliac bones in a pelvis CT image. (e) A slice through the intensity image as well as the initial contours. (f) The first AL query slice. (g) The 3rd AL query slice. (h) The final segmentation after 12 slices have been classified.

Figure 1, demonstrating the active learning queries. In the first row, the radius bone in a wrist CT image is segmented. Figure 1(b) shows the first active query slice, along with its segmentation (blue). Note that the slice passes through a large portion of the radius bone that was, in error, not previously included in the initial segmentation. Looking at the uncertainty field through the same slice in figure 1(c), we see that high uncertainty is detected in the radius bone region not included in the initial segmentation. In the second row, the iliac bones of a pelvis CT image are segmented. We see in figure 1(f) that the query slice is targeting a large portion of the iliac bones that are contained in the initial segmentation. Figure 1(g) shows the third AL query slice beginning to focus on the left iliac bone, in order to better finalize that region of the segmentation.

3.3 Quantitative Validation

We test our method on eight different datasets: radius bone in a CT, femur bone in an MRI, liver in a CT, two images of a vastus lateralus muscle in an MRI, putamen in an MRI, tibia bone in an MRI, and vastus intermedius muscle in an MRI. Image sizes ranged from 150^3 to 190^3 pixels where suggestions took around 20 to 33 seconds, respectively. We report accuracy increase after each training data augmentation step. We first compare our proposed active query planes with random plane selection. Random plane selection involves randomly choosing a plane orientation and offset, constrained

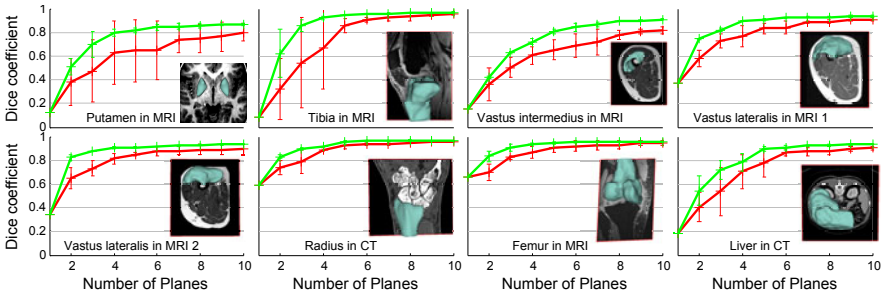
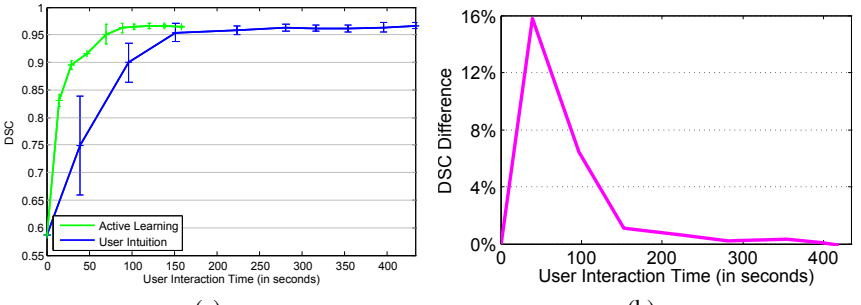


Fig. 2. A comparison of our AL technique to random plane selection. Each graph shows the DSC value of a segmentation versus the number of slices classified. The green curves show the results of using AL for plane selection, while the red curves show the result of randomly choosing planes. Sample slices of each segmented 3D volume are shown to clarify the image example context.



(a) (b)

Fig. 3. Results of the user study on the "Radius in CT" image. (a) An errorbar plot is given showing our AL approach improving the segmentation faster than user decisions. (b) The average difference in DSC percentage versus time.

to pass through the existing segmentation. Since both methods are non-deterministic, we re-segment each dataset three times and average the results. The accuracy is shown in terms of the Dice Similarity Coefficient (DSC) between the produced segmentation and the ground truth segmentation. For all results, in order to maintain consistency, we automatically derive the user segmentations over slice planes by extracting them from the ground truth segmentation. Figure 2 shows the comparison results between AL queries and random plane selection. Notice that the AL queries lead to higher DSC maxima, the DSC increases faster per user-segmented slice, and the average DSC increases by 10% on the first five plane suggestions (batch queries). Often, the first active query suggestion results in a better segmentation than multiple random slice planes.

Next we present the results of a 4 person user-study comparing our method of plane selection to human intuition on "Radius in CT" dataset chosen from above. Figure 3 gives a plot for the dataset showing DSC increase versus user interaction time where errorbars show the variance of the different users' output. Results clearly demonstrate how our AL approach significantly reduces the time needed to achieve any DSC value. Specifically, a total average reduction of 64% in user interaction time was achieved.

4 Conclusions

In this paper we formulate interactive 3D image segmentation in an AL framework. Our contributions include being the first to formulate interactive 3D image segmentation as a formal AL process. In doing so, we alleviate the user from choosing where to provide interactive input. In order to facilitate the batch queries, we automate assessment of uncertainty in the classification as well as the search through the uncertainty for the best query plane. Finally, we show our method to significantly reduce the required user input in interactive 3D image segmentation tasks through a user study. Future work includes incorporating this technique into other interactive segmentation algorithms and learning the energy functional weights automatically via training data.

References

1. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vision 22(1), 61–79 (1997)
2. Chan, T., Vese, L.: Active contours without edges. IEEE Trans. Image Proc. 10(2), 266–277 (2001)
3. Grady, L.: Random walks for image segmentation. IEEE Trans. Pat. Anal. and Mach. Intel. 28(11), 1768–1783 (2006)
4. Grady, L., Schiwietz, T., Aharon, S., Westermann, R.: Random walks for interactive organ segmentation in two and three dimensions: Implementation and validation. In: Duncan, J., Gerig, G. (eds.) MICCAI 2005. LNCS, vol. 3750, pp. 773–780. Springer, Heidelberg (2005)
5. Li, J., Bioucas-Dias, J., Plaza, A.: Supervised hyperspectral image segmentation using active learning. In: 2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, pp. 1–4 (2010)
6. Ma, A., Patel, N., Li, M., Sethi, I.: Confidence based active learning for whole object image segmentation. In: Gunsel, B., Jain, A., Tekalp, A., Sankur, B. (eds.) MRCS 2006. LNCS, vol. 4105, pp. 753–760. Springer, Heidelberg (2006)
7. Mortensen, E.N., Barrett, W.A.: Intelligent scissors for image composition. In: 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, pp. 191–198. ACM, New York (1995)
8. Olabarriaga, S., Smeulders, A.: Interaction in the segmentation of medical images: a survey. Med. Image Anal. 5(2), 127–142 (2001)
9. Pavlopoulou, C., Kak, A.C., Brodley, C.: Application of semi-supervised and active learning to interactive contour delineation. In: ICML 2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, pp. 26–33 (2003)
10. Rother, C., Kolmogorov, V., Blake, A.: GrabCut^{TM}: Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics 23, 309–314 (2004)
11. Settles, B.: Active learning literature survey. Tech. Rep. 1648, University of Wisconsin-Madison (2010)
12. Top, A., Hamarneh, G., Abugharbieh, R.: Spotlight: Automated confidence-based user guidance for increasing efficiency in interactive 3D image segmentation. In: Menze, B., Langs, G., Tu, Z., Criminisi, A. (eds.) MICCAI 2010. LNCS, vol. 6533, pp. 204–213. Springer, Heidelberg (2011)