Evaluating Segmentation Error Without Ground Truth

Timo Kohlberger[1], Vivek Singh[1], Chris Alvino[2], Claus Bahlmann[1], Leo Grady[3]

[1] Imaging and Computer Vision, Siemens Corp., Corporate Research and Technology, Princeton, NJ, USA
[2] American Science and Engineering, Billerica, MA, USA
[3] HeartFlow, Inc., Redwood City, CA, USA

Abstract. The automatic delineation of the boundaries of organs and other anatomical structures is a key component of many medical image processing systems. In this paper we present a generic learning approach based on a novel space of segmentation features, which can be trained to predict the overlap error and Dice coefficient of an arbitrary organ segmentation without knowing the ground truth delineation. We show the regressor to be much stronger a predictor of these error metrics than the responses of Probabilistic Boosting Classifiers trained on the segmentation boundary. The presented approach not only allows us to build reliable confidence measures and fidelity checks, but also to rank several segmentation hypotheses against each other during online usage of the segmentation algorithm in clinical practice.

1 Introduction

Measuring the quality of a segmentation produced by an algorithm is key to creating a deployable system and comparing the effectiveness of different algorithms to address a particular application. In fact, segmentation quality measures form the backbone for judging results of the segmentation challenges embraced by the medical imaging community in recent years (e.g., [8]). Additionally, these quality measures are key to publishing segmentation algorithms in order to demonstrate improved effectiveness of a new algorithm. Recent studies have shown that standard quality measures used in the community (or combinations thereof) serve as good proxies for human evaluation of segmentation quality in a clinical context [5,7].

The basic procedure for applying the existing quality measures is to create **ground truth** (manually segmented) structures and to compare those structures with algorithm-generated segmentations in terms of overlap or boundary differences. Although this procedure is effective for developing and comparing algorithms, there is no automated method for evaluating segmentation quality *after algorithm deployment* since there is no ground truth available after deployment (if there were, then a segmentation algorithm would be unnecessary). Consequently, in the field, our methods for evaluating segmentation quality are not usable due to a lack of ground truth segmentations to compare with. Figure 1 illustrates this difference.

The evaluation of segmentation quality after deployment serves a very different purpose than the evaluation of segmentation quality during algorithm development. During development, the purpose of the evaluation is to compare different

approaches or to optimize parameter settings. In contrast, on-line segmentation evaluation during deployment has several uses:

1. The evaluation can flag the user or system that a poor segmentation was obtained that requires manual review.
2. If a poor segmentation evaluation is obtained, the deployed system can try again to produce a better segmentation by re-running the segmentation with different algorithm parameters or a new algorithm entirely.
3. Every time a segmentation is required for a new dataset, several candidate segmentations may be generated on-line (e.g., in parallel) using different parameter settings and/or algorithms. The candidate segmentations are each evaluated and the segmentation with best evaluation score is then selected to return as output.

Several different types of popular segmentation algorithms are associated with measures that might be considered useful to evaluate segmentation in the absence of ground truth. For example, any of the family of optimization-based segmentation algorithms (e.g., level sets [17], graph cuts [2], random walker [9]) explicitly optimize an objective function to produce the desired segmentation. Therefore, a natural idea might be to use the energy of the output solution as an evaluation metric for segmentation quality. However, this energy of the minimal solution is unsuitable to evaluating segmentation quality since these algorithms are designed to compare *relative* energies of different segmentations and not to measure an *absolute* energy difference between a (possibly locally minimal) solution to the ground truth. Another class of popular segmentation algorithms utilizes learning to produce the segmentation. For these methods, a natural idea would be to use the outputs of the learning system as a confidence measure to perform on-line segmentation evaluation in the absence of ground truth. However, in Section 3 we demonstrate that the learning outputs of one popular learning algorithm, the Probabilistic Boosting Tree (PBT) [19], are poorly correlated with traditional measures of segmentation error when ground truth is known.

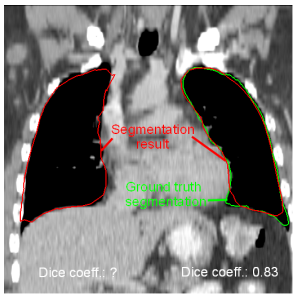

Fig. 1. With ground truth, quantifying the segmentation error is straightforward (left lung). This error relative to ground truth is essential during pre-development to select an algorithm and optimize parameters. In contrast, our method applies to the post-deployment situation where it is necessary to estimate error when no ground truth is available (right lung).

We adopt a hybrid approach to evaluating segmentation quality in the absence of ground truth. First, we calculate features to describe the output segmentation which are derived from the optimization-based segmentation literature. Effectively, we choose features by adopting every generic term in an objective function that we could find from an optimization-based segmentation paper. Second, we train a regression algorithm to predict the conventional segmentation error with respect to a known ground truth. Once trained, the regression algorithm can be used to predict the segmentation error from the calculated features *in the absence of ground truth.*

2 Method

First we introduce a novel space of shape and appearance features to characterize a segmentation. We then use these features to learn a predictor of segmentation error by training on error with respect to ground truth.

2.1 Using Energy Terms as Segmentation Features

We propose the following 42 shape and appearance features, many of which can be found as building blocks of popular energy-based or graph-based segmentation approaches. Thereby we remain agnostic about which feature choices worked well for the final regressor. The features we used can be broken down into five major categories: (weighted or unweighted) geometric features, intensity features, gradient features, and ratio features.

Note that in the following descriptions, we will use three-dimensional (3-D) terminology such as voxels, volume, surface area, and mean curvature, but it is understood that when applied to 2-D problems, the appropriate 2-D counterparts are implied, without loss of generality. In addition, all weights in these descriptions refer to the Cauchy distribution function applied to the appropriate image intensities differences, i.e., $w(I_1, I_2) = \frac{1}{1+\beta\left(\frac{I_1-I_2}{M}\right)^2}$, where I_1 and I_2 are two image intensities in question, β, which is set to 10^4 for all experiments, controls the sensitivity of the weight to intensity difference, and $M = \max_{(x,y)\in S} \|\nabla I(x,y)\|_1$ was the maximum L1 norm of all intensity gradients within the segmentation mask, S. The purpose of M is to normalize the weights. We will also define $w_+(I_1, I_2) = w(I_1, I_2)$ when $I_1 > I_2$ and $w_+(I_1, I_2) = 1$ otherwise. Likewise, we will define $w_-(I_1, I_2) = 1$ when $I_1 > I_2$ and $w_-(I_1, I_2) = w(I_1, I_2)$ otherwise.

Geometric features capture some measure of size of the segmentation mask $S \subset \mathbb{R}^3$, a concept dating back to some of the earliest works on image segmentation [1,16,4]. Of these, we chose: *volume*, defined as the number of voxels in the segmentation mask, $|S|$; *surface area*, the number of edges (assuming a graph structure with a 6-connected lattice) on the boundary of the segmentation, $\sum_{i,j:i\in S, j\in\bar{S}} 1$, where $\bar{S}$ is the set of voxels not in the segmentation mask; and *total curvature*, the sum of the mean curvature defined on the segmentation surface, $\sum_{i,j:i\in S, j\in\bar{S}} H(i,j)$, where $H(i,j)$ is the discretely computed mean curvature on the segmentation surface between voxels i and j and is locally computed as in [6].

Weighted geometric features are similar to the geometric features, but in addition the geometric measure is locally emphasized when intensity values are similar to each other and suppressed when local intensity values are dissimilar to each other. This concept has been pervasive in image segmentation since the work of Caselles *et al.* [3] and has been seen in many other recent works [9]. The geometric weights we use are based on local intensity in the image and are mapped via the Cauchy function $w(\cdot,\cdot)$ shown above. In the cases where we refer to voxel (or vertex $v \in V$) weight, we mean the average weight of all edges leaving that vertex, $w(v) = \frac{1}{D_v}\sum_{i:(v,i)\in E} w(I_v, I_i)$ where D_v is the degree of the vertex v. For weighted geometric features, we chose: *weighted volume*, the sum over the weights of all voxels, $\sum_{v\in S} w(v)$; *weighted cut*, the sum over the all edge weights

along the boundary of the segmentation $\sum_{i,j:i\in S,j\in\bar{S}} w(I_i, I_j)$; *weighted curvature* $\sum_{i,j:i\in S,j\in\bar{S}} w(I_i, I_j)H(i,j)$, the sum of the mean curvature weighted by the local edge weight; *low-hi weighted cut*, $\sum_{i,j:i\in S,j\in\bar{S}} w_+(I_i, I_j)$; and *hi-low weighted cut* $\sum_{i,j:i\in S,j\in\bar{S}} w_-(I_i, I_j)$ along the segmentation boundary.

Intensity features use various measures of the direct image intensities. Of these, we chose: *mean intensity* defined as $\mu_I = \frac{1}{|S|}\sum_{v\in S} I_v$; *median intensity* defined as $\text{median}(\{I_v : v \in S\})$; *sum of intensities* $\sum_{v\in S} I_v$; *minimum intensity* $\min_{v\in S} I_v$; *maximum intensity* $\max_{v\in S} I_v$; *interquartile distance* (defined as half of the difference between the 75th percentile and the 25th percentile values) of intensities; and *standard deviation* of the intensities $\frac{1}{|S|-1}\sum_{v\in S}(I_v - \mu_I)^2$.

Gradient features use various measures of the intensity gradients (local intensity changes). All intensity derivatives comprising these gradients are computed via central differences. Of these, we chose: *sum of the L1 norms of the gradients*, $\sum_{v\in S}\|\nabla I(v)\|_1$; *sum of the L2 norms of the gradients*, $\sum_{v\in S}\|\nabla I(v)\|_2$; *mean of the L1 norms of the gradients* $\frac{1}{|S|}\sum_{v\in S}\|\nabla I(v)\|_1$; *mean of the L2 norm of gradients* $\mu_g = \frac{1}{|S|}\sum_{v\in S}\|\nabla I(v)\|_2$; *median of the L1 norms of gradients* $\text{median}(\{\|\nabla I(v)\|_1 : v \in S\})$; *minimum L1 norm of all gradients* $\min_{v\in S}\|\nabla I(v)\|_1$; *maximum L1 norm of all gradients* $\max_{v\in S}\|\nabla I(v)\|_1$; *interquartile distance of the L1 norms of the gradients*; *standard deviation of the L1 norms of gradients*; and *the standard deviation of the L2 norms of gradients* $\frac{1}{|S|-1}\sum_{v\in S}(\|\nabla I_v\|_2 - \mu_g)^2$.

We opt to explicitly include a selection of features that were ratios of our other features. The intent is not to be completely comprehensive, but rather to use domain knowledge of segmentation problems to explicitly choose combinations that the literature and our experience told us would be good indicators of segmentation performance. The ratio features are simply the ratio of two features above. We only include ratios that either we believe to be meaningful, or have appeared in the segmentation literature thus far. Several fall into the category of cut divided by volume, a concept that has appeared throughout the history of segmentation in various forms [13,10]. Of these, we chose: *all four weighted and unweighted combinations of cut divided by volume*; *all four combinations of low-hi weighted cut or hi-low weighted cut divided by unweighted or weighted volume*; *weighted cut divided by unweighted cut*; *all four combinations of low-hi weighted cut or hi-low weighted cut divided by unweighted or weighted cut*; *blur index* defined as *sum the L2 norms of the gradients* divided by *sum of the L1 norms of the gradients*; *curvature over unweighted cut*; and *weighted curvature over unweighted cut*.

Some of the features, such as the geometric features and most of the intensity-based features, are not meant to be discriminative alone. Rather, they are intended to lend context about the expected values for some of the other more discriminative features for a given candidate segmentation. Our intention is to extract features that might be relevant *independent* of the classifier method, and then to let feature selection or the classifiers determine how the features would be used.

2.2 Learning to Predict Segmentation Error

Based on this novel space of shape and appearance features, we propose to use non-linear regressors in order to separately approximate different segmentation metrics.

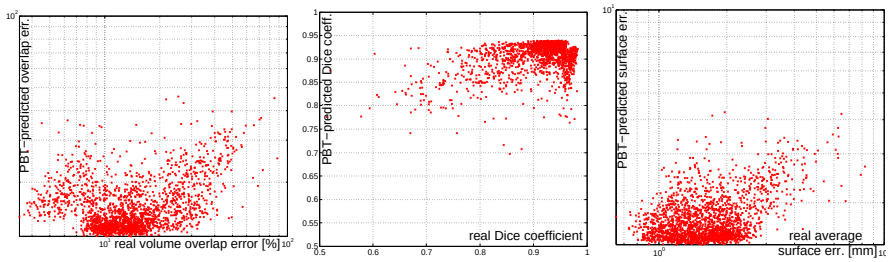

Fig. 2. Real segmentation errors (x-axis) versus linearly regressed PBT-probabilities (y-axis). Correlations coefficients (left to right): 0.45, 0.48, 0.49. Max. surf. err: 0.29. (Note that for readability we have adopted linear and log scaling where appropriate.)

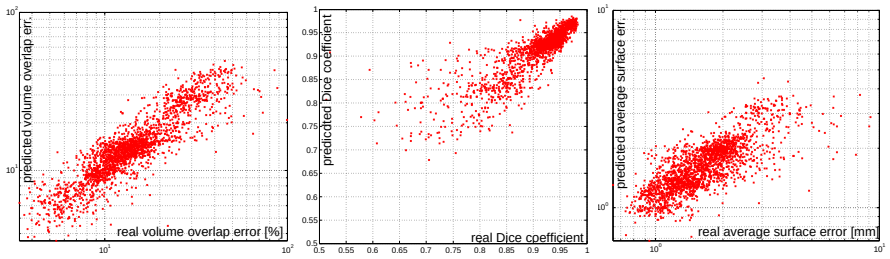

Fig. 3. Real segmentation errors (x-axis) versus SVM regressor-predicted ones (y-axis). Correlations coefficients (fr. l. to r.): 0.85, 0.79, 0.56. Max. surface error: 0.69.

Specifically, we treat the 42 features as independent variables and each of the error metrics, which we will define below, as dependent variables.

In order to obtain a comprehensive quantification of the segmentation error relative to the ground truths, we employ four different error metrics. Let $G, S \subset \mathbb{R}^3$ denote the set of points of the ground truth segment and the computed segment, respectively. As first metric we use the popular *volumetric overlap error* [12]: $E_O(S,G) = 1 - (|S \cap G|)/(|S \cup G|)$, which is 0% for a perfect segmentation (i.e. $S = G$) and 100% if the computed segment does not overlap with the ground truth at all. As a second volumetric measure we employ the *Dice coefficent*: $E_D = 2\,|S \cap G|\,/(|S| + |G|)$, which is similar to the first one, and assigns 1 to a perfect segmentation and 0 to a completely failed one. In practice, S and G are typically represented as binary masks on a regular grid. In case segmentations are represented by surfaces, such masks can be obtained by voxelization. Besides these volumetric measures, we also compute the symmetric surface-to-surface metrics. In particular the well-known *Hausdorff distance*: $E_H = \max\{\sup_{x\in\partial S} \inf_{y\in\partial G} d(x,y), \sup_{x\in\partial G} \inf_{y\in\partial S} d(x,y)\}$, which measures the maximum of the Euclidean distance ($d(x,y) := |x-y|_{l^2}$) of each point on the computed segmentation surface ∂S to the ground truth surface ∂G and vice versa. Besides the maximum of the minimum per-vertex surface distances, we also gauge their mean by computing the *average surface error*: $E_S = \frac{1}{2}\left(\frac{1}{|\partial S|}\sum_{x\in\partial S} \min_{y\in\partial G} d(x,y) + \frac{1}{|\partial G|}\sum_{y\in\partial G} \min_{y\in\partial S} d(x,y)\right)$.

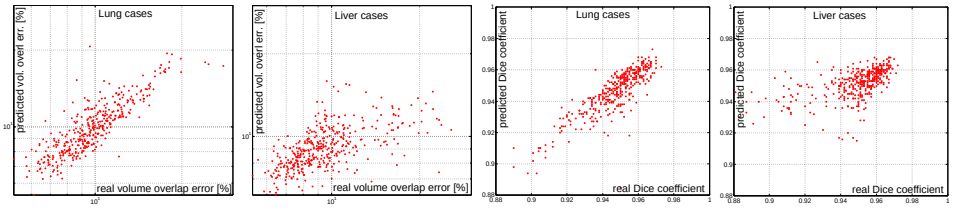

Fig. 4. Real segmentation errors (x-axis) versus predictions (y-axis) for 378 lung (left or right) and 411 liver segmentation from a level set approach [14] (ten-fold cross-validation).

To perform the learning, we experimented with commonly known linear and non-linear regression approaches, all of which are available in the Weka tool [11]. Thereby we found an SVM regressor with a normalized polynomial kernel $\langle \mathbf{x}, \mathbf{y} \rangle / \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle \langle \mathbf{y}, \mathbf{y} \rangle}$ with $\langle \mathbf{x}, \mathbf{y} \rangle = (1 + \mathbf{x} \cdot \mathbf{y})^2$ and an SMO-type optimizer [18] (with $C = 1$) to yield the highest correlations factors using a ten-fold-cross-validation.

3 Experiments

In this section we will address the following questions: Can the response of a commonly used boundary classifier be used to predict the above error metrics? How much better does the proposed regressor predict the segmentation errors than the boundary classifier probabilities? Can the new predictors estimate the error of a typical optimization based segmentation too? How do they perform on individual organs instead of of a whole collection? If we use the regressor to classify results into good and bad, how high is the error rate of this classification?

In order to address the first question, we used the machine learning-based organ segmentation approach described in [20] and [15] as reference. The last stage of this segmentation approach comprises a hierarchical boundary detection, where Probabilistic Boosting Tree boundary classifiers [19] are queried along the normals of an approximate segmentation mesh and the mesh vertices are then being placed at the location of maximum classifier response. We trained this method on eight different organs or organ parts (both referred to as "organs" in the following), for which we had the following number of ground truth segmentations: liver: 411, left lung: 187, right lung: 191, right kidney: 341, left kidney: 379, bladder: 311, prostate: 204, rectum: 149. All of those were generated by manual editing from a pool of 950 different CT scans that cover a variety of different patient anatomies, scanning protocols and parameters (slice resolution range: 1–5mm). For each ground truth segmentation, we applied the PBT and not only recorded the detected segmentation surface, but also the mean of the classifiers' probabilities over each segment surface. Subsequently, we used a linear regressor in order to fit the 2173 probability values to each of the four error metrics. See results in Figure 2. Surprisingly, the mean probabilities and any of the four metrics are only weakly correlated. This observation is in spite of the overall good segmentation accuracies of the system. Despite the individual boundary classifier responses provide a good prediction of the true boundary location on a local scale, in aggregation they seem to be a poor predictor for the overall accuracy of a segmentation shape.

Table 1. Confusion matrices when thresholding the regressor-predicted volume overlap error E_O. Left: for 2×2173 segmentations on all organs using results both from [15] and [14]. Right: for 377 left/right lung segmentations using [14] only.

# of true cases	# of predicted cases $E_O \leq 10\%$	# of predicted cases $E_O > 10\%$
$E_O \leq 10\%$	867	373
$E_O > 10\%$	255	2851

# of true cases	# of predicted cases $E_O \leq 10\%$	# of predicted cases $E_O > 10\%$
$E_O \leq 10\%$	178	39
$E_O > 10\%$	24	136

By contrast, when training a regressor as described in Section 2.2, we observe significantly better correlations between the true and the predicted errors of the PBT-based segmentations, especially for the volume overlap error and the Dice coefficient. See results in Figure 3 using ten-fold cross-validation. In order to investigate a possible bias of the predictors towards PBT-type segmentation errors, we also ran them on segmentations generated by a level set approach [14], which relies on a volumetric shape representation. However, also for those segmentation results we observed very similar error prediction performances, with correlation coefficients being the same as for the PBT method up to the first decimal. In a next step, we trained and tested the regressors on different organ-specific subsets and discovered significant performance differences. For the lungs, for example, the real and predicted overlap and Dice errors are both correlated by a factor of 0.85 each, whereas for the liver only with 0.54, see Fig. 4. Finally, encouraged by the overall good correlation factors between the SVM regressor and the overlap error metric, we investigated the use of the former in classifying segmentations results into acceptable ($E_O \leq 10\%$) and non-acceptable segmentations ($E_O > 10\%$). Results in Table 1 show that the proposed method is capable of classifying into these two classes with low false positive rates (lower left entry) over all organ classes, as well as, e.g., for the lungs only.

4 Conclusion

We presented a method for predicting segmentation error *in the absence of ground truth* based on learning a classifier from errors measured against ground truth. Our method used a series of features derived from objective functions found in the literature for optimization-driven segmentation algorithms and trained our classifier to predict error measured against ground truth using standard error metrics used in the literature to compare segmentation quality. Despite training our classifier on segmentations for 8 very different organs, a strong correlation was observed between the predicted and actual errors when applied to an unseen test set. Furthermore, we demonstrated that a popular learning algorithm (PBT) does not provide the same power to predict segmentation quality.

A method for predicting segmentation error for on-line segmentations after deployment has many uses to improve final segmentation quality (by retrying poor segmentations or choosing the best segmentation from multiple algorithms run in parallel) or to request user review for a segmentation. We believe that the problem of predicting segmentation error without ground truth holds many future opportu-

nities, such as the development of new feature sets, training across modalities and systems that can localize the source of segmentation error.

References

1. Blake, A., Zisserman, A.: Visual Reconstruction. MIT Press (1987) 3
2. Boykov, Y., Jolly, M.P.: Interactive organ segmentation using graph cuts. In: Proc. of MICCAI 2000. pp. 276–286. Pittsburgh, PA (2000) 2
3. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision 22, 61–79, 1997 3
4. Caselles, V., Kimmel, R., Sapiro, G., Sbert, C.: Minimal surfaces based object segmentation. IEEE Trans. on Pat. Anal. and Mach. Int 19(4), 394–398, 1997 3
5. Deng, X., Zhu, L., Sun, Y., Xu, C., Song, L., Chen, J., Merges, R., Jolly, M., Suehling, M., Xu, X.: On simulating subjective evaluation using combined objective metrics for validation of 3d tumor segmentation. In: Proc. of MICCAI. pp. 977–984 (2007) 1
6. El-Zehiry, N., Grady, L.: Fast global optimization of curvature. In: Proc. of CVPR 2010. IEEE Computer Society, IEEE (June 2010) 3
7. Frounchi, K., Briand, L.C., Grady, L., Labiche, Y., Subramanyan, R.: Automating image segmentation verification and validation by learning test oracles. Information and Software Technology 53(12), 1337–1348, 2011 1
8. Ginneken, B.V., Heimann, T., Styner, M.: 3D segmentation in the clinic: a grand challenge. In: Proc. of MICCAI Workshop. pp. 7–15 (2007) 1
9. Grady, L.: Random walks for image segmentation. IEEE Trans. on Pat. Anal. and Mat. Intel. 28(11), 1768–1783, Nov 2006 2, 3
10. Grady, L., Schwartz, E.L.: Isoperimetric graph partitioning for image segmentation. IEEE Trans. on Pat. Anal. and Mach. Int. 28(3), 469–475, 2006 4
11. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA Data Mining Software: An Update. SIGKDD Explorations, 1(11) 6
12. Heimann, T., van Ginneken, B., Styner, M.: Comparison and evaluation of methods for liver segmentation from CT datasets. IEEE Trans. on Medical Imaging 28(8), 1251–1265, 2009) 5
13. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(8):888–905, 2000 4
14. Kohlberger, T., Sofka, M., Zhang, J., Birkbeck, N., Wetzl, J., Kaftan, J., Declerck, J., Zhou, S.: Automatic multi-organ segmentation using learning-based segmentation and level set optimization. In: Proc. of MICCAI 2011. vol. 6893, pp. 338–345 6, 7
15. Ling, H., Zhou, S., Zheng, Y., Georgescu, B., Suehling, M., Comaniciu, D.: Hierarchical, learning-based automatic liver segmentation. In: Proc. of CVPR 2008. pp. 1–8, 2008 6, 7
16. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. Pure and Appl. Math. 42, 577–685, 1989 3
17. Sethian, J.A.: Level Set Methods and Fast Marching Methods. Cambridge University Press (1999) 2
18. Shevade, S., Keerthi, S., Bhattacharyya, C., Murthy, K.: Improvements to the SMO algorithm for SVM regression. IEEE Trans. on Neural Networks, 11(5), 1188–1193, 2000 6
19. Tu, Z.: Probabilistic boosting-tree: learning discriminative models for classification, recognition, and clustering. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. vol. 2, pp. 1589–1596 Vol. 2 (2005) 2, 6
20. Zheng, Y., Barbu, A., Georgescu, B., Scheuering, M., Comaniciu, D.: Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features. TMI 27(11), 1668–1681, 2008 6