Random Forest Classification for Automatic Delineation of Myocardium in Real-Time 3D Echocardiography

Victor Lempitsky[1], Michael Verhoek[2], J. Alison Noble[2], and Andrew Blake[1]

[1] Microsoft Research Cambridge
victlem@microsoft.com, ablake@microsoft.com
[2] University of Oxford
mverhoek@robots.ox.ac.uk, noble@robots.ox.ac.uk

Abstract. Automatic delineation of the myocardium in real-time 3D echocardiography may be used to aid the diagnosis of heart problems such as ischaemia, by enabling quantification of wall thickening and wall motion abnormalities. Distinguishing between myocardial and non-myocardial tissue is, however, difficult due to low signal-to-noise ratio as well as the efficiency constraints imposed on any algorithmic solution by the large size of the data under consideration. In this paper, we take a machine learning approach treating this problem as a two-class 3D patch classification task. We demonstrate that solving such task using *random forests*, which are the discriminative classifiers developed recently in the machine learning community, allows to obtain accurate delineations in a matter of seconds (on a CPU) or even in real-time (on a GPU) for the entire 3D volume.

1 Introduction

Fast and accurate automatic delineation of the myocardium in echocardiograms can be a valuable aid in the assessment of heart-related diseases and abnormalities. Such a visual aid is particularly important in 3D echocardiography, where the visualization and comprehension of the data can be especially hard. 3D cardiography is however a challenging modality to work with due to low signal-to-noise ratio, unpredictable speckle patterns, and large variability in shape and appearance between different subjects. A large number of tissues that have similar appearance to the myocardium such as adjacent muscles or bright vessel walls complicate the discrimination even more. Finally, the sheer amount of the data contained in a 3D echocardiography study should ideally be processed in a matter of a few seconds (ideally in real-time), in order for an algorithm to be useful in clinical practice. Most (if not all) methods (such as [1]) in the published literature fail to meet this criterion.

Automatic delineation of the myocardium in 3D echocardiograms can be regarded as a binary classification problem, where each voxel should be assigned to either myocardium or non-myocardium. In the paper, we investigate a discriminative approach to this classification task. Thus, our classifier is trained to model the

N. Ayache, H. Delingette, and M. Sermesant (Eds.): FIMH 2009, LNCS 5528, pp. 447–456, 2009.

posterior probability of the class given the data directly, without explicit modeling of either the statistics of the shape of the heart or the physics of the imaging process. Furthermore, the non-parametric nature of the classifier we use means that no restrictive assumptions are made about the form of the posterior. Our system is thus much simpler than many of the competing approaches. Importantly, as the output of our method is "soft" (posterior probability for each voxel), it could be used as an input for a more complex algorithm which can incorporate richer geometric information, such as deformable templates, level sets, or graph cuts.

The particular discriminative classifier that we use is the Random Forest classifier [2,3]. Our choice is motivated by the computational efficiency that random forests exhibit at both runtime and training time, as well as their growing popularity for similar segmentation tasks in computer vision (see e.g. [4,5,6]). Capitalizing on the efficiency of random forests, the current CPU implementation of our system is able to accomplish delineation of the full 3D volume in a few seconds. A GPU implementation along the lines of [7] is highly likely to achieve real-time speed for the full 3D volume.

In terms of accuracy, Random Forests have an excellent track record for various machine learning problems on a par or even better than boosting method or max-margin classifiers like SVMs [8]. Towards this end, our approach achieves a considerable delineation accuracy (92% true positive rate at 8% false positive rate).

2 Related Work

Traditional segmentation methods have been applied to cardiac chambers [9], including level sets [10] and active contours [11]. However, the literature on segmentation of the myocardium is limited. Myocardial segmentation has been performed [12] by first finding the endocardium in 2D+T echocardiographic slices, then locating the epicardium by searching outward along normals to the endocardium, using PCA shape and motion models. Others used level sets [1] in 2D MRI slices to find two coupled contours, one for the endocardium and one for the epicardium. Coupled evolving contours were also employed in [13], on 3D+T echocardiography data, using an incompressibility constraint. Compared to our proposed method, these segmentation algorithms are far from real-time.

A useful feature when segmentating myocardium from blood pool in echocardiography is tissue characterization, by fitting a distribution to the histogram of a given signal, which may be either the RF (radio frequency) signal or the intensity in the B-mode image. The results of fitting to several different distributions have been compared, for the RF signal [14], and for B-mode images [15].

Many researchers have investigated the delineation of tissues in ultrasound using discriminative classifiers, typically SVMs or neural networks [16,17]. These machine learning techniques have an inherent limitation for the tissue classification tasks, as each patch has to be described by low- to moderate-dimensional descriptor. This means that a preliminary, typically hand-crafted, feature selection step leading to the loss of information has to be performed. Furthermore, these methods tend to use features that are too expensive to compute for 3D images (e.g. Gabor filters).

Recently, a number of machine learning approaches have been developed, which are able to work with very high-dimensional representations by unifying the feature selection and supervised learning tasks. The most widely known learning framework of this kind is boosting [18], and several papers on ultrasound image segmentation have considered this learning framework (e.g. [19,20]). In this paper, we focus on another framework called *random forests* [3], and investigate the use of this framework for tissue classification in 3D echocardiograms. Compared to boosting, random forests have been demonstrated to achieve a comparable accuracy, while being faster [8]. Random forests also proved to be more robust to significant overlaps between classes and noise in the training labels [3], which is a scenario that is highly relevant to ultrasound classification. Finally, random forests can be effortlessly applied to multi-class classification, whereas boosting methods typically resort to binary classification reductions (such as training several *one-vs-all* binary classifiers). Although we do not investigate multi-class feature classification in this work, this may be useful when several types of cardiac tissues are considered.

Random forests [2,3] have been used for a large number of classification as well as regression tasks. A typical random forest consists of a set of binary decision trees [21]. During training, each non-leaf node in each tree is assigned a binary test that is applicable to any data sample. Depending on the result of the test, a sample can go to one of the two children of a given non-leaf node. This way, a sample can be passed through each of the trees, starting from its root and ending up in one of its leaves.

Random forests are trained in a supervised way. Training involves tree construction as well as assigning to each leaf node the information about the training samples reaching this leaf node, e.g. the class distribution in the case of classification tasks. At runtime, a test sample is passed down all the trees of the forest, and the output is computed by averaging the distributions recorded at the reached leaf nodes.

It has been shown [2,3,8] that assembling (bagging) together several trees trained in a randomized way achieves superior generalization and stability compared to a single deterministic decision tree. The randomization is achieved, firstly, by training each tree on a random subset of the training data, and, secondly, by considering a random subset of possible binary tests at each non-leaf node. Among this random subset, the training procedure picks the binary test that splits the training samples in the optimal way.

To the best of our knowledge, random forests have seen a very limited use in medical imaging so far [22,23,24]. However, in the computer vision community, several groups of authors [5,6,4] have recently proposed to use random forests for semantic segmentation of photographs, and our method has a lot of similarities with those approaches.

3 Delineation Using Random Forests

Problem setting. We formalize our problem as a binary classification of voxel samples. Each voxel sample $\mathcal{V}$ is thus described by the voxel coordinates

$\mathbf{x} = (x_1, x_2, x_3)$ in the 3D echocardiogram $\mathbf{I}$ it is drawn from. At runtime, each voxel sample is classified into myocardium or non-myocardium based on the appearance of the fixed-size 3D neighborhood around it as well as its position in the image. Incorporation of the position information is important (as demonstrated in the experiments below) as a cardiologist positions the heart with respect to the echo-volume roughly in the same way. In our approach, the meaning of *"roughly in the same way"* is to be learned from training data. The position information only acts very weakly as a shape model. Note that a sophisticated shape model derived from healthy hearts will not necessarily represent the shape of a diseased heart. We are also bound to learn the inter-dependencies between the position and the appearance, as the typical appearance of myocardial tissue tends to change with the position in an echocardiogram.

As we learn to classify voxels in a supervised setting, we assume that in the training step we are given a collection of labeled voxels $\{\mathcal{V}^i = (\mathbf{I}^i, \mathbf{x}^i, c^i)\}$, where $\mathbf{I}$ is one of the training echocardiograms, $\mathbf{x}^i$ is the position of the i-th training voxel, whereas c^i is its *class label* provided by an expert, which can be either 1 (if the voxel is annotated as myocardium) or 0 (if the voxel is annotated as non-myocardium).

At test time, the position $\mathbf{x}$ and the appearance of the echocardiogram $\mathbf{I}$ around the voxel are observed, and our goal is to infer the class label c. In this way, given an echocardiogram, we can assign a label c to each voxel according to the prediction of our classifier. As the classifier we use has a probabilistic nature, the labels assigned at run-time would span the interval $[0; 1]$ rather than being binary. Such "soft" labeling conveys the information uncertainty when used to highlight the myocardium during visualization. The uncertainty information is also likely to boost the performance if labeling is used as an input to some further geometric integration step (e.g. level sets, graph cuts, deformable template fitting, etc.). However, if the "hard" binary labeling is required straight away, the voxel labels can be thresholded at some fixed value (e.g. 0.5).

Trees construction. Our random forests consist of a collection of randomized binary trees and below we discuss how these trees are constructed. The construction proceeds in a top-down fashion, starting from the root. At each node, a *training collection* of labeled voxels received from its parent is considered (the root node receives a randomly drawn subset of the size N_{train} of our full training set). A *binary test* is then chosen to minimize the uncertainty of the class labels in the subsets, it splits the training collection into (the exact choice method is discussed below). Once the binary test is chosen, the training collection of voxels is split in the two subsets according to this binary test. The two children nodes are then created and the two subsets are passed to them as their training collections, so that the recursion proceeds.

The recursive tree construction stops either when a certain maximal tree depth $D_{\max}$ is reached or when all voxels in the training collection have the same label. At this point, the node is declared a leaf. Once the tree is constructed (based on the drawn subset of the labeled voxels), we take all available labeled voxels and pass each of them through the tree. For each leaf node, we then record the average class label of the labeled voxels that have reached this node. This average class label can be interpreted as the posterior probability of a voxel being myocardium, given the fact that this voxel falls into the leaf.

Binary tests. Let us now discuss the way the binary tests are chosen. As the voxel description has two components, namely, the position in the volume and the neighborhood appearance, we consider two types of tests. The *appearance tests* are defined as:

$$t_{\text{appearance}}^{a_1,a_2,a_3,\tau_a}(\mathbf{x},\mathbf{I}) = \begin{cases} 0, & \text{if} \sum\limits_{p_1=x_1-a_1}^{x_1+a_1} \sum\limits_{p_2=x_2-a_2}^{x_2+a_2} \sum\limits_{p_3=x_3-a_3}^{x_3+a_3} \mathbf{I}(p_1,p_2,p_3) > \tau_a \\ 1, & \text{otherwise.} \end{cases} \quad (1)$$

An appearance test thus compares a sum of the intensities over an axis-aligned box of the size $(2a_1+1) \times (2a_2+1) \times (2a_3+1)$ centered at the voxels with some threshold τ_a, where the parameter values a_1, a_2, a_3, τ_a are chosen in a randomized fashion as discussed below. Note that such box sums can be evaluated extremely efficiently using just 7 additions and subtractions irrespective of the size of the box using the *integral image* technique [25].

The second type of the tests is the *position test* defined as:

$$t_{\text{position}}^{i,\tau_p}(\mathbf{x}) = \begin{cases} 0, & \text{if } x_i > \tau_p \\ 1, & \text{otherwise.} \end{cases} \quad (2)$$

This test, thus, simply compares the ith coordinate of the voxel with a threshold τ_p, where i, τ_p are chosen randomly as discussed below.

Given a collection of labeled voxels $\mathcal{C} = \{\mathcal{V}^i, c^i\}$, we define the *uncertainty* of its labels as:

$$U(\mathcal{C}) = |\mathcal{C}| \cdot \mathit{Entropy}\left(\{c^i\}\right) = |\mathcal{C}| \left(\bar{c} \cdot \log \bar{c} + (1-\bar{c}) \cdot \log(1-\bar{c})\right) \quad (3)$$

where $\bar{c}$ denotes the average class label over collection. Thus, the uncertainty is maximal for the collections where all voxels are from the same class and minimal (for a given size) for the collections where class labels are split into equal halves.

When choosing a split for a training collection $\mathcal{C}$, we generate N_{tests} random tests. We choose two thirds of our tests to be the appearance tests and one third to be the position tests. When generating an appearance test, we sample a_1, a_2, a_3 independently and uniformly from the interval $[0; R]$ from zero to the neighborhood radius R. We then pick the threshold τ_a by averaging the respective box sums from (1) for the two randomly chosen samples from the training collection. When generating a position test, we pick one of the three dimensions with equal probability and we pick the threshold τ_p equal to the respective coordinate of a randomly chosen sample from the training collection.

For each of the generated tests, we consider the two subsets, it splits the training collection into. We then pick a test, which has the smallest sum of the uncertainties (3) for these subsets. Such a choice ensures that the uncertainty in labels decreases quickly towards the leaves of the tree.

There are two properties worth noting. Firstly, despite the extreme simplicity of the position tests the superposition of such tests at different depths of the tree allows to learn a non-trivial shape prior. Secondly, as the position and appearance tests are interleaved within a tree, we are able to learn the inter-dependencies between the shape and appearance.

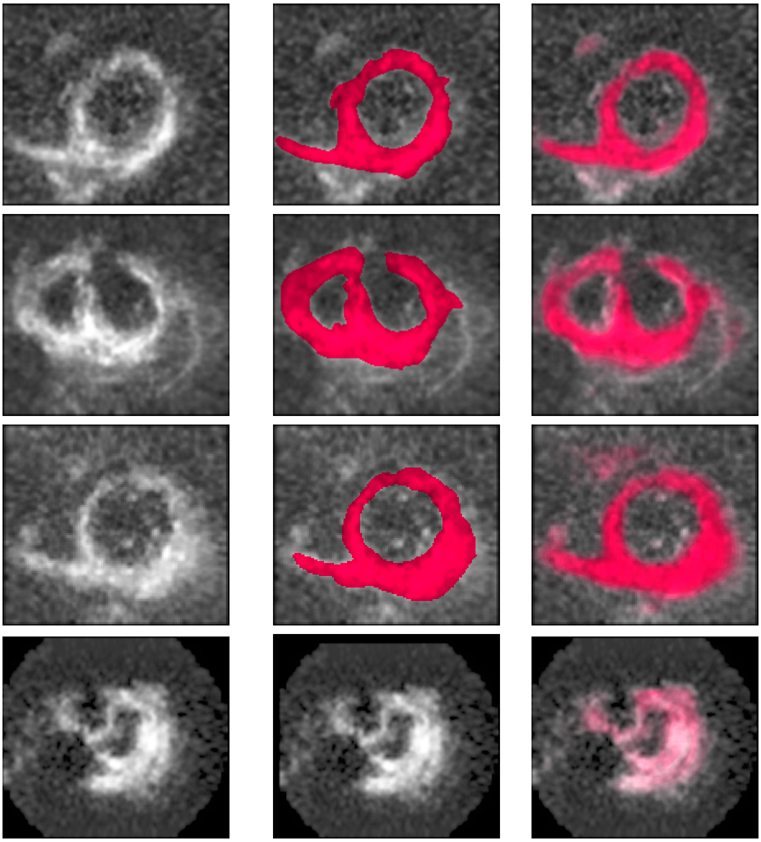
Annotating a voxel using a random forest. Given a random forest, i.e. several trees that were independently trained as discussed above, we can classify a new voxel $\mathcal{V}$ located at the position $\mathbf{x}$ in a previously unseen echocardiogram $\mathbf{I}$. To do that we simply pass the voxel $\mathcal{V} = (\mathbf{x}, \mathbf{I})$ through every tree. Given a set of leaves the voxel ends up in different trees, we simply average the class probabilities recorded in these leaves.

4 Experiments

Experimental protocol. We have evaluated the developed method on several 3D echocardiograms (Figure 1). Towards this end, we have obtained 14 3D+T echocardiograms from different healthy subjects with the myocardium for left and right ventricles delineated by an expert. The echocardiograms were recorded at the Oxford John Radcliffe Hospital, using a Philips iE33 ultrasound system. The echocardiograms had a spatial resolution of 0.93 mm × 0.94 mm × 0.85 mm or 0.82 mm × 0.84 mm × 0.76 mm, depending on the settings of the ultrasound system. We chose to use the end-systolic frame out of each sequence. In these 3D echocardiograms, every fifth short-axis slice was annotated, using a Wacom Cintiq interactive pen display and a Matlab script to record an endocardial and an epicardial contour in each slice. Papillary muscle was regarded as belonging to the blood pool. In places where wall data appeared to be missing due to attenuation or being parallel to the beam, the expert used their knowledge of the shape of the myocardium to construct the contours. The expert could view orthogonal long-axis slices simultaneously to judge the manual segmentation in those views. As the echocardiograms contained significant variations in brightness, we performed a histogram equalization step, so that histograms of all echocardiograms were approximately matched to the histogram of one of the examples. This pre-processing step was beneficial for our method, as the appearance tests (1) employed in our implementation are not invariant to brightness changes.

There are four major parameters in our method that have to be adjusted: the number of training samples for each tree N_{train}, the maximal tree depth $D_{\max}$, the number of random binary tests generated for each node N_{tests}, and the radius of the neighborhood R. Therefore, we singled out the three last echocardiograms and used them for validation, while the first eleven echocardiograms were used for training. As a result of the validation step, the values of parameters were set as follows: $N_{\text{train}} = 100000$, $D_{\max} = 16$, $N_{\text{tests}} = 30$, $R = 32$.

After the parameter tuning, we have evaluated our method on the first eleven echocardiograms. We tested the method on each of the eleven echocardiograms, training a random forest containing 20 trees on the remaining ten datasets (so



sample short-axis slices expert annotations random forest results

Fig. 1. Qualitative results. For the examples of short-axis slices, the random forest predicts "soft" delineations that closely match the delineations performed by an expert. As the bottom row demonstrates, our classifier still "fires" on the bright vessel walls, yet the response is relatively weak.

that training and test data are not mixed). Using the trained forests, we obtained a "soft" segmentation for every voxel in each of the eleven datasets. We then evaluated the accuracy of the hard segmentation obtained with a sequence of thresholds. For each value of threshold, we summed up the number of true positives (voxels correctly labeled as myocardium) and the number of all positives (voxels labeled as myocardium). In Figure 2, we present the results of our method in the form of a recall-precision and ROC-curves [26], generated by varying the probability threshold.

As a baseline, we provide the recall-precision curve for thresholding the raw values of the intensity (note, that this simple approach is likely to benefit a lot from the histogram equalization pre-processing). Although the muscular myocardial tissue has a tendency to be brighter than the surrounding tissues, it can be clearly seen that brightness is not a powerful enough cue in this case (one may wonder if a simple intensity thresholding would benefit from prefiltering the image with a low-pass filter, but in our experiments such prefiltering only worsened the performance of thresholding). We have also tried thresholding of the shifted Rayleigh parametric image obtained by estimating the σ^2 parameter in the shifted Rayleigh distribution, which was used previously [13] to characterize the blood pool and myocardium in B-mode images. We estimated the parameter in a neighborhood of 5^3 voxels around each voxel. The thresholding of the shifted Rayleigh parameter performed slightly better than raw intensity thresholding.

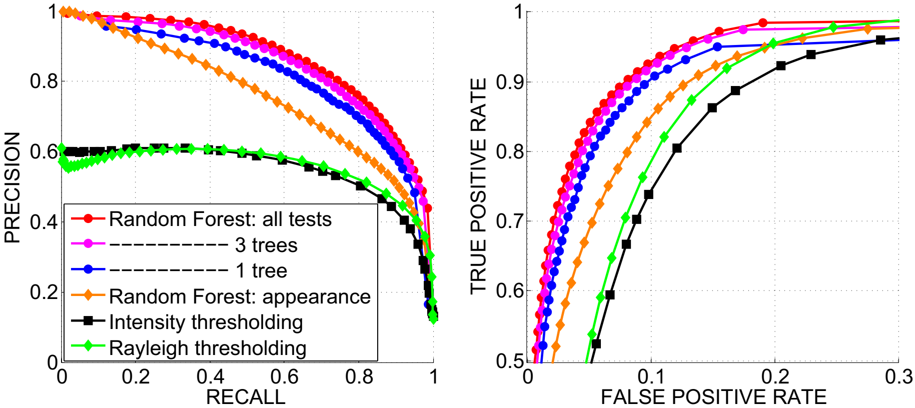


Fig. 2. Quantitative results. Recall-precision (left) and ROC (right) curves demonstrate the significant advantage of our method over the thresholding of raw intensity and of the shifted Rayleigh parameter. As an aside, removing the position tests (Random Forest-A curve) deteriorates the performance of our method much more than reducing the number of trees.

Variations of our method. Figure 2 also demonstrates the recall-precision and ROC curves for the variation of our method, where only appearance tests (1) were allowed, while position tests (2) were not considered. As can be seen (*Random Forest-A* curve), the performance of our method degraded considerably, suggesting that the rough knowledge about the positioning of the myocardium in the echo-volume was picked up and utilized by the training algorithm. It can be observed at the same time that even without the position tests, random forest based classification performs significantly better than thresholding, which suggests that a sophisticated model of appearance can be learned using random forests.

Finally, we demonstrate the performance of our method using a different number of trees smaller than 20. As can be seen, while the performance of a single tree is considerably worse than that of the whole forest, this performance quickly saturates as the number of trees is increased, achieving the comparable accuracy already for 3 trees.

Computational performance. The 3D echocardiograms in our experiments had the size 218 × 202 × 272; around 45% of voxels were inside the echo-cone and, hence, classified with the random forest. For a modern CPU (2.4 GHz), the time required to delineate the entire echocardiogram was around 2.1 seconds for a forest with 3 trees and around 22 seconds for a forest with 20 trees. The GPU implementation as described in [7] is highly likely to run at several frames per second ([7] reported hundred-fold acceleration). The training procedure was also remarkably fast taking around 15 seconds per one tree on a CPU.

5 Conclusions and Future Work

We have evaluated random forests, which are one of the most accurate and computationally efficient discriminative classifiers, on the problem of automatic myocardial tissue delineation in 3D echocardiograms. A significant boost in accuracy compared to thresholding of raw intensities or Rayleigh parameters has been demonstrated on a dataset of real data. Importantly, the developed method is fast (potentially real-time), which distinguishes it from most (if not all) other methods published for this problem.

The future work would include a more thorough evaluation on a larger amount of data as well as assessing the clinical applicability of the algorithm as an aid for myocardium volume estimation and, potentially, regional wall thickening and motion abnormalities detection. We also plan to investigate the use of the output labelings within more sophisticated, higher-level algorithms, such as deformable template fitting, graph cuts, or level sets. However, a care should be taken that the acceptable speed is maintained.

References

1. Lynch, M., Ghita, O., Whelan, P.: Left-ventricle myocardium segmentation using a coupled level-set with a priori knowledge. Computerized Medical Imaging and Graphics 30(4), 255–262 (2006)
2. Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. Neural Computation 9(7), 1545–1588 (1997)
3. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
4. Winn, J.M., Shotton, J.: The layout consistent random field for recognizing and segmenting partially occluded objects. In: CVPR (1), pp. 37–44 (2006)
5. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR (2008)
6. Schroff, F., Criminisi, A., Zisserman, A.: Object class segmentation using random forests. In: BMVC (2008)
7. Sharp, T.: Implementing decision trees and forests on a GPU. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 595–608. Springer, Heidelberg (2008)
8. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: ICML, pp. 161–168 (2006)
9. Noble, J., Boukerroui, D.: Ultrasound image segmentation: a survey. IEEE Trans. Med. Imaging 25(8), 987–1010 (2006)

10. Yan, J., Zhuang, T.: Applying improved fast marching method to endocardial boundary detection in echocardiographic images. Pattern Recognition Letters 24(15), 2777–2784 (2003)
11. Angelini, E.D., Homma, S., Pearson, G., Holmes, J.W., Laine, A.F.: Segmentation of real-time three-dimensional ultrasound for quantification of ventricular function: A clinical study on right and left ventricles. Ultrasound in Medicine & Biology 31(9), 1143–1158 (2005)
12. Jacob, G., Noble, J., Behrenbruch, C., Kelion, A., Banning, A.: A shape-space-based approach to tracking myocardial borders and quantifying regional left-ventricular function applied in echocardiography. IEEE Trans. Med. Imaging 21(3), 226–238 (2002)
13. Zhu, Y., Papademetris, X., Sinusas, A., Duncan, J.S.: Segmentation of myocardial volumes from real-time 3D echocardiography using an incompressibility constraint. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part I. LNCS, vol. 4791, pp. 44–51. Springer, Heidelberg (2007)
14. Nillesen, M.M., Lopata, R.G., Gerrits, I.H., Kapusta, L., Thijssen, J.M., de Korte, C.L.: Modeling envelope statistics of blood and myocardium for segmentation of echocardiographic images. Ultrasound in Medicine & Biology 34(4), 674–680 (2008)
15. Tao, Z., Tagare, H., Beaty, J.: Evaluation of four probability distribution models for speckle in clinical cardiac ultrasound images. IEEE Trans. Med. Imaging 25(11), 1483–1491 (2006)
16. Chang, R., Wu, W., Moon, W.K., Chou, Y., Chen, D.: Support vector machines for diagnosis of breast tumors on US images. Academic Radiology 10(2), 189–197 (2003)
17. Drukker, K., Giger, M.L., Vyborny, C.J., Mendelson, E.B.: Computerized detection and classification of cancer on breast ultrasound1. Academic Radiology 11(5), 526–535 (2004)
18. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: ICML, pp. 148–156 (1996)
19. Pujol, O., Rosales, M., Radeva, P., Nofrerias-Fernández, E.: Intravascular ultrasound images vessel characterization using adaboost. In: Magnin, I.E., Montagnat, J., Clarysse, P., Nenonen, J., Katila, T. (eds.) FIMH 2003. LNCS, vol. 2674, pp. 242–251. Springer, Heidelberg (2003)
20. Georgescu, B., Zhou, X.S., Comaniciu, D., Gupta, A.: Database-guided segmentation of anatomical structures with complex appearance. In: CVPR (2) (2005)
21. Quinlan, J.R.: Induction of decision trees. Machine Learning 1(1), 81–106 (1986)
22. Andres, B., Köthe, U., Helmstaedter, M., Denk, W., Hamprecht, F.A.: Segmentation of SBFSEM volume data of neural tissue by hierarchical classification. In: Rigoll, G. (ed.) DAGM 2008. LNCS, vol. 5096, pp. 142–152. Springer, Heidelberg (2008)
23. Hope, T.A., Gregson, P.H., Linney, N.C., Schmidt, M.H., Abdolell, M.: Selecting and assessing quantitative early ultrasound texture measures for their association with cerebral palsy. IEEE Trans. Med. Imaging 27(2), 228–236 (2008)
24. Vibha, L., Harshavardhan, G.M., Pranaw, K., Shenoy, P.D., Venugopal, K.R., Patnaik, L.M.: Lesion detection using segmentation and classification of mammograms. In: AIAP 2007, pp. 311–316 (2007)
25. Viola, P.A., Jones, M.J.: Robust real-time face detection. In: ICCV, p. 747 (2001)
26. van Rijsbergen, C.: Information Retrieval. Butterworth (1979)