

# Intravascular Ultrasound Images Vessel Characterization using AdaBoost

Oriol Pujol<sup>1</sup>, Misael Rosales<sup>1</sup>, Petia Radeva<sup>1</sup>, and Eduard Nofrerias-Fernández<sup>2</sup>

<sup>1</sup> Computer Vision Center, Universitat Autònoma de Barcelona, Edifici O, Campus  
UAB, Bellaterra. Spain

`oriol@cvc.uab.es, misael@cvc.uab.es, petia@cvc.uab.es`

<sup>2</sup> Hospital Universitari Germans Trias i Pujol, Can Ruti. Barcelona. Spain

**Abstract.** This paper presents a method for accurate location of the vessel borders based on boosting of classifiers and feature selection. Intravascular Ultrasound Images (IVUS) are an excellent tool for direct visualization of vascular pathologies and evaluation of the lumen and plaque in coronary arteries. Nowadays, the most common methods to separate the tissue from the lumen are based on gray levels providing non-satisfactory segmentations. In this paper, we propose and analyze a new approach to separate tissue from lumen based on an ensemble method for classification and feature selection. We perform a supervised learning of local texture patterns of the plaque and lumen regions and build a large feature space using different texture extractors. A classifier is constructed by selecting a small number of important features using AdaBoost. Feature selection is achieved by a modification of the AdaBoost. A snake is set to deform to achieve continuity on the classified image. Different tests on medical images show the advantages.

## 1 Introduction

Intravascular Ultrasound Images (IVUS) are an excellent tool for direct visualization of vascular pathologies and evaluation of the lumen and plaque in coronary arteries. However, visual evaluation and characterization of plaque require integration of complex information and suffer from substantial variability depending on the observer. This fact explains the difficulties of manual segmentation prone to high subjectivity in final results.

Automatic segmentation will save time to physicians and provide objective vessel measurements. [6] Nowadays, the most common methods to separate the tissue from the lumen are based on gray levels providing non-satisfactory segmentations. This leads to use more complex measures to discriminate lumen and plaque. One of the most wide spread methods in medical imaging for such task is texture analysis. The problem of texture analysis has played a prominent role in computer vision to solve problems of object segmentation and retrieval in numerous applications [4], [5]. This approach, encodes the textural features of our image, and provide a feature space in which a classification based on such primitives is easier to perform.

In general, two approaches are used for texture analysis: supervised and unsupervised analysis. Our scheme will use supervised texture analysis. Texture analysis has an important problem in both approaches, the precise location of textured object boundaries. Previous works in segmentation of IVUS images have shown different ways to segment lumen and to classify tissues [7], [8], [9]. However, these approaches usually are semi-automatic, and very sensitive to image artifacts. The classification process is critical step in any image segmentation problem. Recently, arcing and boosting techniques have been applied successfully to different computer vision areas [1]. In this paper we analyze the relevance of boosting techniques, and in particular AdaBoost in Intravascular Ultrasound Image analysis for a dual task, creation of a strong classifier and feature selection. This process is integrated in an automatic framework for discrimination of lumen and plaque. The method is divided in 4 steps, corresponding to preprocessing step, feature extraction, feature selection and classification, and higher level organization of data using deformable models. An objective evaluation of the different approaches is made and validated by the physicians in patients with different pathologies and images with different topologies. The paper is organized as follows: section 2 introduces the AdaBoost procedure for feature selection and classification assembling of "weak" classifiers; section 3 describes the framework in which the AdaBoost is integrated, pre-processing considerations, the feature space selected and deformable models; section 4 shows the results of the methods and section 5 discuss the future lines.

## 2 AdaBoost for feature selection and classification

Considering the general problem of texture analysis without any *a priori* information about the scale, orientation, ... a high dimensional feature space is generated, involving texture descriptors of different scales, orders, orientation, etc. Nowadays, there are different kinds of feature spaces that try to encode the texture information: Gabor filters responses, derivatives of Gaussian filters, wavelets, co-occurrence matrices measures, fractal spaces, Markov random fields, ... Therefore, a lot of processing time is invested in the characterization of the texture and the decision of the feature space. This problem emphasizes the importance of relevant feature selection, allowing subsequent reduction of time processing in the classification step. On the other hand, the decision of a classifier is not an easy task. Recently, arcing procedures (Adaptative reweighing and combining) have been credited to lead to very high classification ratios using "weak" learning processes [15], [1], [2]. Boosting allows the use of "weak" classifiers with accuracy on the training set greater random classification. The goal is to create a high performant classifier ensemble of "weak" classifiers.

### 2.1 AdaBoost procedure

Adaptative Boosting (AdaBoost) is an arcing method which allows the designer to continue adding "weak" classifiers until some desired low training error has

been achieved. A weight is assigned to each of the feature points. These weights measure how accurate the feature point is being classified. If it is accurately classified, then its probability of being used in subsequent learners is reduced, or emphasized otherwise. This way, AdaBoost focuses on difficult features.

To address both problems: feature selection and classification process, a weak learning algorithm is designed to select the single features which best separate the different classes. In our problem, the different classes are tissue and blood. For each feature, the weak learner determines the optimal classification threshold function, so that the minimum number of feature points is misclassified. The weak classifier,  $h_j(x)$  consists of a feature  $f_j$ , and the parameters of the classifier  $\Theta_j$ . Those parameters are a threshold  $\theta_j$ , a parity  $p_j$  and  $W_j$ , a classifier trained by linear discriminant analysis. Although the threshold separates the two classes it is not enough to identify which class is in either side of the threshold. Therefore a parameter  $p_j$  (parity) is needed to indicate the direction of the inequality sign when classifying:

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

The algorithm is described as follows

- Determine a supervised set of feature points  $\{x_i, c_i\}$  where  $c_i = \{-1, 1\}$  is the class associated to each of the features classes (blood and tissue respectively)
- Initialize weights  $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$  for  $c_i = \{-1, 1\}$  respectively, where  $m$  and  $l$  are the number of feature points for each class.
- For  $t = 1..T$ :

- Normalize weights

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,i}}$$

so that  $w_t$  is a probability distribution.

- For each feature,  $j$  train a classifier,  $h_j$  which is restricted to using a single feature. The error is evaluated with respect to  $w_t$ ,  $\epsilon_j = \sum_i w_i |h_j(x_i) - c_i|$ .
- Choose the classifier,  $h_t$  with the lowest error  $\epsilon_t$ .
- Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{e_i}$$

where  $e_i = 1$  for each well-classified feature and  $e_i = 0$  otherwise.  $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$ . Calculate parameter  $\alpha_t = -\log(\beta_t)$ .

– The final "strong" classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the strong classifier is the ensemble of a series of simple classifiers ("weak"). Parameter  $\alpha_t$  is the weighting factor of each of the classifiers. The loop ends whether the classification error of a "weak" classifier is over 0.5, the estimated error for the whole "strong" classifier is lower than a given error rate or if we achieve the desired number of "weak". The final classification is the result of the weighted classifications of the "weak". The process is designed so that if  $h(x) > 0$ , then pixel  $x$  belongs to class "tissue".

## 2.2 Weak classifiers

The weak classifier has an important role in the procedure. Different approaches can be used, however it is relatively interesting to center our attention in low time-consuming classifiers.

The first approach we have tried is a classical learner, modelling the feature points as Gaussian distributions. This is an easy scheme which allow us to introduce the weights easily, simply calculating the weighed mean and covariance of the classes at each step of the process:

$$\mu_{i,t}^j = \sum w_{i,t} x_i \quad \Sigma_{i,t}^j = \sum w_{i,t} (x_i - \mu_{i,t}^j)^2$$

for each  $x_i^j$  point in class  $C_j$ .  $w_{i,j}$  are the weights for each data point.

This classifier, though it leads to fairly good results is highly constrained to the N features of the N-dimensional feature space. If N is not enough large, the procedure could not improve its performance. Therefore we propose another classifier for relatively low dimensional spaces (2 magnitude orders). Because the selection of a single feature for each of the classifiers is quite a hard constraint, we can look for the most significant pair of features which discriminates better the blood and the tissue.

For each pair of features of our space we use linear discriminant analysis to find the transformation which leads to the most discriminant axis. We chose the pair of features with the lowest error. We can describe this "weak" classifier as follows.

$$h(x) = \begin{cases} 1 & \text{if } p_j W_j^t x < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

where  $p_j$  and  $\theta_j$  are the parity and threshold parameters and  $W_j$  is defined as follows:

$$W_j = \Sigma_j^{-1} (\mu_{-1,j} - \mu_{1,j})$$

which is the canonical variate.  $W_j$  is the principal axis of the solution of the linear discriminant analysis system which maximizes

$$J(W) = \frac{W^t S_B W}{W^t S_W W}$$

, where  $S_B$  is the between-class scatter  $S_B = \sum_{i=1}^C N_i(\mu_i - \mu)(\mu_i - \mu)^t$  and  $S_W$  is the within-class scatter  $S_W = \sum_{i=1}^C \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^t$ , where  $\mu$  is the mean value of the whole data,  $c$  is the number of classes and  $N_i$  is the number of samples in class  $i$ .

### 3 Integration framework

#### 3.1 Cartesian to Polar conversion

Intravascular Ultrasound Images are usually displayed as a cartesian image although the transducer acquires the echoes while rotating. This gives an accurate description of the physiology of the artery, but it also has some disadvantages. Due to the fact that the transducer casts rays at a definite angular speed, not all the space can be covered. That means that the image shown is inaccurate because of the interpolation process performed to fill the distance between rays. The interpolation process creates false structures, which are seen as long circular areas with similar intensities, while the real structure has not such length.



**Fig. 1.** (a) Original image. (b) Polar transformed image.

This press us to work with the polar representation of the data, keeping just the rays with information. Figure 1 shows an example of the process. On the left, the polar image, which the physicians see, and on the right side the polar conversion of the image. Since we do not have access to the raw data from the transducer, or to some technical specifications of the same, we have chosen a 360 degrees representation of the image.

#### 3.2 Feature Extraction

The input of the feature extraction module is the transformed image. The response of the method is a feature vector. The texture feature spaces selected for our approach is twofold: the co-occurrence matrices space and the cumulative moments space. They both provide good representation of the texture space for our problem.

The Gray Level Co-occurrence Matrix is a well-known statistical tool for extracting second-order texture information from images [10]. In the co-occurrence method, the relative frequencies of gray level pairs of pixels at certain relative displacement are computed and sorted in a matrix, the co-occurrence matrix  $P$ . For  $G$  gray levels in the image,  $P$  will be of size  $G \times G$ . If  $G$  is large, the number of pixel pairs contributing to each element,  $p_{i,j}$  in  $P$  is low, and the statistical significance poor. On the other hand if the number of gray levels is low, much of the texture information may be lost in the image quantization. The element values in the matrix, when normalized are bounded by  $[0, 1]$ , and the sum of all element values is equal to 1.

$$P(i, j, D, \theta) = P(I(l, m) = i \text{ and } I(l + D\cos(\theta), m + D\sin(\theta)) = j)$$

where  $I(l, m)$  is the image at pixel  $(l, m)$ ,  $D$  is the distance between pixels and  $\theta$  is the angle. As suggested by other researchers [11] [12] the nearest neighbor pairs at orientations  $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$  will be used in the experiments. To describe this two-dimensional probability density function, several features are computed. We will consider six of the most used measures: Energy, Entropy, Inverse Differential Moment, Shade, Prominance and Inertia, for each of the orientations.

Geometric moments have been used effectively for texture segmentation in many occasions [13]. Actually, any set of parameters obtained by projecting an image onto a 2D polynomial basis are called moments. Then, since different sets of polynomials up to the same order define the same subspace, any complete set of moments up to given order can be obtained from any other set of moments up to the same order. A fast computation set of moments is the accumulation local moments. Two kind of accumulation local moments can be computed, direct accumulation and reverse accumulation. Since direct accumulation is more sensitive to round off errors and small perturbations in the input data [14], the reverse accumulation moments are used.

The reverse accumulation moment of order  $(k-1, l-1)$  of matrix  $\mathbf{I}_{ab}$  is the value of  $\mathbf{I}_{ab}[1, 1]$  after bottom-up accumulating its column  $k$  times (i.e., after applying  $k$  times the assignment  $\mathbf{I}_{ab}[a-i, j] \leftarrow \mathbf{I}_{ab}[a-i, j] + \mathbf{I}_{ab}[a-i+1, j]$ , for  $i = 0$  to  $a-1$ , and for  $j = 1$  to  $b$ ), and accumulating the resulting first row from right to left  $l$  times (i.e., after applying  $l$  times the assignment  $\mathbf{I}_{ab}[1, b-j] \leftarrow \mathbf{I}_{ab}[1, b-j] + \mathbf{I}_{ab}[1, b-j+1]$ , for  $j = 1$  to  $b-1$ ). The reverse accumulation moment matrix is defined so that  $\mathbf{R}_{mn}[k, l]$  is the reverse accumulation moment of order  $(k-1, l-1)$ .

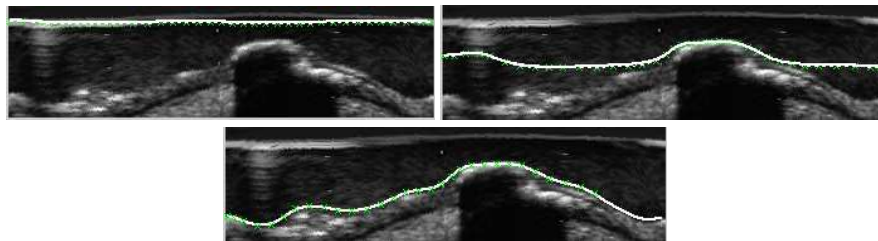
Once the image is represented in the feature space, the next step is the classification process and feature selection, explained in detail in the former section.

### 3.3 Snake-based accurate location of borders

The result after the classification step is a clear image in which the main structures are clearly visible, but there is no connection among them. This is a

good example in which a snake can be useful. The basic target of active contours [16] [17] [18] is to find a parameterized curve that minimizes the weighted sum of its internal energy and external energy. Given a traditional snake curve  $\mathbf{x}(s) = (x(s), y(s)), s \in [0, 1]$ , the snake can be formulated as the minimization of the equation  $S(\mathbf{x}) = \int_0^1 (\alpha |\mathbf{x}'(s)|^2 + \beta |\mathbf{x}''(s)| + E_e) ds$ ; where  $\alpha$  and  $\beta$  are weighting factors and  $E_e$  the external energy.

The typical potential function designed to lead a deformable contour toward step edges is  $P(x, y) = -\gamma |\nabla [G_\sigma(x, y) * I(x, y)]|^2$  where  $\nabla$  is the gradient operator,  $\gamma$  is a weighting parameter,  $G_\sigma(x, y)$  is the gaussian filter of standard deviation  $\sigma$ , and  $I(x, y)$  is the image data. As can be observed greater  $\sigma$  will increase the attraction range but the edges will blur. In our experiments the snake is initialized at the top row of the image and is attracted by the classification edges; therefore  $I(x, y)$  will be the resulting classification image.



**Fig. 2.** Evolution of a snake on the IVUS image

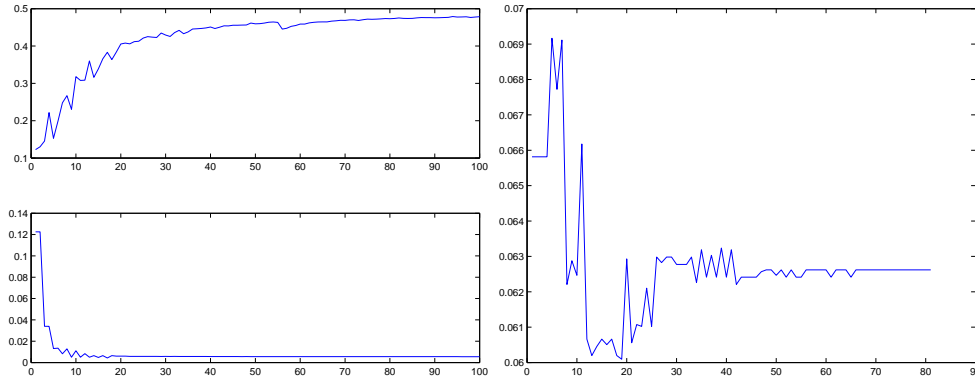
Figure 2 shows the deformation process of a snake. The snake is initialized on the top row of the image and deforms to the contours of the classification map. Parameters  $\alpha$  and  $\beta$  has been kept constant during all the experiments at a value of 0.3 each.

## 4 Experimental Results

We have used images from five different patients to build the training set. Segmentation of those images in the training step has been manually guided by experts. The test set is composed by different images of 5 different patients. The feature space is composed of the Co-occurrence matrices space with number of gray levels  $G = \{256, 64\}$ , neighborhoods of  $N = [5 \times 5], N = [8 \times 8]$ , and distances  $D = \{2, 3\}$ ; and cumulative moments  $N = 3$  and maximum order  $(k, l) = (9, 9)$ . As a result we have a  $24 \times 4$  space for co-occurrence matrices and a 81 dimensional space for cumulative moments. The analysis of the resulting features after the feature selection is quite significant since the 85% of the features selected by the AdaBoost are from the co-occurrence space. This encourages us to find work with the co-occurrence space, applying our second "weak" classifier

for feature pairs. The results with the "weak" classifier restrict the feature space to half of the feature space, reducing the processing time in the test step to a half.

Figure 3 shows the typical behavior of the training process of the AdaBoost clas-

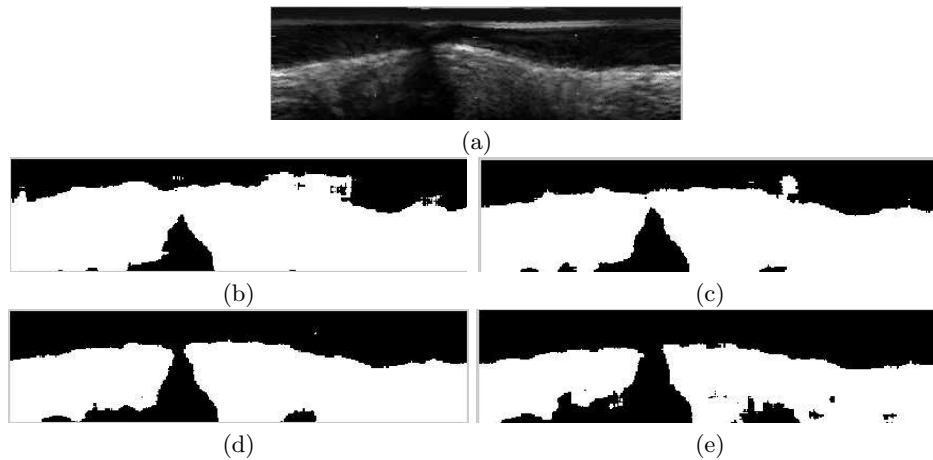


**Fig. 3.** Training error of the process for a mixed set of feature spaces. Weak classification error (top). Strong classification error (bottom). Test classification error (right)

sifier for a mixed set of feature spaces and a simple "weak" classifier approach. Figure 3 (top) shows the error rate of each of the "weak" classifiers. Each time a "weak" is assembled, feature are more difficult to classify due to the weights influence, therefore the "weak" classification error increases. However, 3 (bottom) shows how the joint error of the ensemble of the "weak" classifiers (the "strong" classifier) decreases as more "weaks" are assembled. Figure 3 (right) illustrates the test error. Note that after the 20th "weak" assemble the method overfits. Although theoretically the training error in the AdaBoost methods asymptotically tend to zero, in practical applications it has been shown that it does not necessarily converge to any desired error-rate [3]; this is most certain in high noise scenarios. IVUS images are one of these high noise image data. However the method can be applied successfully to create fast classifiers.

Figure 4 shows the evolution of the classification image. It can be seen that the classification tends to be better the most classifiers are assembled. However it must be said that most of the time it tends to overfit or to stabilize in a fixed error rate; hence, the number of the reliable "weak" classifiers for the application must be found. This can be done using cross-validation processes.

We have applied this fully automatic integration framework described in the former section to different sequences from 5 patients, for validation of the methodology. The mean error rate for the estimated contours in our experiments is  $0.18 \pm 0.04$  mm. and maximum error rate of  $0.43 \pm 0.06$  mm. This error rates are comparable to the detection schemes found in literature. However, the clas-



**Fig. 4.** Test classification images at different stages of the strong classifier. (a) Original image (b) "Strong" with 3 "weaks" (c) "Strong" with 7 "weaks" (d) "Strong" with 17 "weaks" (e) "Strong" with 22 "weaks"

sification step is performed in very little time, allowing this kind of schemes apt for real-time detection processes.

## 5 Conclusion and Future lines

We have presented an integration framework for tissue-blood segmentation in IVUS images using an classifier ensemble with Adaboost. The Adaboost process is also used for feature selection so the feature extraction process is accelerated in the testing step. The classifier found is apt for real time classification. The method has convergence problems with high noise images, this emphasizes the necessity of a regularization scheme for better performance of the method. The future work lead us to investigate in fast tissue characterization for IVUS images, a very important problem due to the fact that any patient study needs segmenting of about thousand of images that takes about an hour to the medical experts. This method is robust to blood area size. Therefore occlusive plaques can be segmented correctly. The method can be used in future with suitable descriptors of the plaque to segment the plaque from the normal tissue.

## Acknowledgements

This work was partially supported by the project TIC2000-1635-C04-04 of CI-CYT, Ministerio de Ciencia y Tecnologia of Spain.

## References

1. R. E. Schapire. "The Boosting Approach to Machine Learning. An Overview." MSRI Workshop on Nonlinear Estimation and Classification. 2002.
2. P. Viola and M. Jones. "Rapid Object Detection using a Boosted Cascade of Simple Features". Accepted Conference on Computer Vision and Pattern Recognition 2001.
3. G. Ratsch, T. Onoda, K-R. Muller. "Soft Margins for AdaBoost". NeuroColt2 Technical Report Series. NC-TR-1998-021.
4. J. Malik, S. Belongie, T. Leung and J. Shi. "Contour and Texture Analysis for Image Segmentation". International Journal of Computer Vision. 43(1),7-27,2001.
5. J. Puzicha, T. Hoffman and J. Buhmann. "Unsupervised texture segmentation in a deterministic annealing framework". Trans. on Pattern Recognition and Machine Intelligence. 20(8):803-818,1998.
6. M. Sonka and X. Zhang and M. Siebes et al. "Segmentation of intravascular ultrasound images: A knowledge based approach". IEEE Trans. on Medical Imaging. 14: 719-732. 1995.
7. X. Zhang and C.R. McKay and M. Sonka. "Tissue Characterization in intravascular ultrasound images". IEEE Trans. on Medical Imaging. 17: 889-899. 1998.
8. C. von Birgelen and A. van der Lugt and A. Nicosia et al. "Computerized assessment of coronary lumen and atherosclerotic plaque dimensions in three-dimensional intravascular ultrasound correlated with histomorphometry". Amer. J. Cardiol. 78: 1202-1209, 1996.
9. J.D. Klingensmith and R. Shekhar and D.G. Vince, " Evaluation of Three-Dimensional Segmentation Algorithms for Identification of Luminal and Medial-Adventitial Borders in Intravascular Ultrasound Images", IEEE Trans. on Medical Imaging, 19(10): 996-1011,2000.
10. R. Haralick and K. Shanmugam and I. Dinstein. "Textural Features for Image Classification". IEEE Trans. System, Man, Cybernetics. 3:610-621. 1973.
11. P.P. Ohanian and R.C. Dubes. "Performance Evaluation for Four Classes of Textural Features". Pattern Recognition. 25(8),819-833, 1992.
12. Trygve Randen and John H. Husoy. "Filtering for Texture Classification: A Comparative Study". Pattern Recognition. 21(4):291-310. 1999.
13. M. Turceyan. "Moment Based texture segmentation". Pattern Recognition Letters. 15:659-668. 1994.
14. J. Martinez and F. Thomas. "Efficient computation of local geometrical moments". Submitted to IEEE Trans. on Image Processing.
15. Richard O. Duda and Peter E. Hart and David G. Stork. "Pattern Classification". Wiley-Interscience, 2001. 2nd Ed.
16. M. Kass and A. Witkin and D. Terzopoulos. "Snakes, Active contour models". Int. J. Computer Vision, 1(4): 321-331. 1987.
17. V. Caselles, F. Catte, T. Coll and F. Dibos. "A geometric model for active contours". Numerische Mathematik. 66:1-31, 1993.
18. T. McInerney and D. Terzopoulos. "Deformable models in medical images analysis: a survey". Medical Image Analysis. 1(2):91-108, 1996.