AUTOMATING SNAKES FOR MULTIPLE OBJECTS DETECTION

Anonymous ACCV submission

Paper ID 329

Abstract. Active contour or snake has emerged as an indispensable interactive image segmentation tool in various applications. However, snake fails to serve many significant image segmentation applications that require complete automation. Here, we present a novel technique to automate snake/active contour for multiple objects detection. We first apply a probabilistic quad tree based approximate segmentation technique to find the regions of interest (ROI) in an image, evolve modifed GVF snakes within ROIs and finally classify the snakes into object and non-object classes using boosting. We propose a novel loss function for boosting that is more robust to outliers concerning snake classification and we derive a modified Adaboost algorithm by minimizing the proposed loss function to achieve better classification results. Extensive experiments have been carried out on two datasets: one has importance in oil sand mining industry and the other one is significant in bio-medical engineering. Performances of proposed snake validation have been compared with competitive methods. Results show that proposed algorithm is computationally less expensive and can delineate objects up to 30% more accurately as well as precisely.

1 Introduction

Snake/active contour [2] has made its recognition as an interactive image segmentation tool for the last two decades. However, it is yet to be seen as a completely automated segmentation tool. Snake algorithms consist of three sequential steps: snake initialization, snake evolution and snake validation [3]. For multiple objects detection, seeds are chosen inside the objects at the initialization step, then snakes are evolved from those seed points and finally the evolved snakes are passed through a validation procedure to examine whether the snakes delineate the desired objects [3]. Substantial endeavors have taken place on the initialization and evolution steps towards snake automation. Most of the existing initialization algorithms [4] exploit the local maxima or other characteristics of the external energy that help to generate seed points within the objects. However, clutters in the noisy and poorly illuminated images generate considerable amount of seed points and snakes evolved from those seeds do not converge to the object boundaries. This necessitates a good validation scheme after snake evolution. Unfortunately, the validation step has not received much attention till date.

Saha *et al*. [3] proposed a snake validation scheme using principal component analysis (PCA). Their method places seeds blindly on the entire image and evolve one snake from each seed. When all snakes converge, a pattern image (an annular band) is formed across each snake contour. Each pattern image is then projected into an already trained PC (principal component) space and PCA reconstruction error is computed. The snakes associated with lower reconstruction errors than a threshold are considered as objects. Pattern images bear information regarding bright-to-dark (or vice-versa) transition across the object contours and show good discrimination capability between object and non-object classes. This validation technique is effective when the gradient strength of object boundaries is considerably high. Besides, throwing a large number of seeds blindly over an entire image might not be feasible for some applications, since the snake evolution can be computationally expensive. Thus, a handful of crucial seed points are always helpful.

In this paper, we propose a probabilistic quad tree (QT) based snake initialization scheme which is computationally inexpensive. QT automatically seeks ROIs from an image where the probabilities of locating objects are very high. We throw seeds only within ROIs and evolve one modified Gradient Vector Flow (GVF) snake [5] from each seed. Then we validate each evolved snake to verify whether they belong to object or non-object class. During validation, each snake is passed through a strong classifier formed by Adaboost [6]. We classify snake contours into objects and non-objects based on a set of features and we apply Adaboost for selecting important features. The parameters of the adaboost algorithm are estimated by minimizing an exponential loss function. Here, it is noted that one shortcoming of the exponential loss function associated with Adaboost algorithm is that the penalty increases exponentially negative margins which incurs high misclassification error rates due to outliers [6]. We propose a novel loss function that incurs smaller penalties in the negative margin, and thus make Adaboost more robust to outliers. Also, we can choose the amount of penalty judiciously from the training set using cross validation. We exploit the advantages of multiple features including region, edge and shape over PCA-based intensity feature proposed earlier [3]. Note that our proposed initialization and validation algorithm could be successfully used as plugins with any existing snake evolution techniques. We have carried out experiments on two real datasets: (a) oil sand mining images [5]: analyzing these images help to improve the performance of oil sand extraction process and (b) leukocyte images [7]: processing these images help in the study of inflammation as well as in the design of anti/pro inflammatory drugs. Results illustrate that our proposed algorithm is faster, more reliable and robust than competitive methods.

The organization of this paper is as follows. Section 2 discusses proposed quad tree based snake initialization technique. Section 3 elaborates snake validation using boosting and illustrates proposed regularization into boosting framework. Section 4 demonstrates the performances of proposed techniques and displays comparative analysis of proposed techiques with competitive methods. Section 5 concludes our proposed work. Appendix includes derivation of proposed discrete Adaboost algorithm.

2 Quad tree based snake initialization

Quad tree [8] based segmentation algorithm receives an image as an input, and then divides it into four adjacent, non-overlapping quadrants if it meets pre-specified criteria, subsequently each quadrant is divided similarly and the process proceeds iteratively until it fails the pre-defined criteria. Consequently, the algorithm locates objects by smaller rectangular boxes. In our application here, the QT algorithm computes a posterior probability and splits the current region into four quadrants if the value of the posterior probability is between two predetermined thresholds. If the value of the posterior probability is greater than the upper threshold then the region is likely to contain objects ; if it is less than the lower threshold then it is likely to contain background. We locate objects by finding homogeneous regions based on local brightness and texture properties. We compute the posterior probability of a region (O) being object/non-object: $P(O/T, B) \propto P(T/O)P(B/O)P(O)$, where $P(O)$ is the prior probability. $P(T/O)$ and $P(B/O)$ are the likelihood of the region regarding texture and brightness respectively. Proposed probabilistic QT algorithm converges faster and delineates objects more accurately than deterministic quad tree algorithm if a suitable, application specific prior can be chosen. We compute texture energy (T) by the response of Gabor filters [8] and brightness (B) by the maximum singular value decomposition (SVD) [9] of the region. Maximum SVD encodes average brightness and Gabor filter response represents discriminative texture information for the objects. The details of computing posterior probability and two thresholds are mentioned in section 4.

3 Snake validation using boosting

We compute different features for each converged snake contour, such as, contour shape features (form factor, convexity, extent, modification ratio [10] etc.), regional features (intra and inter class variance, entropy etc.), and edge based features (GICOV [7], gradient strength etc.) for snake validation. We use Adaboost (variant of boosting) for selecting important features. At the training phase, boosting picks only important features for snake validation from a set of features computed on training snake contours and finds the weights associated with those features. We place seeds blindly over the training images and evolve one snake from each seed and classify the snakes as objects manually that converge at object contours found on the ground truth made by the experts; otherwise consider the snakes as non-objects and thus form a training set consisting of both positive (object) and negative (background) samples. The Adaboost algorithm forms a strong classifier by combining a set of weak learners linearly in an iterative manner [6]. We use decision stump (threshold) [6] as weak classifier. Decision stump is a single level decision tree. Decision stump, $G_j(x)$ for feature

f_j is defined as, $G_j(x) = 1$ if $x_j > \theta_j$, otherwise, $G_j(x) = 0$, where θ_j is some feature value of x_j chosen as threshold and $x = [x_1, x_2, x_3, ..., x_j, ...x_n]$ is the feature set. Finding the best decision stump at each stage is similar to learning a node in a decision tree. We search over all possible features $x = [x_1, x_2, x_3, ..., x_n]$ and for each feature, we search over all possible thresholds θ induced by sorting the observed values of x and pick x_k with θ_k that gives lowest misclassification error among all given features during training. At test phase, proposed QT algorithm discussed in section 2 locates ROIs (rectangular regions/patches) over the test images where the probability of localizing objects is greater than a predetermined upper threshold. We place seeds only within ROIs and grow one snake from each seed. When all snakes are fully converged, we compute the values of the important features for each snake and multiply them with the weights associated with the features chosen by boosting during training phase and subsequently add them to form a strong classifier, $G(x) = sign(\sum_{m=1}^{M} \alpha_m G_m(x))$, where, α_m is the weight associated with weak classifier $G_m(x)$. If the sign of the response of the strong classifier for a snake contour is positive then it is classified into object class, otherwise it is classified into non-object class. For classification, Adaboost minimizes an exponential loss function: where y is the response and f is the prediction. The drawback of this exponential loss function is that it incurs substantial misclassification error rate as the penalty increases exponentially for large increasing negative margin due to outliers [6]. To address this problem, we propose a novel loss function: $L(y, f(x)) = exp(-yf(x) + \lambda|y - G(x)|)$, where $\lambda < 0$ and $G(x)$ is the prediction of the weak classifier chosen at the current stage. We have mainly incorporated one extra term in the existing exponential loss function that acts as a regularizer. At any boosting iteration, the proposed loss function is the same as the existing loss function if the misclassification error rate at current stage is zero (proposed term vanishes when $\lambda = 0$). The only difference between the proposed and the exponential loss function is that the penalty associated with the proposed loss function is less than that of the exponential one, if the misclassification error rate at current stage is not equal to zero (shown in Fig.1(a) where loss is plotted against a function of the classification margin $y.f$). This modification leads to a low misclassification error rate and it becomes more robust to outliers. One additional advantage of this proposed loss function is that the user can adjust the amount of penalty for negative margins after observing the classifier performance over a training data set. Accordingly, we determine the value of λ through cross validation (λ is a function of k shown in the appendix and the value of k is determined experimentally). We derive a modified Adaboost algorithm by minimizing the proposed loss function (The derivation is shown in Appendix).

Our modified Adaboost finds the feature weight, $\alpha_m = log(k(1-err_m)/err_m)$, $k \geq 1$, where, for the existing Adaboost algorithm the value of k is always 1. This leads to the weights associated with misclassified observations at any stage is k times as much as the existing Adaboost (derivation is shown in the Appendix). The value of k for our modified Adaboost is determined by cross-validation and is discussed in the next section.

Our proposed term in the existing loss function acts as a regularizer in the boosting framework. There are two well known regularized boosting algorithms, ϵ-boosting [6] and l_1- regularized boosting [11] available in the literature. Unlike other two methods, our method can adaptively adjust the effects of regularization in the boosting framework by selecting the propoer value of k from the training data set. The regularization strategy in ϵ - boosting is imposed through shrinking the contribution of each feature (feature weight).In l_1- regularized boosting, the exponential loss function is minimized with l_1- regularization. This provides sparse solution and acts as a regularizer.

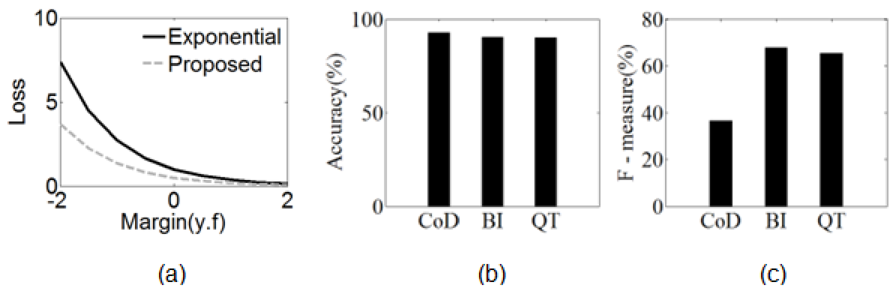

Fig. 1. (a) Loss functions for two class classification. (b) Accuracy and (c) F-measure for three different snake initialization methods.

4 Results and Discussions

We have carried out experiments on two real data sets: oil sand images and leukocyte microscopy images.

4.1 Oil Sand Images

In the oil sand extraction process, oil sand ore is crushed, broken into smaller particles through crusher and then passed through screens to reject oversize ores and the undersize ores are transported to hydrotransport plant for further processing. Here, ore size is an important measure to estimate crusher as well as screen efficiency. Towards achieving this goal, oil sand images are captured through camera mounted over conveyor belt before and after the crusher as well as screen. Oil sand particles are detected in the images using the proposed method and then the particle size distribution (PSD) is computed. PSD is a histogram showing frequency of the particles over their sizes. In this paper, we have concentrated on the automatic detection of the oil sand particles. We construct a training set using 20 images and test set using 100 images sampled randomly from an online video of oil sand particles over conveyor belt. For QT based snake initialization, we find the distribution for prior and likelihood as well as the two threshold values ((P_{th1}) and (P_{th2})) of the posterior probability ($P(O/T,B)$) experimentally from the training set. We have $P(O/T,B) \propto P(T/O)P(B/O)P(O)$, where T and B represent texture and brightness respectively. Maximum Singular Value Decomposition (SVD) encodes average brightness of a region where average of the response of the gabor filter on a region encodes texture of the region. Experimentally it is found that maximum SVD of the oil sand patch follows doubly truncated exponential (DTE) distribution. Probability density function (pdf) of DTE [12] is given by, $P(B) = \frac{exp(-(B-\mu)/\sigma)}{\sigma[1-exp(-(x_0-\mu)/\sigma)]} I_{[\mu,x_0]}(B), \mu \leq B \leq x_0$. On the otherhand, the response of the gabor filter follows doubly truncated normal distribution (DTN); pdf of DTN is given by, $P(T) = \frac{\frac{1}{\sigma\sqrt{2\pi}} exp(-(T-\mu)^2/2\sigma^2)}{\Phi(\frac{b-\mu}{\sigma})-\Phi(\frac{a-\mu}{\sigma})} I_{[a,b]}(T), a \leq T \leq b$, where Φ is the standard normal cumulative density function(cdf) [12]. The value of indicator function, $I_{[a,b]} = 1$ if $a \leq T \leq b$, and is 0 otherwise. $I_{[\mu,x_0]}(B)$ is defined similarly. A region will have high oil sand particles density if $P(O/T,B) \geq P_{th2}$. The two threshold values of the posterior probability (P_{th1} andP_{th2}) are determined experimentally from the training set. The parameters of the above distributions are estimated using maximum likelihood estimation (MLE). Fig. 2(a) and fig. 2(b) shows the distribution of the brightness and texture of the oil sand particles respectively.

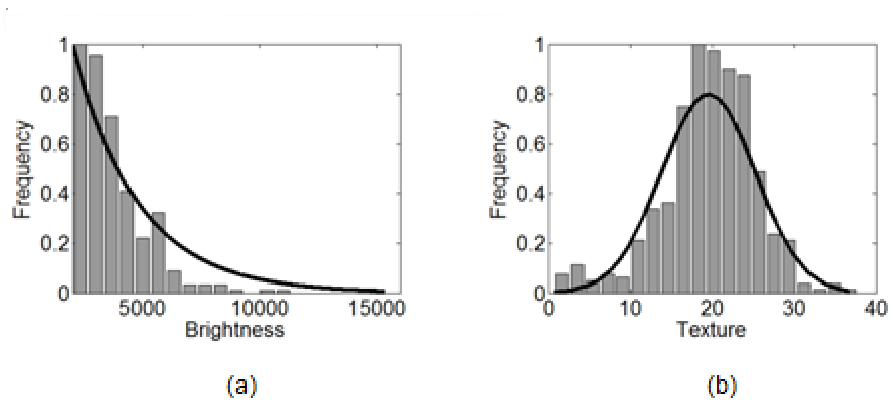

Fig. 2. Histogram of brightness and texture of oil sand particles.

Regions of Interest (ROI) generated by QT and seeds generated by Center of Divergence (CoD) [4] method are shown in Fig. 4. Table 1 illustrates the number of seeds generated by the proposed QT, CoD and blind initialization (BI) [3]. CoD refers to the local maxima of the external Gradient Vector Flow (GVF) field. The point from which the GVF vectors to all of its neighboring pixels radiate is considered as CoD. CoD is supposed to be located within the object and the snake evolved from CoD converges to the actual boundary of the

Table 1. Comparison among three snake initialization techniques.

Datasets	# of objects	# of seeds generated by		
		CoD	BI	**QT**
Oil Sand	349	3786	3000	**686**
Leukocyte	193	2402	4375	**799**

object in noise-free settings. Fig. 1(b) and 1(c) show accuracy and F-measure for CoD, BI and QT techniques with proposed modified Adaboost based validation technique respectively. F-measure combines both recall and precision into a single entity [13]. Results show that though all techniques possess the same accuracy, both BI and QT achieve 30% more F-measure value than that of CoD but QT generates significantly fewer seeds (Table 1) than other competitve methods.

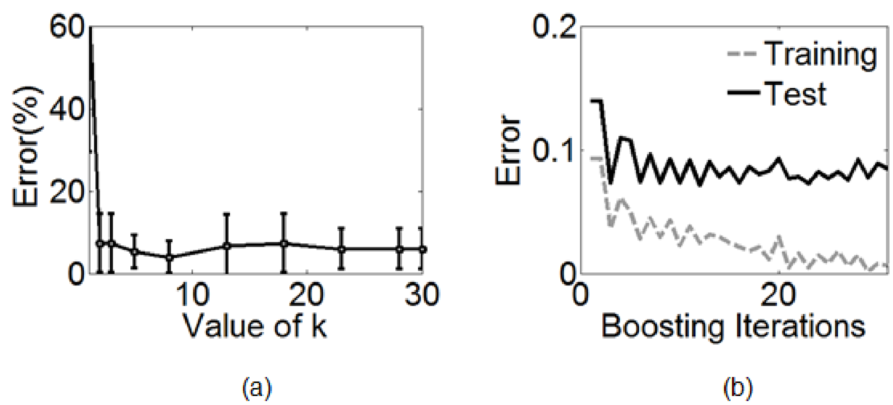

Fig. 3. (a) fivefold cross validation curve with standard error bars; the curve has minima at $k = 8$. (b) Misclassification error rate over the number of iterations for oil sand images.

Next, we determine the value of k (discussed regarding feature weight in section 3) using five-fold cross validation [6] technique. We compute misclassification errors for different values of k and is shown in Fig. 3(a). Standard error bars indicate the standard errors of the individual misclassification error rates for each of the five parts. It is observed that both the average misclassification error rate and standard error is minimum for $k = 8$ for oil sand images. For existing Adaboost algorithm, the value of k is always 1. Modified Adaboost always outperforms the existing Adaboost algorithm because the modified one can select the best value of k for which the misclassification error is minimum. The misclassification error rate for boosting with decision stumps [6], as a function of the number of iterations for $k = 8$ is shown in Fig. 3(b).

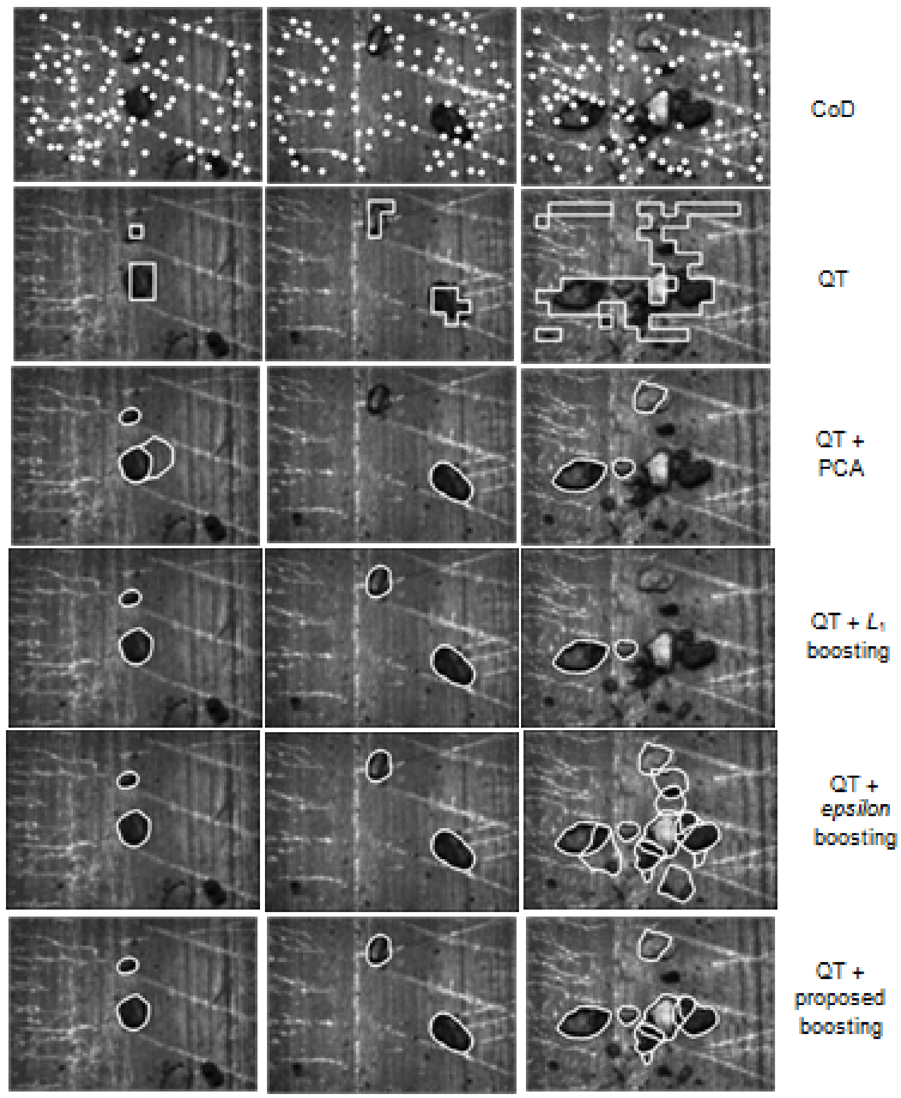

Fig. 4. Results of different methods on oil sand images.

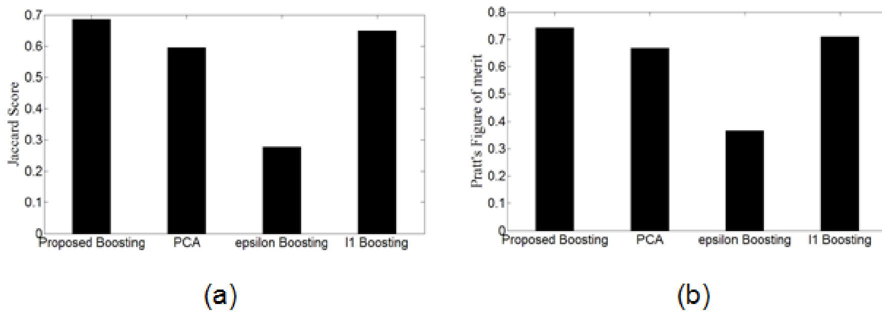

Fig. 5. Segmentation scores: (a) Jaccard Score and (b) Pratt's Figure of Merit of different methods on oil sand images.

Fig. 4 shows the results of proposed Adaboost, ϵ-boosting [6], l_1 regularized boosting [11] and PCA [3] on oil sand images and their comparisons are shown in Fig. 5 and Fig. 8(a). Fig. 5(a) shows the average Jaccard Score [1] and Fig. 5(b) shows the average Pratts figure of merit (PFOM) [14] for these methods. Jaccard Score measures the fraction of overlap area among detected and true objects. Pratt's figure of merit determines the closeness among detected and actual edge pixels. Domain expert visually determines actual edge pixels and true object area from an image. Both Jaccard Score and Pratt's figure of merit are important to judge the segmentation quality of an algorithm and both are bounded by 0 and 1. Superior performance of a segmentation algorithm is indicated by higher PFOM as well as Jaccard Score values.

4.2 Leukocyte Images

Leukocyte plays an important role in the study of inflammation. Inflammation is a natural defense mechanism initiated by tissue damage. During inflammatory responses, endothelium cell is activated, then leukocyte starts deviating from mainstream blood flow and contact the activated endothelium cell. This slowdown movement of leukocyte in contact with endothelium cell is known as rolling. Finally, from the rolling stage, leukocyte diffuses through the vascular wall, reach the injured tissues, and encounter the germs. Although inflammation is a normal defense mechanism, it sometimes becomes dangerous in the context of various inflammatory diseases. To combat such diseases, anti-inflammatory drugs are developed by blocking or controlling any of the necessary processes of inflammatory response. Here, the rolling velocity of leukocyte is an important factor in the study of inflammation. To measure and analyze the rolling velocity of leukocyte from the *in vivo* experiments, video recordings of the postcapilary vennule of a cremaster muscle are made through a CCD camera coupled with the intravital microscope. Then leukocyte cells are detected from the video frames using the proposed method and a correspondence analysis is carried out between consecutive images and finally cell motility is measured [7]. In this paper, we have concentrated only leukocyte detection. We have carried out experiment on a training set of 5 and a test set of 25 leukocyte images. Detections obtained by proposed Adaboost, ϵ-boosting [6], l_1 regularized boosting [11] and PCA [3] techniques are shown in Fig. 6 and their comparisons are shown in Fig. 7 and Fig. 8(b).

4.3 Interpretation of Results

One can interpret that proposed adaboost based validation is better than ϵ-boosting [6], l_1 regularized boosting [11] and PCA [3] based technique since it can detect more oil sand particles and leukocytes accurately and precisely. Segmentation score (Jaccard Score and Pratt's Figure of Merit) as well as area under ROC curve of proposed adaboost is greater than that of other methods.

5 Conclusion and Fututre works

Towards complete automation of snake algorithm, we have proposed an initialization as well as validation algorithm that could be utilized as a successful plug-in for existing snake/active contour tools. Existing research mainly focuses on the snake initialization and evolution steps and ignores the validation step. Here, we emphasize that we cannot omit the validation step in spite of applying the smart initialization technique of snake algorithm used for multiple objects detection. We have proposed probabilistic quad tree based approximate segmentation for snake initialization. We show that our proposed initialization outperforms existing initialization methods. We have successfully incorporated regularization into boosting framework and we demonstrate that our intended loss function is more robust to outliers concerning snake classification into object and non-object classes. We also show that proposed boosting based snanke validation technique outperforms existing PCA based validation method. Results of extensive experiments illustrate that proposed method is fast, reliable and more accurate than existing methods.

We would like to incorporate our initialization and validation methods with other well known snake evolution methods. Also we will further explore the characteristics of proposed regularization into boosting frameworks extensively by conducting experiments with available benchmark datasets.

6 Appendix

Derivation of Proposed Discrete Adaboost Algorithm

Proposed loss function is: $L(y, f(x)) = exp(-yf(x) + \lambda|y - G(x)|)$, where $\lambda < 0$. Let $f_m(x) = f_{m-1}(x) + \beta_m G_m(x)$ be the strong classifier composed of first m classifiers. We can pose m-th iteration of adaboost as the following optimization, $(\beta_m, G_m) = \underset{\beta,G}{argmin} \sum_{i=1}^{N} exp[-y_i(f_{m-1}(x_i) + \beta G(x_i)) + \lambda|y_i - G(x_i)|]$

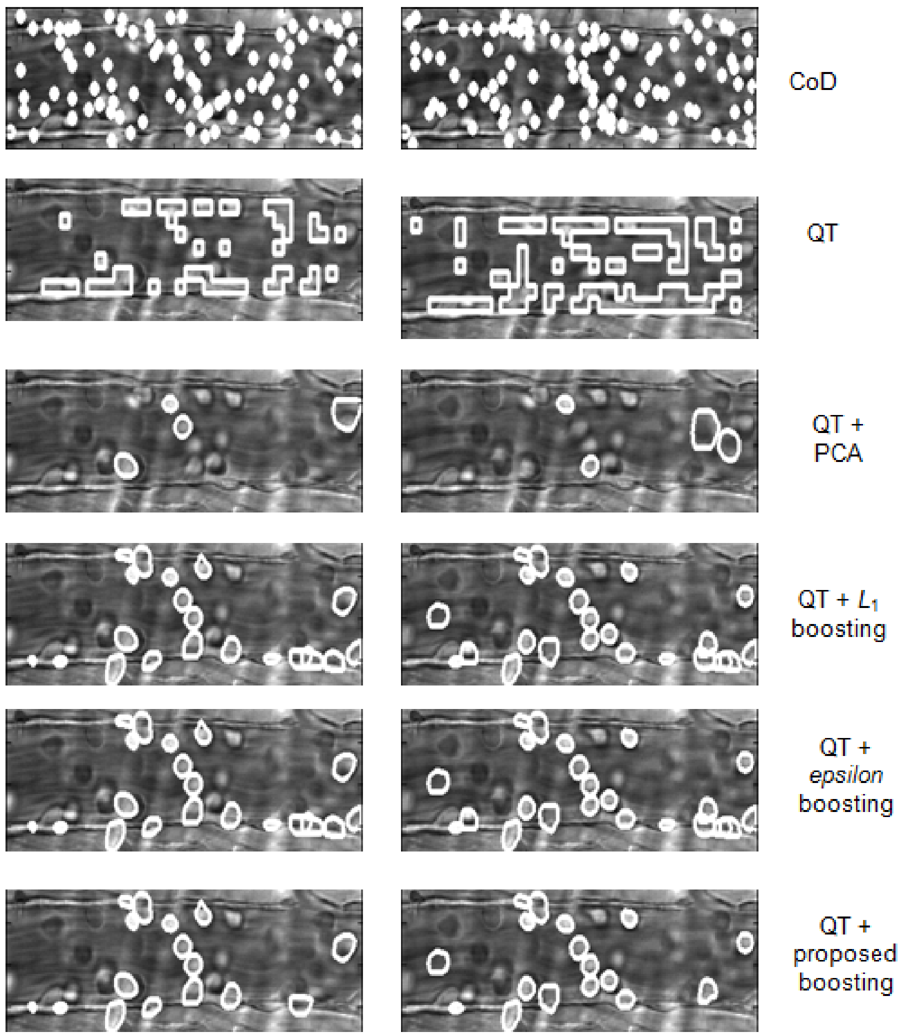

Fig. 6. Results of different techniques on on leukocyte images.

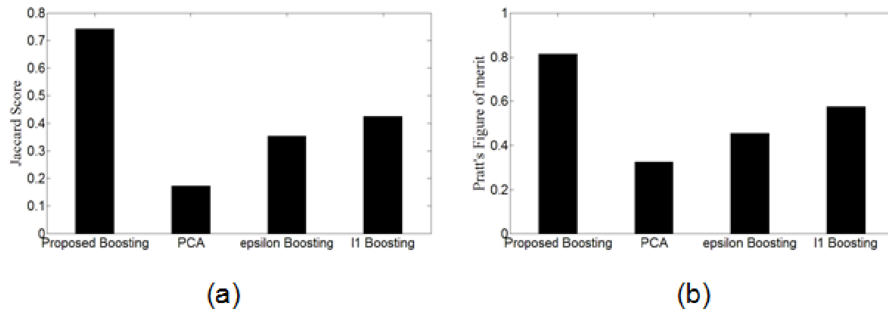

Fig. 7. Segmentation scores: (a) Jaccard Score and (b) Pratt's Figure of Merit of different methods on leukocyte images.

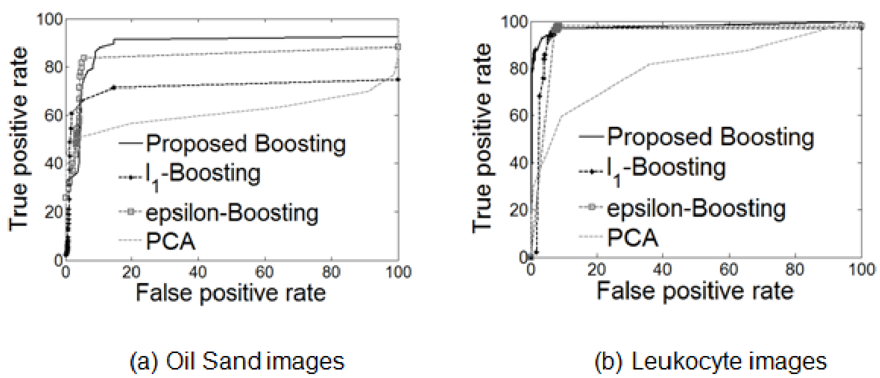

Fig. 8. Receiver Operating Characteristic (ROC) curves.

$\Rightarrow (\beta_m, G_m) = \underset{\beta,G}{argmin} \sum_{i=1}^{N} w_i^m exp[-y_i \beta G(x_i)) + \lambda|y_i - G(x_i)|]$ where, $w_i^m = exp(-y_i f_{m-1}(x_i))$ is free of both β and $G(x)$.
$\Rightarrow (\beta_m, G_m) = \underset{\beta,G}{argmin}[exp(-\beta) \sum_{i:y_i=G(x_i)} w_i^m + exp(\beta + 2\lambda) \sum_{i:y_i \neq G(x_i)} w_i^m]$.
$= \underset{\beta,G}{argmin}[exp(\beta+2\lambda) - exp(-\beta)) \sum_{i:y_i \neq G(x_i)} w_i^m + exp(-\beta) \sum_{i=1}^{N} w_i^m]$. The solution for$\beta_m$ and G_m can be obtained in two steps. First, for any value of $\beta > 0$, the solution for G_m is: $G_m = \underset{G}{argmin} \sum_{i=1}^{N} w_i^m I(y_i \neq G(x_i))$. Let $err_m = \underset{G}{argmin} \sum_{i=1}^{N} w_i^m I(y_i \neq G(x_i)) / \sum_{i=1}^{N} w_i^m$, then $\beta_m = \frac{\partial}{\partial \beta}(\sum_{i=1}^{N} w_i^m((exp(\beta + 2\lambda) - exp(-\beta))err_m + exp(-\beta))) = 0 \Rightarrow \beta_m = \frac{1}{2}(log\frac{1-err_m}{err_m}) - \lambda = \frac{1}{2}(logk\frac{1-err_m}{err_m})$, where, $\lambda = -\frac{1}{2}log(k), k > 0$. Now, $w_i^{m+1} = w_i^m exp(-\beta_m y_i G_m(x_i))$. Using the fact that $-y_i G_m(x_i) = 2I(y_i \neq G(x_i)) - 1$, we get, $w_i^{m+1} = w_i^m exp(\alpha_m I(y_i \neq G(x_i)))exp(-\beta_m)$ where, $\alpha_m = 2\beta_m = log(k((1 - err_m)/err_m))$. So, $w_i^{m+1} = w_i^m exp(\alpha_m I(y_i \neq G(x_i)))$. The factor $exp(-\beta_m)$ multiplies all weights by the same value, so it has no effect.

References

1. Jaccard, P.: Distribution de la flore alpine dans le bassin des dranses et dans quelques rgions voisines. Bulletin de la Socit Vaudoise des Sciences Naturelles **37** (1901) 241–272
2. M. Kass, A.W., Terzopoulos: Snakes: active contour models. IJCV **1** (1987) 321–331
3. B. N. Saha, N.R., Zhang, H.: Snake validation: A pca-based outlier detection method. IEEE Signal Processing Letters **16** (2009) 549–552
4. Ge, X., Tian, J.: An automatic active contour model for multiple objects. ICPR **2** (2002) 881–884
5. B. N. Saha, N.R., Zhang, H.: Computing oil sand particle size distribution by snake-pca algorithm. ICASSP (2008) 977–980
6. T. Hastie, R.T., Friedman, J.: The elements of statistical learning: Data mining, inference, and prediction (2009) Springer, Second Edition.
7. G. Dong, N.R., Acton, S.T.: Intravital leukocyte detection using the gradient inverse coefficient of variation. IEEE Transaction on Medical Imaging **24** (2005) 910–924
8. M. Mirmehdi, X.X., Suri, J.: Handbook of texture analysis (2008) Imperial college Press.
9. D. Omerevi, R. Perko, A.T.T.J.O.E., Leonardis, A.: Vegetation segmentation for boosting performance of mser feature detector. Computer Vision Winter Workshop (2008) 17–23
10. Russ, J.C.: The image processing handbook (1995) Third Edition, CRC & IEEE press.
11. Y. T. Xi, Z. J. Xiang, P.J.R., Schapire, R.E.: Speed and sparsity of regularized boosting. 12th International Conference on Artificial Intelligence and Statistics (AISTATS) **5** (2009) 615–622
12. Maritz, J.S.: Distribution-free statistical methods (1995) Chapman & Hall, Second Edition.

13. van Rijsbergen, C.J.: Information retireval (1979) Butterworths, London.
14. Abdou, I.E., Pratt, W.K.: Quantitative design and evaluation of enhancement/thresholding edge detectors. proceedings of the IEEE **67** (1979) 753–763