A Learning Based Approach for 3D Segmentation and Colon Detagging

Zhuowen Tu[1], Xiang (Sean) Zhou[2], Dorin Comaniciu[1], and Luca Bogoni[2]

[1] Integrated Data Systems Department, Siemens Corporate Research,
750 College Road East, Princeton, NJ, USA
[2] CAD Solutions, Siemens Medical Solutions
51 Calley Stream Parkway, Malvern, PA, USA

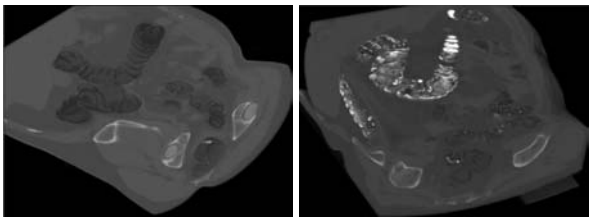
Abstract. Foreground and background segmentation is a typical problem in computer vision and medical imaging. In this paper, we propose a new learning based approach for 3D segmentation, and we show its application on colon detagging. In many problems in vision, both the foreground and the background observe large intra-class variation and inter-class similarity. This makes the task of modeling and segregation of the foreground and the background very hard. The framework presented in this paper has the following key components: (1) We adopt *probabilistic boosting tree* [9] for learning discriminative models for the appearance of complex foreground and background. The discriminative model ratio is proved to be a pseudo-likelihood ratio modeling the appearances. (2) Integral volume and a set of 3D Haar filters are used to achieve efficient computation. (3) We devise a 3D topology representation, *grid-line*, to perform fast boundary evolution. The proposed algorithm has been tested on over 100 volumes of size $500 \times 512 \times 512$ at the speed of $2 \sim 3$ minutes per volume. The results obtained are encouraging.

1 Introduction

There have been many 3D segmentation methods proposed recently [4, 11, 14]. In these methods, Gaussian/mixture/non-parametric i.i.d. forms, or Markov random fields are often adopted to model the appearances/textures of patterns of interest. Often, they have problems in dealing with situations in which the foreground and the background are complex and confusing.

Virtual colonoscopy is a new technology being developed to find polyps in 3D CT data. However, patients currently are required to physically cleanse their colons before the examination, which is very inconvenient. By tagging the residual materials (stool) to make them bright in CT volumes, we can remove stool electronically [3, 15]. This process is also called *colon detagging*, which can be done if we can successfully perform colon segmentation since residual materials are always inside the colon. However, residual materials observe large variations in appearance depending upon where they are, what the patients eat, and how they are tagged. Fig. (1) shows a view of a typical 3D CT volume. There are two types of objects inside a colon, air and stool. Though most of them appear to be

either very dark (air) or very bright (if successfully tagged), there are still a large portion of residual materials which have similar intensity values as tissues due to poor tagging. Also, some colon walls are bright due to the interference of the surrounding tagged stool. In addition, there are two types of tagging methods, liquid or solid, in which residual materials have very different textures. Fig. 9 shows some examples. For an input volume, we don't know what type of tagging it is and it can even a mixture of both. All these factors make the task of colon detagging very challenging.



(a) (b)

Fig. 1. Examples of clean and tagged CT volumes. (a) gives a view of a physically cleansed volume. The bright parts are bones in this volume. (b) is a view of an uncleansed volume. The bright parts on the upper part of the volume are tagged materials and the lower parts are bones same as in (a).

In this paper, we propose a learning based algorithm for 3D segmentation and show its application on colon detagging. The algorithm learns the appearance models for the foreground and the background based on a large set annotated data by experts. The system therefore is highly adaptive and nearly has no parameter to tune. To account for the large intra-class variation, we adopt *probabilistic boosting-tree* [9] to learn the discriminative models. One common solution in 3D segmentation is to define/learn high-level shape models and use them as priors in defining a posterior distribution. High-level knowledge or more specifically, contextual information, plays a key role in telling whether some part belongs to colon or background. However, not only is high-level knowledge very hard to capture, but also it introduces additional computational burden in the inference phase. Instead, we put the support of contextual information implicitly in discriminative models, which are nicely turned into pseudo-likelihood appearance model ratio. This is done by learning/computing the discriminative models of each voxel based on its surrounding voxels. The use of PBT approach has several advantages over many existing discriminative methods [2, 1]. First, it inherits the merit in the boosting methods which select and fuse a set of weak classifiers from a very large pool of candidates. Second, it outputs a unified discriminative probability through a hierarchical structure. Third, combined with integral volume and 3D Haar filters, it achieves rapid computation.

Here, we design a 3D representation, *grid-line*, for boundary evolution. In spirit, it is similar to the discrete surface model proposed by Malandain et al. [7]. Instead of representing the topology implicitly [12], we code the region topology explicitly on the grid node of each slice of a volume. Thus, the neighborhood structure of the boundaries can be traced explicitly.

Training of the discriminative models is performed on 10 typical volumes. The overall system is capable of robustly segmenting uncleansed colon for a volume of $500 \times 512 \times 512$ in $2 \sim 3$ minutes on a modern PC. It has been tested on around 100 volumes with fixed setting and we report some results in Sect. (6).

2 Problem formulation

In this section, we give the problem formulation for 3D segmentation and show that the pseudo-likelihood ratio is essentially a discriminative model ratio, which can be learned and computed by a discriminative learning framework. We start our discussion with an ideal model and show that the pseudo-likelihood model is an approximation to it.

2.1 An Ideal Model

For an input volume, $\mathbf{V}$, the task of foreground and background segmentation in 3D is to infer what voxels belong to the foreground and what voxels belong to the background. The solution W can be denoted as

$$W = ((R_{-1}, \theta_{-1}), (R_{+1}, \theta_{+1})),$$

where R_{-1}, R_{+1}, θ_{-1}, and θ_{+1} are the domains (voxel set) and model parameters for the background and the foreground respectively. We have $R_{-1} \bigcup R_{+1} = \Lambda$ where Λ defines the 3D lattice of the input $\mathbf{V}$, which is the set of all the voxels. $R_{-1} \cap R_{+1} = \emptyset$. The optimal solution W^* can be inferred by the Bayesian framework

$$\begin{aligned} W^* &= \arg max_W \; p(W|\mathbf{V}) \\ &= \arg max_W \; p(\mathbf{V}|(R_{-1}, \theta_{-1}), (R_{+1}, \theta_{+1})) \cdot p((R_{-1}, \theta_{-1}), (R_{+1}, \theta_{+1})). \end{aligned} \quad (1)$$

This requires the knowledge about the complex appearance models of the foreground and the background, their shapes, relations, and configurations. This "ideal" model is often out of reach in reality.

2.2 Pseudo-likelihood Models

A popular model for segmentation is the Mumford-Shah model [8]

$$\int_{\omega} (u - u_0)^2 dxdy + \mu \int_{\Omega/C} |\nabla u|^2 dxdy + \nu |C|.$$

The first term is the fidelity term encouraging the estimation u to be similar to the observation u_0, the second term penalizes big change in u, and the third term favors compact regions. Many similar models assume i.i.d. likelihood in modeling the texture. They are usually hard to resolve the confusion between the foreground and the background. The first column in Fig. (4) shows two slices along different planes in a volume. The second column in Fig. (4) displays the results by doing thresholding at an optimal value. We observe the "ring" effect

due to the influence of tagged materials to the air. These interface voxels have similar intensity patterns as the background. Intuitively, the decision of where to place the colon boundary should be made jointly according to the overall shape and appearance of a colon. This information can be accounted in the "ideal" models discussed before. However, we don't know what ideal models are and it is very difficult to learn and compute them in reality. Therefore, we seek approximations to the "ideal" models.

Let a segmentation result now be $W = (R_{-1}, R_{+1})$, where R_{-1} and R_{+1} are the domains for the background and foreground respectively. Instead, we can put the contextual information into a model as

$$\hat{p}(W|\mathbf{V}) \propto \prod_{s\in R_{-1}} p(\mathbf{V}(s), y=-1|\mathbf{V}(N(s)/s)) \cdot \prod_{s\in R_{+1}} p(\mathbf{V}(s), y=+1|\mathbf{V}(N(s)/s)) \cdot p(R_{-1}, R_{+1}), \tag{2}$$

where $N(s)$ is the sub-volume centered at voxel s, $N(s)/s$ include all the voxels sub-volume except for s, and $y \in \{-1, +1\}$ is the label for each voxel, and $p(R_{-1}, R_{+1})$ defines the shape prior of the colon border. Our goal is to find the optimal W^* that maximizes the posterior $\hat{p}(W|\mathbf{V})$. Next, we show how to learn these models. Let

$$-\log \hat{p}(W|\mathbf{V}) = E_1 + E_2 + E_c$$

where E_c is a constant and does not depend on R_{-1} and R_{+1},

$$E_2 = -\log p(R_{-1}, R_{+1}),$$

and

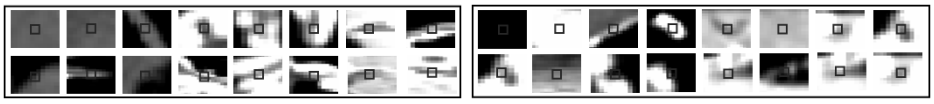
$$\begin{aligned} E_1 &= -\sum_{s\in R_{-1}} \log p(\mathbf{V}(s), y=-1|\mathbf{V}(N(s)/s)) - \sum_{s\in R_{+1}} \log p(\mathbf{V}(s), y=+1|\mathbf{V}(N(s)/s)) \\ &= -\sum_{s\in \Lambda} \log p(\mathbf{V}(s), y=-1|\mathbf{V}(N(s)/s)) - \sum_{s\in R_{+1}} \log \frac{p(y=+1|\mathbf{V}(N(s)))p(y=-1)}{p(y=-1|\mathbf{V}(N(s)))p(y=+1)}. \end{aligned} \tag{3}$$

This is done by taking a common part for $p(\mathbf{V}(s), y=-1|\mathbf{V}(N(s)/s)$ in R_{+1}. The first term in the above equation does not depend on R_{-1} and R_{+1}. Therefore, maximizing the probability $\hat{p}(W|\mathbf{V})$ is equivalent to minimizing the energy

$$E = -\sum_{s\in R_{+1}} \log \frac{p(l=+1|\mathbf{V}(N(s)))}{p(l=-1|\mathbf{V}(N(s)))} - |R_{+1}| \cdot \log \frac{p(y=-1)}{p(y=+1)} - \log p(R_{-1}, R_{+1}), \tag{4}$$

where $|R_{+1}|$ is the size of volume of R_{+1}. Here, the models capturing the appearances of foreground and background are nicely turned into the discriminative probability model (classification) ratio. Note that $p(y=+1|\mathbf{V}(N(s)))$ is the posterior probability of a voxel s belonging to the foreground (colon) given the sub-volume centered at s. The optimal segmentation W^* is the one that minimizes the above energy E.

3 Learning Discriminative Models



(a) Slice view of background sub-volume. (b) Slice view of colon sub-volume.

Fig. 2. Slice view of 3D sub-volumes of background and colon. We consider the center voxel here. They observe large intra-class variability and inter-class similarity.

Now the task is to learn and compute the discriminative model $p(y|\mathbf{V}(N(s)))$ for each voxel s given a sub-volume centered at s. As shown in Fig. (2), both the foreground and the background show complex patterns. Therefore, in order to make a firm decision, we need to combine various types of information together, e.g., intensities, gradients, and the surrounding voxels in the sub-volume.

AdaBoost algorithm [5] proposed by Freund and Schapire combines a number of weak classifiers into a strong classifier $H(x) = sign(f(x)) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$. Moreover, it is proved that AdaBoost and its variations are asymptotically approaching the posterior distribution [6].

$$p(y|x) \leftarrow q(y|x) = \frac{exp\{2yf(x)\}}{1 + exp\{2yf(x)\}}. \tag{5}$$

However, AdaBoost algorithm is still shown to be rigid and hard to deal with large intra-class variation. We adopt a new learning framework, probabilistic boosting tree [9], to learn complex discriminative models.

3.1 Probabilistic Boosting-Tree

The details of the discussion of PBT can be found in [9]. It has also been applied to learn affinity maps in perceptual grouping in [10]. We use it to learn appearance models here. The algorithm is intuitive. It recursively learns a tree. At each node, a strong classifier is learned using a standard boosting algorithm. The training samples are then divided into two new sets using the learned classifier, the left one and the right one, which are then used to train a left sub-tree and right sub-tree respectively. Under this model, positive and negative samples are naturally divided into sub-groups. Fig. (3b) illustrates an abstract version of a tree learned. Samples which are hard to classify are passed further down leading to the expansion of the tree. Since each tree node is a strong classifier, it can deal with samples of complex distributions. Compared with other existing hierarchical discriminative models, PBT learns a strong classifier at each tree node and outputs a unified posterior distribution.

During the testing stage, the overall discriminative model is computed as

$$\begin{aligned} p(y|x) &= \sum_{l_1} \tilde{p}(y|l_1, x) q(l_1|x) = \sum_{l_1, l_2} \tilde{p}(y|l_2, l_1, x) q(l_2|l_1, x) q(l_1|x) \\ &= \sum_{l_1, .., l_n} \tilde{p}(y|l_n, ..., l_1, x), ..., q(l_2|l_1, x) q(l_1|x). \end{aligned}$$

The procedure is consistent with the training stage. For each sample, it computes a probability at each node. Then the sample is sent to either the left, the right, or both sides of the tree based on this probability. At the top of the tree, information is accumulated from its descendants and an overall posterior distribution is reported. It is worth to mention that the Vapnik-Chervonenkis dimension theory shows that the test error is bounded by

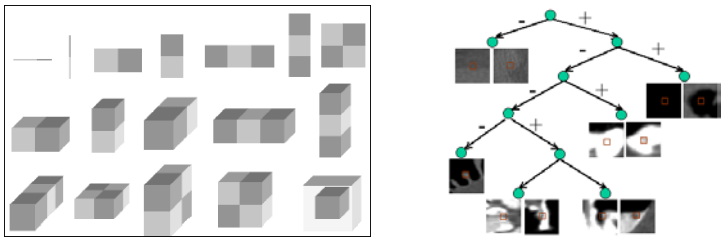
$$TESTERR(\alpha) \leq TRAINERR(\alpha) + \sqrt{\frac{h(\log(2N/h + 1 - \log(\eta)}{N}},$$

where N is the number of training samples and h is the VC dimension of a classifier. In PBT, h is decided by the complexity of weak classifiers d, the number of classifiers on each tree node T, and the maximum depth of the tree L. By extending a derivation from [5]

$$h(PBT) \leq 2(d+1)(T+1)\log_2[e(T+1)](2^L - 1).$$

In this application, to keep the test error under check, we set the maximum depth of the tree to be 9 and train a classifier with half a million samples through bootstrapping.

3.2 Weak Classifiers and Features

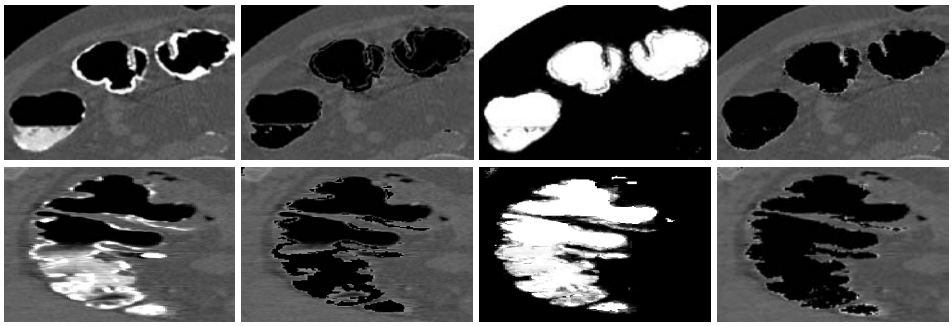


(a) 1D, 2D, and 3D Haar filters (b) Illustration of a PBT learned

Fig. 3. (a) shows various Haar filters in 1D, 2D, and 3D used. (b) illustrates an abstract version of a tree learned.

Each training sample is of size $31 \times 31 \times 31$ and we want to learn a classification model $p(y|\mathbf{V}(N(s)))$ for the center voxel s. PBT selects and combines a set of weak classifiers into a strong classifier out of a large number of candidates. For a training sample, the features are the intensity and gradient values, curvatures at the center voxel and its surrounding voxels. Also, we design 1D, 2D, and 3D Haar filters at various locations with different aspect ratios, which are shown in Fig. (3). Therefore, local and context information are combined to give an overall decision on how likely a voxel is on the colon or not. There are around 25,000 candidate features each of which corresponds to a weak classifier.

For an input volume, we compute *integral volume* first, similar to the *integral image* used in [13]. At each location (x_1, y_1, z_1), an integral volume is computed $\int_{x_1}\int_{y_1}\int_{z_1} V(x,y,z)dxdydz$. The computational cost of computing Haar filters is



(a) (b) (c) (d)

Fig. 4. a) shows two slice views of part of a volume. b) illustrates the results by thresholding at an optimal value. We can clearly see some "ring" effects and a big part of colon in the second row is not removed. c) displays the saliency (probability) maps $p(y = +1|\mathbf{V}(N(s)))$. The higher the intensity values, the more likely it belongs to the foreground colon. d) illustrates the results by thresholding on $p(y = +1|\mathbf{V}(N(s)))$ at 0.5. The results are much better than direct thresholding in (b) though it is bit jagged. This is ameliorated by using the $p(y = |\mathbf{V}(N(s)))$ as a soft value with a local shape prior in the energy minimization formulation. Fig. (8) shows improved results by the overall algorithm.

therefore largely reduced since every time we only need to sum up the values of corners of the Haar in the integral volume. Also, due to the tree structure as shown in Fig. (3).b, majority of the sub-volumes are only passed onto the top levels. Fig. (4) shows some results. We see the improvement on the place where context information is needed. Training of discriminative models is performed on 10 typical volumes (by liquid and solid tagging) with a couple of rounds of bootstrapping. We also implemented two other approaches, one node AdaBoost and a cascade of AdaBoost. The training errors for both the methods are significantly worse than that by PBT. For the cascade approach, the training error can not decrease too much after 4 levels due to the confusing patterns of the foreground and the background.

4 3D Representation for Boundary Evolution

Once we compute the discriminative model $p(y|\mathbf{V}(N(s)))$ for each voxel s, we then need to search the optimal segmentation that minimizes the energy E in eqn. (4). If we only do thresholding at 0.5 based on $p(y = +1|\mathbf{V}(N(s)))$, as shown in Fig. (4).d, the colon borders are not so smooth.

A popular implementation for boundary evolution in variational method is by level-set approaches [12]. Here, we design another 3D representation, *grid-line*, for fast boundary evolution, which is in spirit similar to [7]. Instead of representing the topology implicitly by different level sets, we code the topologies explicitly on the grid node of each slice of a volume. Thus, the neighborhood structure of the boundaries can be traced explicitly. Fig. (5) illustrates an example. For each voxel in the volume $\mathbf{V}$, we explicitly code its label by $+1$ if it is on the foreground (colon part), and -1 if is on the background. With the

label map only, it does not easily facilitate the process of boundary evolution. We also code the segmentation topology at each slice along XY, XZ, and YZ planes. On each slice, boundary nodes have 4 corners with two types of labels. We code each possible situation for a boundary node on the grid. This is illustrated in Fig. (5c). Given any grid node on the boundary, we can obtain its most immediate nodes (clockwise or counter clockwise)) based on the configuration of the current node and its 4 connected neighboring nodes (special care needs to be taken on the nodes along the edge of the volume). Therefore, at each grid node on the boundary, we can explicitly compute its normal direction, curvature etc. Also, the explicit 3D representation allows us to have the property that the foreground is connected. This is often a desirable property in object specific 3D segmentation in which occlusion usually does not exist.

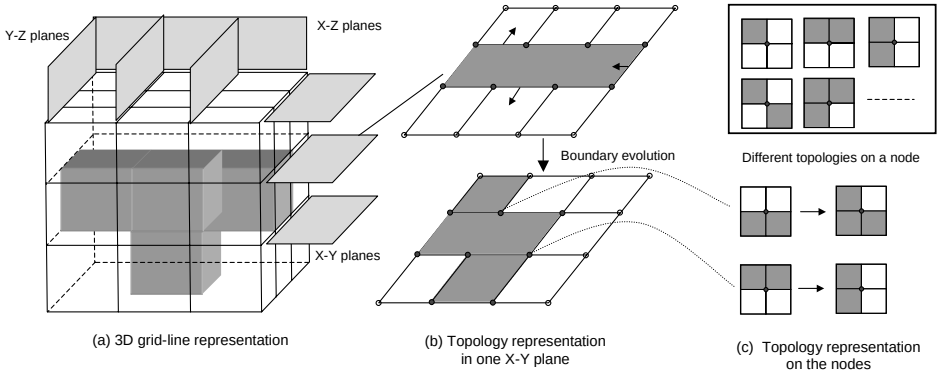


Fig. 5. A 3D topology representation for boundary evolution. In the volume shown in (a), we explicitly code the label of each voxel being either on the foreground, $+1$, or on the background, -1. In addition, we code the topology of each grid node of slices at the XY, XZ, and YZ planes. This is illustrated in (b). (c) lists various possible topologies of a grid node on the boundary. We also show an example of a boundary move in (b) and (c).

The term $p(R_{-1}, R_{+1})$ for shape prior is left undefined in eqn. (4). Indeed, part of the shape information is implicitly modeled in the discriminative model $p(y|\mathbf{V}(N(s)))$. Intuitively, the possibility of a voxel label is decided by its own intensity and the appearances of its surrounding voxels based on various features including gradients and curvatures. This implicitly reinforces certain degree of spatial coherence. In addition, we put an explicit shape prior term to encourage the boundaries to be smooth. Let A be the surface between R_{-1} and R_{+1}

$$-\log p(R_{-1}, R_{+1}) = \alpha \int_A ds$$

By Euler-Lagrange equation on E in eqn. , we obtain eqn. (4), we have

$$\frac{dE}{ds} = -(\log \frac{p(y=+1|\mathbf{V}(N(s)))}{p(y=-1|\mathbf{V}(N(s)))} + \log \frac{p(y=-1)}{p(y=+1)} + \alpha H)\mathbf{n}$$

where H and $\mathbf{n}$ are the mean curvature and normal direction at s respectively. The boundary evolution is performed using the above evolution equation based on the grid-line representation discussed above.

5 Outline of the Algorithm

The outline of the overall algorithm is illustrated below.

- Given an input volume, compute $p(y|\mathbf{V}(N(s)))$ for each voxel s.
- Perform thresholding on $p(y|\mathbf{V}(N(s)))$.
- Find seed regions in 2D slices and perform morphological region growing to obtain an initial 3D segmentation.
- Perform boundary evolution.
- Remove the segmented colon part in the original volume to perform detagging.
- Report the final segmentation results.

Fig. 6. Outline of the overall algorithm.

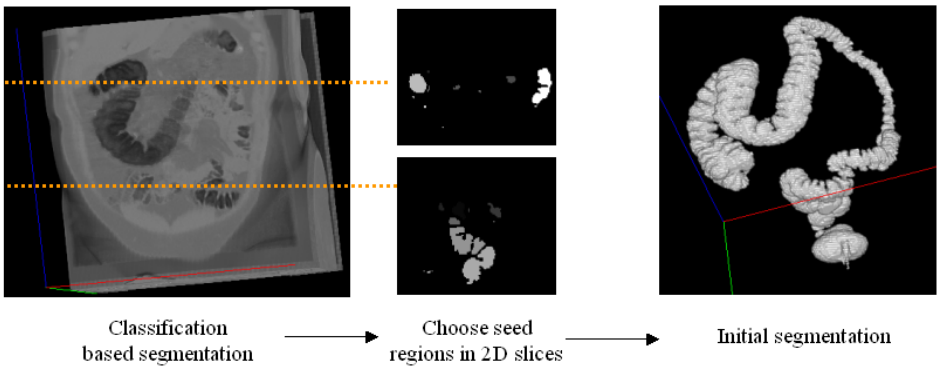


Fig. 7. Initial segmentation. The image in the left column shows the volumes by thresholding at 0.5 for $p(y = +1|\mathbf{V}(N(s)))$. We then obtain a number of slice images on the thresholded volumes along the XY planes. The colon part in these slices appear to be more or less round. Some seed regions are the selected based on its size and shape. These are shown in the middle of the figure. An initial segmentation, shown in the right, is then obtained using morphological region growing.

After computing the discriminative models, the algorithm further proceeds for two more steps: (1) Based on thresholding on $p(y = |\mathbf{V}(N(s)))$, sample slices are taken along the XY plane to select some regions which show round shapes. We then use morphological region growing to obtain an initial 3D segmentation. (2) We perform boundary evolution method discussed in the previous section to obtain refined segmentation.

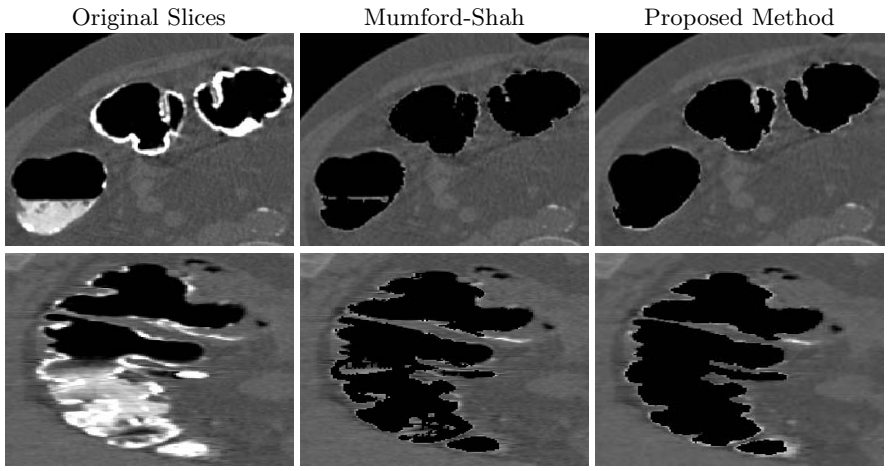


Fig. 8. The first column shows two original slices. The second column some results by Mumford-Shah model. The results by the proposed algorithm is shown in the third column.

6 Experiments

We use 10 typical volumes for training. Fig. (8) shows the results by our method on a testing volume and those by Mumford-Shah model. We see some improvements on the place where context information is needed. The boundaries obtained are smoother than using just classification in the last row of Fig. (4). We have tested the reported algorithm on 100 volumes with the same setting and the results are very promising. Four of which are shown in Fig. (9). The first 3 volumes are by solid tagging and the last one is by liquid tagging. Since it is very hard to obtain the ground truth for even a single 3D volume, we measure the error by comparing the results with manual annotation at some typical slices by experts. We use randomly selected 20 volumes with 15 slices in each volume. The measurement is taken by the difference of the overlaps $error = (miss(R_{+1}) + miss(R_{-1}))/|\mathbf{V}|$, where $miss(R_{+1})$ is the number of miss segmented voxels in the foreground and $miss(R_{-1})$ is the number of miss segmented voxels in the background. The error rate is lower than 0.1% by the algorithm. If we only consier those voxels that are within certain distance of the true boundary, the error rate is 5.2% while it is 20.3% for direct thresholding. Bones in these volumes appear to be very bright and their local sub-volumes look very like tagged materials. An example can be seen in Fig. (1). In our algorithm, the seed selection stage avoids picking up bones since they don't have round structures in 2D slices. Also, by enforcing the foreground regions to be connected in the boundary evolution, bones will not be touched in the boundary evolution stage. This is an important feature in our method. Existing methods [3, 15] only deal with liquid tagging, like the one in the fifth column in Fig. (9), which is relatively easy. Also, they usually do not distinguish between bones and colons, leading bones being mistakenly removed.

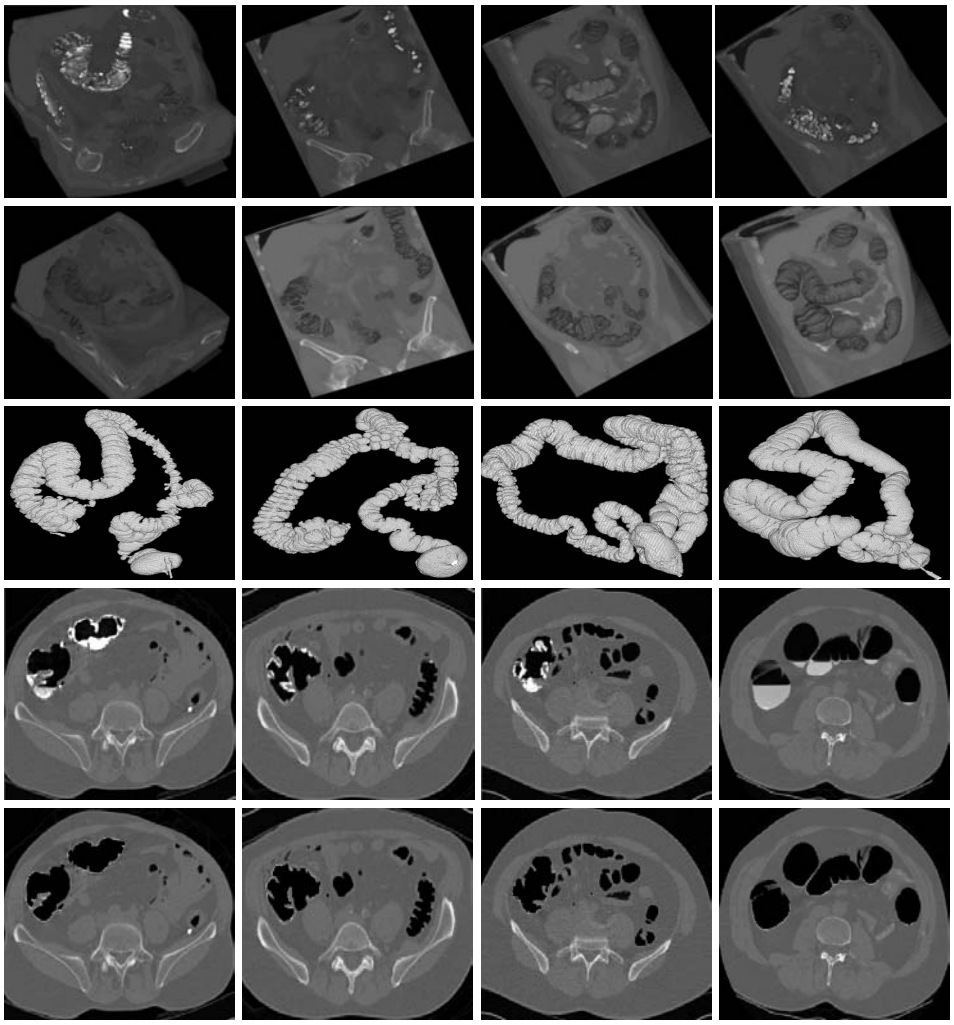


Fig. 9. Some results on colon segmentation. The first rows shows some input volumes. The first three uses solid tagging and the last one uses liquid tagging. The second row shows the volume by segmenting out the colon volume. The third row demonstrates the colon part only. The fourth row illustrates some 2D slice views of the original volume. The last row shows the corresponding views after detagging.

7 Discussion

In the paper, we have introduced a new learning based framework for 3D segmentation and shown its application on colon detagging. We use a probabilistic boosting tree (PBT) method to learn pseudo-likelihood models for the complex patterns. Integral volume and 1D, 2D, and 3D Haar wavelets are designed for fast computation. A 3D representation is used to efficiently evolve the boundary.

This gives rise to a system capable of automatically segmenting colon volume of $512 \times 512 \times 400$ in $2 \sim 3$ minutes. There is no need to specify liquid or slid tagging, and the system is fully automatic. Also, the system learns the model based on a large database of annotation, which makes it very general and highly adaptive. It can be used in many problems in medical imaging and computer vision.

Our algorithm still has some problems to deal with situations where stool is very poorly tagged. The sub-volume used in computing the discriminative models is yet not big enough to capture big scope of context. Increasing it size will largely increase the complexity of the learner. It still remains to see how to combine high-level shape prior to further improve the results.

References

1. K. P. Bennett and J. A. Blue, "A Support Vector Machine Approach to Decision Trees", *Proc. of IJCNN*, 1998.
2. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and Regression Trees", *Wadsworth International, Belmont, Ca*, 1984.
3. D. Chen, Z. Liang, M. Wax, L. Li, B. Li, and A. Kaufman, "A Novel Approach to Extract Colon Lumen from CT Images for Virtual Colonoscopy", *IEEE Tran. Medical Imaging*, vol. 19, no. 12, 2000.
4. R.H. Davies, C.J. Twining, T.F. Cootes, J.C. Waterton, C.J. Taylor, "3D Statistical Shape Models Using Direct Optimisation of Description Length", *ECCV*, 2002.
5. Y. Freund and R. Schapire, "A Decision-theoretic Generalization of On-line Learning And an Application to Boosting", *J. of Comp. and Sys. Sci.*, 55(1), 1997.
6. J. Friedman, T. Hastie and R. Tibshirani, "Additive logistic regression: a statistical view of boosting", Dept. of Statistics, Stanford Univ. Technical Report. 1998.
7. G. Malandain, G. Bertrand, N. Ayache, "Topological Segmentation of Discrete Surfaces", *Intl. J. of Computer Vision*, 10(2), 1993.
8. D. Mumford and J. Shah, "Optimal approximation by piecewise smooth functions and associated variational problems", *Comm. Pure Appl. Math.*, 42:577685, 1989.
9. Z. Tu, "Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering", *Proc. of ICCV*, 2005.
10. Z. Tu, "An Integrated Framework for Image Segmentation and Perceptual Grouping", *Proc. of ICCV*, 2005.
11. S. Pizer et al., "Deformable M-Reps for 3D Medical Image Segmentation", *Intl. J. of Computer Vision*, 55(2), 2003.
12. J.A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 1999.
13. P. Viola and M. Jones, "Fast Multi-view Face Detection", *Proc. of CVPR*, 2001.
14. A. Yezzi and S. Soatto, "Stereoscopic segmentation", *Intl. J. of Computer Vision*, 53(1), 2003.
15. M. E. Zalis, J. Perumpillichira, and P. F. Hahn, "Digital Subtraction Bowel Cleansing for CT Colonography Using Morphological and Linear Filtration Methods", *IEEE Trans. Medical Imaging*, vol. 23, no. 11, 2004.