

# Hierarchical Part-Based Detection of 3D Flexible Tubes: Application to CT Colonoscopy

Adrian Barbu<sup>1</sup>, Luca Bogoni<sup>2</sup>, and Dorin Comaniciu<sup>1</sup>

<sup>1</sup> Siemens Corporate Research, Princeton NJ

<sup>2</sup> Siemens CAD, Malvern, PA

**Abstract.** In this paper, we present a learning-based method for the detection and segmentation of 3D free-form tubular structures, such as the rectal tubes in CT colonoscopy. This method can be used to reduce the false alarms introduced by rectal tubes in current polyp detection algorithms. The method is hierarchical, detecting parts of the tube in increasing order of complexity, from tube cross sections and tube segments to the whole flexible tube. To increase the speed of the algorithm, candidate parts are generated using a voting strategy. The detected tube segments are combined into a flexible tube using a dynamic programming algorithm. Testing the algorithm on 210 unseen datasets resulted in a tube detection rate of 94.7% and 0.12 false alarms per volume. The method can be easily retrained to detect and segment other tubular 3D structures.

## 1 Introduction

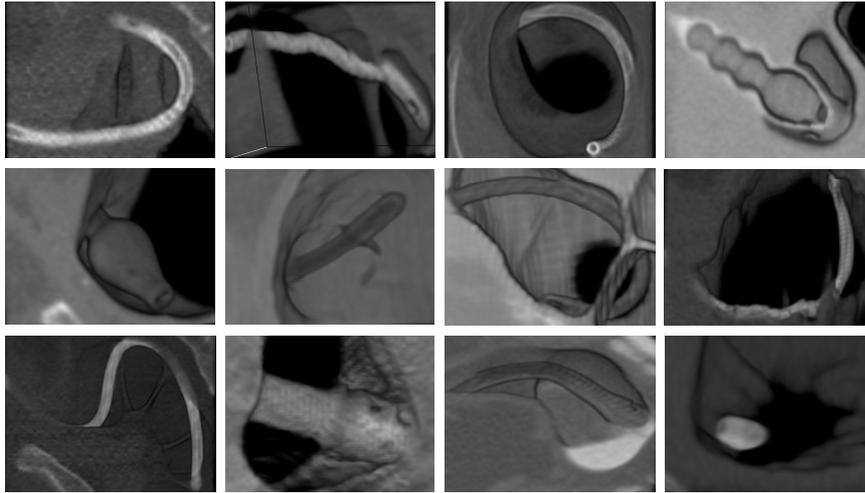
Prevention of colon cancer can be achieved by detecting and surgically removing the polyps from the colon wall. However, the colonoscopy procedure used for detecting the polyps is a time consuming procedure that produces great discomfort for the patient. Virtual colonoscopy is an increasingly popular alternative, in which the patient's colon is inflated with air through a rectal tube and then one or two CT scans of the abdomen are performed. A polyp detection algorithm is run on the CT scans and the detection results are reported to the doctor for inspection. The current polyp detection algorithms exhibit a relatively large number of false alarms due to the rectal tube used to inflate the colon. These false alarms can be reduced by detecting and segmenting the rectal tube and discarding any potential positives that are very close to the rectal tube.

A rectal tube detection algorithm should be fast and have very low false alarm rate, since false alarms can decrease the detection rate of the overall polyp detection system.

In this paper we present a supervised learning approach to rectal tube detection, where the detector is trained from manually annotated data to obtain greater robustness. As one can see in Figure 1, the rectal tubes have a great degree of variability in length, shape and appearance.

A previous method for RT detection [2] handled the appearance by template matching, which is a relatively rigid method for detection, and the shape variability by tracking 2D slices. The tracking assumes that the tube is relatively

perpendicular to one of the axes, which is often not true, as shown in Figure 1. The method [2] only handled two types of rectal tubes and was validated on a relatively small number of cases (80 datasets). It involved a large amount of potentially time consuming morphological operations (region growing).



**Fig. 1.** The rectal tubes are flexible and have variable shape and appearance

The other method [4] for reduction of false positives due to rectal tubes involved using a Massive Trained Artificial Neural Network (MTANN) to distinguish between polyps and rectal tubes, which raises questions about the degree of control of the generalization power of the system.

In this paper we take a hierarchical, part based learning approach to rectal tube detection and segmentation, which provides a large degree of control against overfitting the data. Tube parts are detected, starting from simpler structures (tube cross-sections) with a smaller number of parameters, to more complex structures (short tube segments), using learned discriminative models. A top-down algorithm based on dynamic programming constructs the final tube segmentation from the detected sub-parts. A voting strategy is used to greatly reduce the search for location, direction and size of the cross-sections.

The training algorithm is based on a Probabilistic Boosting Tree [5] that yields a robust detector with good generalization power. The features used are not based directly on intensity, but they are based on gradient and 3D curvature. All features are invariant to axial rotation, to avoid overfitting the data.

## 2 Description of Our Method

The input of our system is a  $512 \times 512 \times N$ ,  $N \sim 500$ , isometric CT volume and a set of query locations that are the output of a first stage of polyp detection.

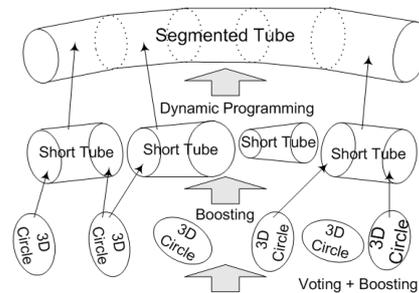
The rectal tube detection algorithm assigns a label "RT"/"non-RT" to each query location, stating whether the query location is at distance of at most 5mm from a Rectal Tube or not. The query locations labeled as "non-RT" are passed through a second stage of polyp detection using a more involved algorithm, and the remaining locations are reported to the physician as detected polyps.

## 2.1 Preprocessing

For each CT volume there are on average 50 query locations as input to our Rectal Tube detector. Of those, about 40 locations are automatically labeled as negative because they are outside of a predefined box of size 200x350x200 in the caudal region of the CT volume, where all rectal tubes of the training data have been found to reside.

The remaining locations are clustered together by computing connected components in a graph. There is an edge between two locations if they are less than 35 voxels apart. For each cluster (connected component of the graph), the bounding box is computed and enlarged by 30 voxels on each side. The corresponding subvolume is cropped and the tube segmentation algorithm described below is used in the subvolume. This way, candidate locations that are clustered together will be processed at the same time. Any location that is closer than 5mm from the segmented tube is labeled as "RT", and the rest as "non-RT".

The rest of the paper is concentrated on the detection and segmentation of the rectal tube in a cropped subvolume.



**Fig. 2.** In our hierarchical learning based method, more and more complex parts are gradually constructed, starting with 3D Circles (tube cross-sections), which are combined into short tubes, that are finally combined into a long flexible tube

## 2.2 Overview of Our Framework

Using a trained classifier to detect rectal tubes provides a convenient way to manage the generalization power and the false alarm rate of the system. The price to pay is the computational expense to search for all possible parameters that define the classifier. It is practically impossible to detect the entire rectal tube using a single classifier, because there are too many parameters to search,

the tube being flexible, as shown in Figure 1. Instead, we use a part-based approach, starting with simpler and more rigid shapes (tube cross-sections) and connecting them into short tubes and then into a long free-form tube. To detect the tube cross-sections, or "3D circles", the algorithm should apply the trained detector at all possible locations  $X = (x, y, z)$ , directions  $D = (d_x, d_y, d_z)$  and radii  $R$ , which is computationally prohibitive. Instead, we choose to restrict the application of the detector to promising locations, given by a voting strategy.

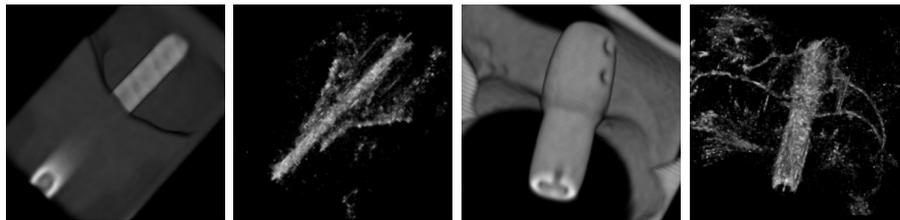
Our rectal tube segmentation method proceeds as follows:

1. Candidate tube cross-sections with parameters  $C = (X, D, R)$  (location, direction and radius respectively) are found using a **voting** scheme, as described in subsection 2.3.
2. A trained **3D Circle detector**, described in 2.4, is used to keep only the most promising cross-sections of the 3D tube.
3. **Short tubes** are detected from 3D circle pairs, as described in 2.5. First, for each detected 3D circle, the 10 best aligned 3D circles on each side are found. Then a trained short tube detector is used to prune away pairs that are not supported by the data.
4. A **dynamic programming** algorithm uses the short tubes and their adjacency graph to find the best tube segmentation.

The following subsections describe in more detail each of the above steps.

### 2.3 Proposing 3D Circles by Voting

Inside the cropped subvolume (typically of size 60x60x60 ), the gradient at all locations is computed. At the places where the gradient is larger than 100, the 3D curvatures and the principal directions of the curvature are computed.



**Fig. 3.** The voting strategy gives possible locations for the axes of the detected tubes

The voting proceeds as follows. Each voxel  $\mathbf{x}$  casts one vote at the location  $v(\mathbf{x})$  in the direction of the gradient  $g_{\mathbf{x}}$  at distance equal to the inverse of the largest curvature  $k(\mathbf{x})$ . That is, the vote is casted at location  $v(\mathbf{x}) = \mathbf{x} + \frac{g_{\mathbf{x}}}{|g_{\mathbf{x}}|} \frac{1}{k(\mathbf{x})}$ . For a tubular structure, all locations on a tube cross-section will vote the center of the cross-section. The votes for two input tubes are shown in Figure 3, white representing 5 votes. At locations  $\mathbf{y}$  with at least 5 votes, the tube direction is

computed as the median of the second principal directions at locations  $\mathbf{x}$  that voted  $\mathbf{y}$ , i.e.  $v(\mathbf{x}) = \mathbf{y}$ . In that direction, the most promising 3D Circles  $C_{\mathbf{y}}(R)$  are obtained by computing the voting number

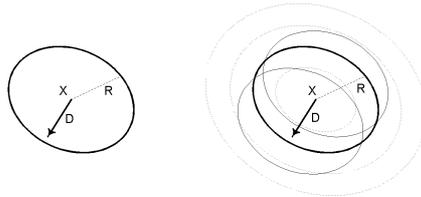
$$N_{\mathbf{y}}(R) = |\{\mathbf{x} \in C_{\mathbf{y}}(R), 0.5 \leq R * k(\mathbf{x}) \leq 2\}| / |C_{\mathbf{y}}(R)| \quad (1)$$

for some discretization of  $C_{\mathbf{y}}(R)$ . For a perfect tube or radius  $R$  and  $\mathbf{y}$  on its axis, all  $\mathbf{x} \in C_{\mathbf{y}}(R)$  would have curvature  $k(\mathbf{x}) = 1/R$  and the voting number  $N_{\mathbf{y}}(R)$  would be  $\pi$ . In reality, we keep the candidate circles  $C_{\mathbf{y}}(R)$  having  $N_{\mathbf{y}}(R) \geq 1.3$ .

## 2.4 Training the 3D Circle Detector

The 3D circle detector is specialized in detecting cross-sections of the rectal tube. The parameters of a 3D Circle are (see Fig. 4):

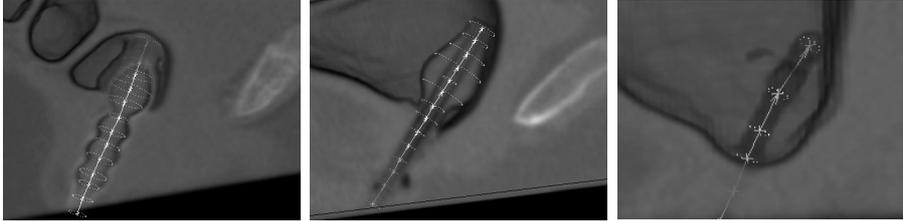
- the center location  $X = (x, y, z)$ .
- the direction  $D = (d_x, d_y, d_z)$ ,  $|D| = 1$ , normal to the plane of the 3D circle.
- the radius  $R$  of the 3D circle.



**Fig. 4.** Left: The model and parameters of a 3D circle. Right: the features are computed on 12 circles (3 locations and 4 radii) relative to the 3D circle.

To avoid overfitting the data, all the features are invariant to rotation about the 3D Circle axis. The features are obtained as 8 types of axial invariant statistics: mean, variance, central symmetry mean, central symmetry variance, 25, 50 and 75 percentile and voting number. Each invariant statistic is computed on one of  $4 \times 3 = 12$  circles, having one of 4 radii ( $R/3, R, 5/3R, 7/3R$ ) and one of 3 locations along the circle direction ( $X$  and  $X \pm 2D$ ). Each of the 12 circles is discretized and subsampled and one of the 8 types of statistics is computed for one of 70 different combinations of gradient, curvature and principal directions (sum, difference, product, etc.). In total there are  $8 \times 4 \times 3 \times 70 = 6720$  features.

For training, the rectal tubes of 154 CT volumes are annotated using a generalized cylinder model. There are 3 different types of tubes in the training data. We have a semiautomatic algorithm based on dynamic programming to compute a tube annotation given two manually marked endpoints. The algorithm produces circle sections of the tube spaced 10 voxels apart, starting from one endpoint of the tube and ending in the other endpoint. The circle locations and radii are manually corrected to obtain the best alignment possible. The annotations of three volumes are shown in Figure 5.



**Fig. 5.** Manual annotation of different types of rectal tubes from our dataset

For training of the 3D circle detector, 15000 positive examples are generated from the manual annotations by interpolation, excluding the region close to the tip of the tube where are lateral holes.

From the candidate locations obtained by voting, 207000 samples that are at distance of at least 35 voxels from the tube annotations are chosen as negative examples.

The training algorithm used in this paper is the Probabilistic Boosting Tree (PBT) [5] that learns a binary tree of strong classifiers, where each node is trained by Adaboost [3], starting from the root. At each node, after training, the positives and negatives are run through the detector of that node and the detected positives and false alarms are passed as positives and negatives for the right subtree, while the rejected positives and negatives are passed as training data for the left subtree. After training, the PBT can assign a probability to any new sample, representing the learned probability that the new sample is a positive example.

A PBT with 6 levels is trained using 15 weak classifiers per node, with the first two levels enforced as a cascade. The detection rate on the training samples was 95.6% and the false alarm rate 1.7%.

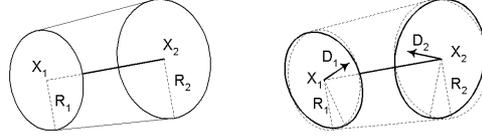
The 3D Circle detector usually misses the part of the tube that is not circular, where there are lateral holes in the tube. This will be corrected by the next level.

## 2.5 Detecting Short Tubes from the 3D Circles

The short tubes are the parts from which the dynamic programming algorithm described in the next subsection constructs the final segmentation. For good performance, there should be approximately the same number of short tubes starting at each of the detected circles.

For that, the short tubes are detected in two steps. In the first step, 10 candidate tubes are found on each side of any detected 3D Circle. For each 3D circle  $C_1 = (X_1, D_1, R_1)$ , the 10 neighbor circles  $C_2 = (X_2, D_2, R_2)$  with the smallest alignment cost  $A(C_1, C_2)$  are found. The alignment cost depends on the relative position of the circles, and their radii (see also Fig. 6):

$$A(C_1, C_2) = \alpha^2 + \beta^2 + 0.1(d - 10)^2 + 0.2(R_1 - R_2)^2 - 0.5(R_1 + R_2) \quad (2)$$



**Fig. 6.** Left: The parameters of a short tube. Right: for each pair of aligned 3D circles  $C_1 = (X_1, D_1, R_1), C_2 = (X_2, D_2, R_2)$  a short tube  $T = (X_1, X_2, R_1, R_2)$  is constructed.

where  $\alpha, \beta < \pi/2$  are the angles between the axis  $\overline{X_1 X_2}$  and  $D_1$  respectively  $D_2$ . This way, are preferred circles  $C_2$  which are best aligned in direction, with similar and large radii and at distance close to 10 voxels.

The parameters of a short tube are  $T = (X_1, R_1, X_2, R_2)$ . For each pair of aligned 3D circles  $C_1 = (X_1, D_1, R_1), C_2 = (X_2, D_2, R_2)$  found as above, a candidate short tube is constructed, with parameters  $T = (X_1, R_1, X_2, R_2)$ , using only the radii and positions of the 3D circles, as illustrated in Figure 6.

The second step validates the constructed short tubes using the input data. For that, we train a short tube detector and keep only the short tubes whose probability is greater than a threshold (0.35 in our case).

The short tube detector has the same features as the 3D circle detector, with the difference that the 12 circles on which the features statistics are computed have positions  $X_1, (X_1 + X_2)/2$  and  $X_2$  and radii  $R/3, R, 5/3R, 7/3R$  with  $R = R_1, (R_1 + R_2)/2, R_2$  respectively.

For training, 13700 positive samples 10 voxels long were created from the manual annotations.

Also, 40000 negatives were obtained, at locations, directions, and radii obtained by voting, all of length 10 and of identical radii  $R_1 = R_2$ . Another 9000 negatives were obtained from aligned pairs of 3D circles that are at least 35 voxels away from the manual annotations.

A PBT is trained with 6 levels, 20 weak classifiers per node, and first two levels enforced as cascade. The detection rate on the training samples was 95.1% and the false alarm rate was 3.6%.

We trained another "pure" short tube detector, with the same positives but with 198000 negatives of length 10 and identical radii  $R_1 = R_2$ . The detection rate on the training samples was 97.5% and the false alarm rate was 0.1%.

## 2.6 Tube Segmentation by Dynamic Programming

From the previous steps, we obtained a set of short tubes  $T = \{T_1, \dots, T_n\}$  and a graph  $G = (T, E)$  whose nodes are the short tubes  $T$ . Two short tubes  $T_i$  and  $T_j$  are connected through a graph edge  $E_{ij} \in E$  if they share one endpoint  $X$  and they have the same radius  $R$  at that endpoint, e.g.  $T_i = (A, R_1, X, R)$  and  $T_j = (X, R, B, R_2)$ . All edges  $E_{ij}$  for which the 3D angle  $\alpha_{ij} = \widehat{AXB}$  is not close to  $\pi$ , ( i.e.  $\alpha_{ij} < 5\pi/6$  or  $\alpha_{ij} > 7\pi/6$ ) are removed. The weight of the edge is a measure of good continuation of the tubes:

$$E_{ij} = |\alpha_{ij} - \pi| \tan |\alpha_{ij} - \pi| \tag{3}$$

There is also a unary cost for each short tube  $T = (X_1, R_1, X_2, R_2)$ :

$$c(T) = -\ln(P(T)) + 0.2(R_2 - R_1)^2 \quad (4)$$

where  $P(T)$  is the probability given by the trained short tube classifier.

In our dynamic programming framework, we denote by  $C_{ij}^k$  the cost of the best a chain of  $k$  short tubes starting with  $T_i$  and ending in  $T_j$ . Then we have the following recurrence formula:

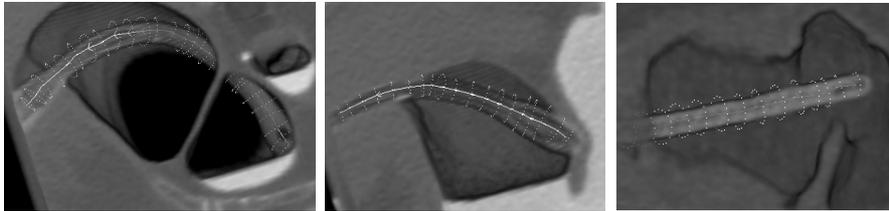
$$C_{ij}^{k+1} = \min_s [C_{is}^k + E_{sj} + c(T_j)] \quad (5)$$

For each  $k$  we find the chain of short tubes  $S_k$  corresponding to the smallest  $C_{ij}^k$ . The chain  $S_k$  with the largest length (in voxels) is the segmentation result.

### 3 Results

To illustrate the segmentation quality of our method, we show in Figure 7 a few segmentation results obtained on unseen data (not used for training). Testing has been performed on 210 datasets not used in training, containing about 5 types of rectal tubes. To validate our system, we present testing results for the system components, and of the whole system, all using the voting step:

- Using only the 3D circle detector, the recognition rate was 99.1%, a false alarm rate of 2.1 per volume and a running time of 3.3 seconds/volume.
- Using only the short tube detector, with the same radius at both ends, the detection rate was 92.2%, the false alarm rate was 0.28/volume and the running time was 4.1 seconds/volume.
- Using only the "pure" short tube detector (described in section 2.5) the detection rate was 95.0%, the false alarm rate was 0.20/volume and the running time was 4.0 seconds/volume.
- Using only the "pure" short tube detector, but searching for radius differences of up to 3, the detection rate was 97.5%, the false alarm rate was 0.99/volume and the running time was 5.5 seconds/volume.
- Finally, by using the whole system, the detection rate was 94.7%, the false alarm rate was 0.12/volume and the running time was 5.3 seconds/volume. Moreover, none of the 26 false alarms from the 210 datasets was a polyp.



**Fig. 7.** Some segmentation results on unseen data

## 4 Discussion

In this paper, we presented a hierarchical, part-based method for segmenting 3D tubes, for example rectal tubes from CT data. The method is learning based, provides great generalization power, and is part-based for better computational efficiency.

## References

1. Bernstein, E. and Amit, Y., Part-based statistical models for object classification and detection. *CVPR* 2005
2. Iordanescu G, Summers RM. Reduction of false positives on the rectal tube in computer-aided detection for CT colonography. *Med Phys.* 2004; 31(10):2855-62.
3. Schapire R.E. The boosting approach to machine learning: An overview. *Nonlinear Estimation and Classification* D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, B. Yu, editors, Springer, 2003
4. Suzuki K., Yoshida H., Nappi J. J., Armato S. G. and Dachman A. H. False-positive Reduction in Computer-aided Detection of Polyps in CT Colonography Based on Multiple Massive Training Artificial Neural Networks. *RSNA* 2005
5. Tu Z., Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering, *ICCV*, 2005.