# Adaptive Branch Tracing and Image Sharpening for Airway Tree Extraction in 3-D Chest CT

Marco Feuerstein[1], Takayuki Kitasaka[2,3], Kensaku Mori[1,3]

[1] Graduate School of Information Science, Nagoya University, Japan,
fmarco@suenaga.m.is.nagoya-u.ac.jp,
[2] Faculty of Information Science, Aichi Institute of Technology, Japan,
[3] MEXT Innovation Center for Preventive Medical Engineering, Nagoya University, Japan.

**Abstract.** The automated extraction of the airway tree from 3-D chest CT data can greatly reduce the workload of physicians during diagnosis (e.g. quantification of airway morphology) and treatment (computer-aided bronchoscopy) of lung disease. This paper presents a method to automatically extract the airways driven by a sharpening filter, which enhances the branch edges in the input image based on the Laplacian of Gaussian, and adaptive cuboidal volumes of interest that an adaptive region growing algorithm uses to trace the airway tree. The method was trained on 20 data sets and evaluated on another 20 data sets from various scanners, using a wide range of acquisition and reconstruction parameters, including low dose scans. Compared to other state-of-the-art methods, our algorithm features the highest detection and extraction rates of bronchial branches. Future research needs to focus on the development of a method for automatic leakage detection.

## 1 Introduction

During the last century, from the first X-ray images taken 1895 and the advent of 3-D computed tomography (CT) in the seventies, medical imaging technologies greatly advanced. This went along with the generation of high-resolution data sets allowing more accurate as well as reliable diagnosis and treatment of diseases. However, as physicians diagnosing a patient need to handle this vast amount of data in a timely manner, various researchers and companies also started the development of methods for automated processing and analysis of patient data to allow a faster computer-aided diagnosis.

One example is the automatic extraction of the tracheobronchial (airway) tree from chest CT data to reduce the workload of physicians during diagnosis (e.g. quantification of airway morphology) and treatment (computer-aided bronchoscopy) of lung disease. This is a very active field of research, which started to evolve in about 1995 [1]. Sluimer et al. provide a short overview of methods developed during the first 10 years of research [2]. As all methods have to face problems highly related to image quality and acquisition such as low-dose scanning, noise, partial volume effects, and cardiac motion, and patient-specific

aspects such as pathology, implants, and mucus, the perfect method is yet to be developed. Hence, many more recent works can be found in the literature addressing one or more of these problems, e.g. [3–6].

We here present an enhancement of our previously developed adaptive airway tree extraction method [7, 8], which can work on noisy and low-dose scans, is faster than our initial method, and was evaluated on a large number of datasets.

## 2 Method

As our method requires input data with reasonably low noise and a seed point in the trachea to start the airway extraction, we are first performing a couple of preprocessing steps, as described in section 2.1. Section 2.2 describes the core of our algorithm. Basically, it first reconstructs a small volume of interest (VOI) around the current branch of the airway tree (starting at the trachea), which is automatically adapted to the size of the branch. Within the VOI, first the edges of the airways are enhanced by a sharpening filter, followed by region growing using an adaptive threshold to take care of obvious leakages and a routine to detect bifurcations or trifurcations of the tree. Each bifurcation or trifurcation is followed by generating more volumes of interest and repeating the process, until no more furcations can be detected.

### 2.1 Preprocessing

*Seed Point Search* As our algorithm depends on a reliable seed point inside the trachea, we first determine air voxels within the body region of the patient, whose intensities are lower than about -500 Hounsfield units (HU). Based on a method similar to [9], we then extract the lung and its bounding box. Within the most superior axial slice of the bounding box, we automatically determine the trachea, which is of circular shape, specific size (i.e. a diameter of less than about 5 cm), approximately in the center of the bounding box along the mediolateral axis, and in between the center and the anterior of the bounding box along the anteroposterior axis. It is important to use this particular slice to start the search and not another more superior slice e.g. in the oral cavity, as the air region there is not shaped as the trachea and sometimes disconnected from the rest of the airways. From all trachea voxels extracted in the axial slice we simply choose their centroid as our seed point for airway extraction.

*Image Smoothing* Our algorithm also requires image data that contains a reasonably low amount of noise, which usually does not hold for certain scan types (e.g. low dose scans) and reconstruction parameters (e.g. very sharp convolution kernels). We therefore analyze image noise by computing the mean of the gradient magnitude of all air voxels identified previously inside the bounding box of the lung. Depending on the level of image noise, we perform $n$ image smoothing iterations. During smoothing, we utilize a modified curvature diffusion equation [10], which is a level-set equivalent of the anisotropic diffusion equation proposed

in [11]. However, it preserves and enhances airway tree edges better than conventional anisotropic diffusion. Using default parameters, i.e. a time step of 0.0625 in the computation of the level set evolution and a value of conductance of 3.0, we alter the number of iterations $n$ depending on the mean gradient magnitude $\overline{\nabla}$:

$$n = \begin{cases} 0 & \text{if } \overline{\nabla} \leq 100 \\ 1 & \text{if } 100 < \overline{\nabla} \leq 150 \\ 2 & \text{if } 150 < \overline{\nabla} \leq 200 \\ 3 & \text{otherwise} \end{cases} \tag{1}$$

### 2.2 Airway Tree Extraction

An airway tree is extracted by tracing its structure branch by branch using VOIs of adaptive size and starting at the trachea. We here explain the procedures briefly. More details can be found in [7, 8, 12].

*Initial Threshold Selection* First, we obtain an initial threshold for airway tree extraction, $T_{ini}$, by performing a region growing based method [1], which extracts the largest region whose number of voxels is smaller than a predefined value (e.g. one tenth of the lung volume) starting at our previously determined seed point.

*VOI Setting* We define a VOI as a cuboid with eight vertices $\boldsymbol{V}_1$ - $\boldsymbol{V}_8$ and two points $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, which are the centers of the two faces $S_1$ and $S_2$ (see Fig. 1). The size of a VOI is denoted by the width $W$, height $H$, and depth $D$. The z-axis of a VOI indicates the running direction of a bronchial branch. Hereafter, the image circumscribed by a VOI is used to calculate local features of the bronchus and is called *VOI image*.

*Local Image Enhancement* Whenever a new VOI image is generated, it is first enhanced by performing a sharpening filter based on the Laplacian of Gaussian [7]. Using the sharpening filter, the intensity of the bronchial wall becomes higher while that of the lumen is weakened. This is to reinforce bronchial features that were lost due to partial volume effects.

In detail, a VOI image $\boldsymbol{F}$ is enhanced to $\boldsymbol{F}_{\text{sharpened}}$ by

$$\boldsymbol{F}_{\text{sharpened}} = \boldsymbol{F} - \beta \cdot \frac{LoG(\boldsymbol{F}) + L'oG(\boldsymbol{F})}{2} \tag{2}$$

where $LoG$ is the Laplacian of Gaussian, $L'oG$ is a modified version of it, and $\beta$ determines the degree of sharpening. The $L'oG$ filter is using the same Gaussian convolution kernel as the $LoG$ filter, but its Laplacian convolution kernel ignores any voxels being greater than the center voxel of the kernel.

The standard Laplacian filter gives negative values for voxels whose neighbors are of lower intensities and positive values for neighbors of higher intensities. This is, if airway branches are clearly separable from surrounding tissue, bronchial

wall voxels will tend to give negative values, while bronchial lumen voxels will tend to give positive values, and thus airway branches are enhanced. However, if the bronchial wall is adjacent to tissue of higher intensity than the wall itself, the standard *LoG* filter can provoke the contrary result, weakening the wall. To counteract this effect, we introduce the modified Laplacian of Gaussian, whose output is always negative and therefore never weakens the bronchial wall. We equally weight *LoG* and $L'oG$.

In our current implementation, we set the scale $\sigma_L$ of the Laplacian kernel to

$$\sigma_L = \min\left(3, \lfloor 15 \cdot reso_{\min}/r \rfloor\right) \tag{3}$$

and the scale $\sigma_G$ of the Gaussian kernel to

$$\sigma_G = reso_{\min} \tag{4}$$

where $reso_{\min}$ is the minimum voxel resolution of the VOI image and $r$ is the radius of the currently processed branch (see below for details on how to obtain $r$). However, in the future we will unify the sizes of the convolution kernels, so they are both dependent on the radius of the current branch.

*VOI for Trachea* We set the VOI for the trachea, which is the root of the airway tree, based on the seed point detected above. First, the orientation of the coordinate system of the VOI is set to be equal to that of the CT image. The center point $\boldsymbol{P}_1$ of the VOI is translated so as to coincide with the seed point. The VOI is of cuboidal shape and its edges are twice as long as the diameter of the trachea region intersecting with the most superior axial slice of the bounding box of the lung. The bronchial region inside the VOI image is then extracted by a region growing with threshold $T$, which is initialized by $T_{ini} + 100$[HU].

Then, the VOI is iteratively adjusted in z direction using a binary search [12]. This is, the VOI is extended or shortened while halving the search interval of the binary search, until we find a furcation or the end of a branch (see below).

*Identification of Furcation* After each iteration, we check the number of connected components, $N_c$, on the VOI surface (except $S_1$) to find out whether the current branch furcates or not. Each connected component is denoted by $C_i$ $(1 \leq i \leq N_c)$. Depending on the number of connected components, we iteratively execute one of the following cases:

- $\mathbf{N_c = 0}$: If the binary search interval is still greater than $reso_{\min}$, we shorten the VOI. If it reaches $reso_{\min}$, we terminate tracing.
- $\mathbf{N_c = 1}$: The VOI is extended and the bronchial region is re-segmented using region growing thresholded at $T$.
- $\mathbf{N_c = 2}$ or $\mathbf{N_c = 3}$: We found a furcation. If the binary search interval is still greater than $reso_{\min}$, we shorten the VOI. If it is equal to $reso_{\min}$, we terminate tracing for the current branch and calculate its final dimensions. Its thickness is measured by averaging the extracted bronchial region on each slice of the VOI image along the z-axis of the VOI. The radius of the

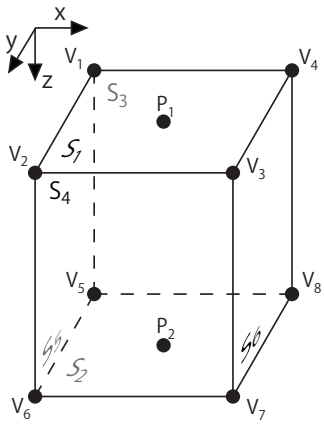



Fig. 1: Definition of VOI. It has eight vertices $\boldsymbol{V}_1$ - $\boldsymbol{V}_8$ and two points $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ that are the centers of the faces $S_1$ and $S_2$. The VOI size is $W \times H \times D$.

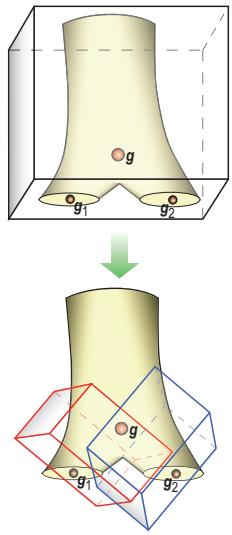



Fig. 2: Generation of child VOIs. Each child VOI is oriented along the line from a detected furcation point $\boldsymbol{g}$ to the center of gravity $\boldsymbol{g}_i$ of child branches. Width and height of a child VOI are set to be twice as large as the radius of the child branch.

branch, $r$, is calculated from this thickness. The furcation point, $\boldsymbol{g}$, is then determined as the gravity center of the bronchial region on the (D-r)-*th* slice. The gravity center of each $C_i$, $\boldsymbol{g}_i$, is also calculated. Then, child VOIs are generated based on $\boldsymbol{g}$, $\boldsymbol{g}_i$, and $r$, as described in the next paragraph.

– $\mathbf{N_c} > \mathbf{3}$: If the binary search interval is still greater than $reso_{\min}$, we shorten the VOI. If it is equal to $reso_{\min}$, the threshold T is reduced by $\Delta T$ and the bronchial region is updated.

*Generation of Child VOIs* If a bifurcation or trifurcation appears, we generate child VOIs for each $C_i$ to further trace the airway branches (Fig. 2). $P_1$ of a child VOI for $C_i$ is set to the furcation point $\boldsymbol{g}$, while $P_2$ is set to $\boldsymbol{g}_i$. That is, the VOI is oriented from the furcation point to the child branch. Width and height of a child VOI are set to be twice as big as the radius of the child branch, which is calculated from the region $C_i$. To prevent tracing from growing into sibling branches, we set levees $L_j$ among VOIs as shown in Fig. 3. No voxels beyond these levees are extracted. Then, the bronchial region in a child VOI is updated. If the radius of a branch is smaller than a predefined value $T_r$, then the resolution of the current VOI image is doubled by using tricubic interpolation. The bronchial

region is updated again and we repeat the iterative process described before to check whether a furcation appears.

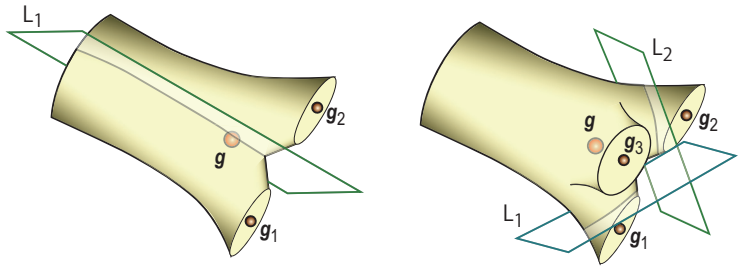



Fig. 3: Levees to prevent tracing from growing into sibling branches. They are set among newly created child VOIs for the cases of (left) bifurcation and (right) trifurcation, respectively.

*Reconstruction of Airway Tree* After finishing all processes, we reconstruct the airway tree by projecting bronchial regions inside all VOI images into the original CT image. In addition, the tree structure is extracted by connecting $P_1$ and $\boldsymbol{g}$ of each VOI.

## 3 Results

Our method was centrally evaluated by a team of trained observers within the scope of an airway tree segmentation challenge to compare the performance of different algorithms [13]. We applied our method to 40 CT data sets acquired by various scanners located in hospitals in Denmark, Germany, the Netherlands, USA, and Japan, using a wide range of acquisition and reconstruction parameters, including low dose scans. We used 20 data sets to train our method and 20 data sets for evaluation. The free parameters of our algorithm, as described in Section 2, were selected as $\Delta T = 4$, $T_r = 15 \cdot reso_{\min}$, and $\beta = 0.05$, using the training data.

To obtain correctly (true positive) and falsely (false positive) extracted airway tree voxels and branches and in order to compare our method to other algorithms, ground truth segmentations were generated from the extraction results of all algorithms. Therefore, for each algorithm and data set, following evaluation steps were performed:

1. Determination of all airway tree branches from the segmentation using a fast marching based algorithm similar to [14].
2. Visual evaluation of the airway tree branches using MPR (multiplanar reformation) and oblique views as well as reformatted views with straightened airway centerlines.

3. Labeling of each branch to be true positive or false positive, i.e. whether it is part of the airway tree or not, by at least two observers; its exact shape and dimensions are ignored.

The overall ground truth segmentation for each data set is then generated from all true positive branches of all algorithms. For evaluation, following seven measures were defined and computed:

- *Branch count:* Number of true positive branches with a centerline length of more than 1 mm.
- *Branch detected:* Percentage of the number of true positive branches divided by the number of ground truth branches.
- *Tree length:* Overall length of the centerlines of all true positive branches.
- *Tree length detected:* Percentage of the true positive tree length divided by the ground truth tree length.
- *Leakage count:* Number of unconnected true positive regions in a 26-connected neighborhood of all false positive voxels, i.e. within a one voxel-thick border around all false positive voxels. This measure indicates how difficult it is to manually separate leakages from true positive branches.
- *Leakage volume:* The volume of all false positive voxels.
- *False positive rate:* Percentage of the number of false positive voxels divided by the number of all detected voxels.

During evaluation, the trachea was excluded from any measures related to branch and tree length or count. For the leakage measures, the trachea as well as the left and right main bronchi were excluded.

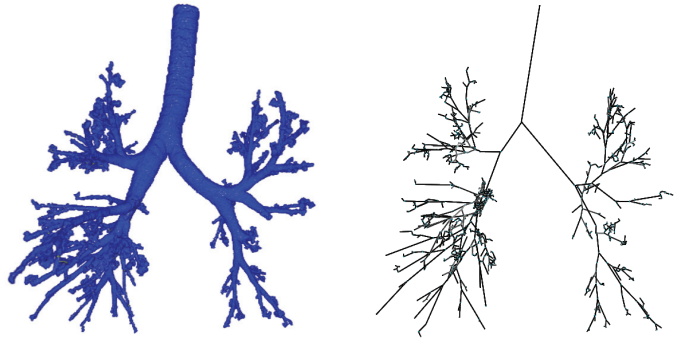
Table 1 summarizes the results of the evaluation of our method. Exemplary airway tree extractions depicting extreme cases of leakages and branch extractions are shown on Fig. 4. Depending on the size of the data set and the number of branches to extract, our method takes about $5 \pm 3$ minutes to process a case on a standard workstation with two 64-bit Quad-Core Intel Xeon 5355 processors.

## 4 Discussion

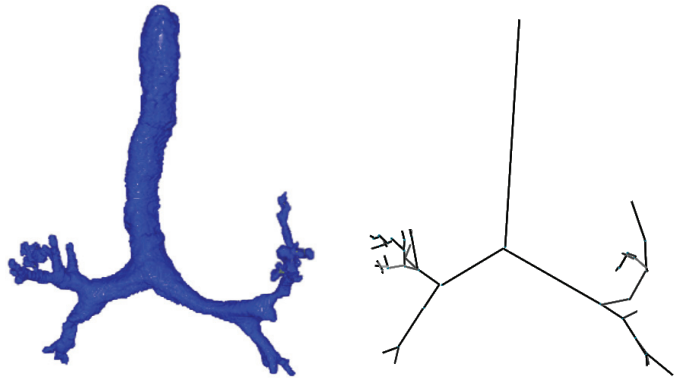
As can be seen in Tables 1 and 2, our method can extract the airway tree more completely than any method of the other teams. However, it is also obvious that our method currently produces a lot of leakages. All leakages occur in peripheral branches, where the bronchus walls are very thin and hence allow their interior to easily merge with the exterior (due to partial volume effects and insufficient image resolution). The lacking availability of a more appropriate stopping criterion or leakage detection of our algorithm attributes to the high number of false positive branches. Various solutions for this could be thought of. One reasonable approach would be to integrate simultaneous labeling of branches into our algorithm (similar to [14]), which could verify the currently extracted tree structure against an atlas or a database of ground truth trees and automatically detect branches that are heavily deviating from the ground truth data,

Table 1: Evaluation results for the twenty cases in the test set.

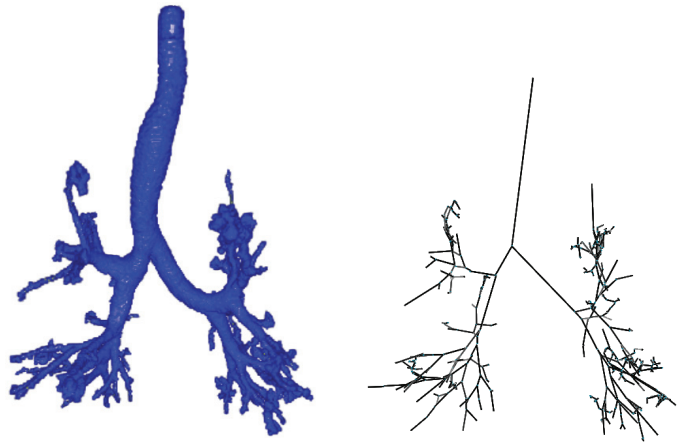
| | Branch count | Branch detected (%) | Tree length (cm) | Tree length detected (%) | Leakage count | Leakage volume ($mm^3$) | False positive rate (%) |
|---|---|---|---|---|---|---|---|
| CASE21 | 104 | 52.3 | 58.4 | 52.8 | 9 | 1138.8 | 10.43 |
| CASE22 | 292 | 75.5 | 233.6 | 70.7 | 103 | 7558.5 | 20.37 |
| CASE23 | 242 | 85.2 | 198.4 | 76.3 | 92 | 5387.7 | 16.83 |
| CASE24 | 164 | 88.2 | 140.3 | 86.3 | 50 | 6690.9 | 18.04 |
| CASE25 | 210 | 89.7 | 226.3 | 89.8 | 33 | 4944.8 | 11.99 |
| CASE26 | 42 | 52.5 | 30.6 | 46.5 | 2 | 92.6 | 1.52 |
| CASE27 | 72 | 71.3 | 48.9 | 60.3 | 2 | 63.2 | 0.61 |
| CASE28 | 116 | 94.3 | 97.4 | 88.8 | 13 | 2011.7 | 12.68 |
| CASE29 | 137 | 74.5 | 99.8 | 72.3 | 21 | 4417.9 | 22.42 |
| CASE30 | 157 | 80.5 | 117.6 | 77.0 | 24 | 10404.7 | 35.88 |
| CASE31 | 187 | 87.4 | 158.7 | 90.4 | 30 | 11956.8 | 30.90 |
| CASE32 | 181 | 77.7 | 166.5 | 76.4 | 34 | 8452.1 | 21.75 |
| CASE33 | 138 | 82.1 | 111.7 | 75.9 | 30 | 1722.3 | 12.28 |
| CASE34 | 339 | 74.0 | 248.7 | 69.5 | 61 | 5287.9 | 11.47 |
| CASE35 | 301 | 87.5 | 270.2 | 87.3 | 70 | 7188.3 | 17.34 |
| CASE36 | 265 | 72.8 | 274.6 | 66.6 | 7 | 809.2 | 2.43 |
| CASE37 | 170 | 91.9 | 155.4 | 87.4 | 42 | 7966.4 | 24.28 |
| CASE38 | 66 | 67.3 | 43.9 | 66.1 | 5 | 255.3 | 2.24 |
| CASE39 | 248 | 47.7 | 202.6 | 49.5 | 34 | 5817.6 | 16.76 |
| CASE40 | 305 | 78.4 | 290.9 | 75.2 | 47 | 10598.2 | 21.08 |
| Mean | 186.8 | 76.5 | 158.7 | 73.3 | 35.5 | 5138.2 | 15.56 |
| Std. dev. | 86.4 | 13.3 | 82.2 | 13.4 | 28.6 | 3754.5 | 9.52 |
| Min | 42 | 47.7 | 30.6 | 46.5 | 2 | 63.2 | 0.61 |
| 1st quartile | 116 | 71.3 | 97.4 | 66.1 | 9 | 1138.8 | 10.43 |
| Median | 176 | 78.0 | 157.0 | 75.6 | 32 | 5337.8 | 16.79 |
| 3rd quartile | 292 | 88.2 | 248.7 | 87.4 | 61 | 8452.1 | 22.42 |
| Max | 339 | 94.3 | 290.9 | 90.4 | 103 | 11956.8 | 35.88 |



(a) CASE22 (Highest number of leakages)

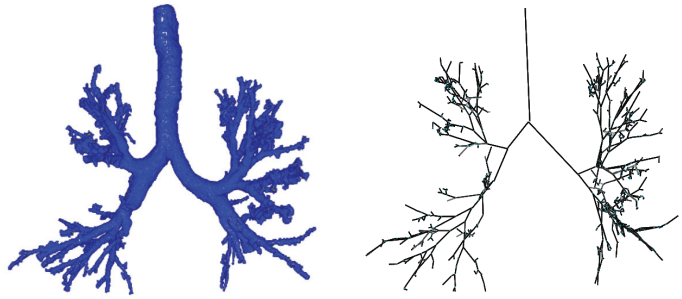


(b) CASE26 (Lowest number of detected branches)

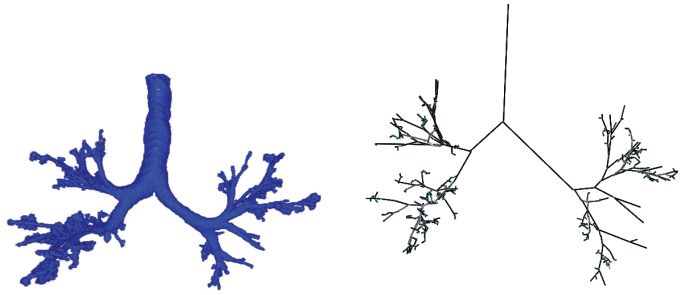


(c) CASE31 (Highest leakage volume)

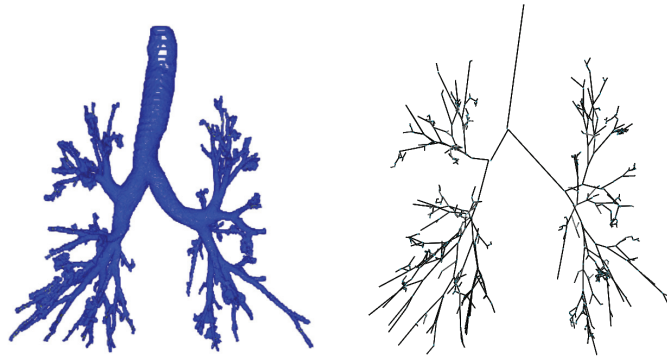
Fig. 4: Exemplary results visualized by (left) volume rendering of the extracted airways and (right) the corresponding tree structure.



(d) CASE34 (Highest number of detected branches)



(e) CASE37 (Highest percentage of detected branches)



(f) CASE40 (Longest tree length)

Fig. 4: Exemplary results visualized by (left) volume rendering of the extracted airways and (right) the corresponding tree structure.

e.g. in terms of branch length, branch direction, or furcation patterns. Another more pragmatic solution would be to check for extracted regions not looking like branches, i.e. having a much bigger radius than their parent branches or not being of tubular shape. This could be performed in a post-processing step.

Table 2: Comparison of our results to all other teams participating in the challenge. Teams using semi-automated segmentation methods are marked by a *.

| | Branch count | Branch detected (%) | Tree length (cm) | Tree length detected (%) | Leakage count | Leakage volume ($mm^3$) | False positive rate (%) |
|---|---|---|---|---|---|---|---|
| Team 1 | 91.1 | 43.5 | 64.6 | 36.4 | 2.5 | 152.3 | 1.27 |
| Team 2 | 157.8 | 62.8 | 122.4 | 55.9 | 12.0 | 563.5 | 1.96 |
| *Team 3 | 74.2 | 32.1 | 51.9 | 26.9 | 4.2 | 430.4 | 3.63 |
| (Ours) Team 4 | 186.8 | 76.5 | 158.7 | 73.3 | 35.5 | 5138.2 | 15.56 |
| Team 5 | 150.4 | 59.8 | 118.4 | 54.0 | 1.9 | 18.2 | 0.11 |
| *Team 6 | 77.5 | 36.7 | 54.4 | 31.3 | 2.3 | 116.3 | 0.92 |
| Team 7 | 146.8 | 57.9 | 125.2 | 55.2 | 6.5 | 576.6 | 2.44 |
| *Team 8 | 71.5 | 30.9 | 52.0 | 26.9 | 0.9 | 126.8 | 1.75 |
| Team 9 | 139.0 | 56.0 | 100.6 | 47.1 | 13.5 | 368.9 | 1.58 |
| Team 10 | 79.3 | 32.4 | 57.8 | 28.1 | 0.4 | 14.3 | 0.11 |
| *Team 11 | 93.5 | 41.7 | 65.7 | 34.5 | 1.9 | 39.2 | 0.41 |
| Team 12 | 130.1 | 53.8 | 95.8 | 46.6 | 5.6 | 559.0 | 2.47 |
| Team 13 | 152.1 | 63.0 | 122.4 | 58.4 | 5.0 | 372.4 | 1.44 |
| Team 14 | 161.4 | 67.2 | 115.4 | 57.0 | 44.1 | 1873.4 | 7.27 |
| *Team 15 | 148.7 | 63.1 | 119.2 | 58.9 | 10.4 | 158.8 | 1.19 |

Furthermore, it is worth mentioning that our algorithm as well as any other algorithms based on region growing, level sets, fast marching, and the like starting from a single seed point inside the trachea will fail to extract the airways, if any of the branches are interrupted by tumors. This issue needs to be tackled by the research community in their upcoming work.

## 5 Conclusion

We presented a fully automatic method for airway tree extraction in 3-D chest CT. After automatically finding a seed point in the trachea, it tracks and extracts the airway branches within adaptive cuboidal volumes of interest and by enhancing the input image utilizing a sharpening filter based on the Laplacian of Gaussian. As our current implementation does not include an automatic leakage detection method, we get false positive branches in the peripheral branches. Apart from that, our method is able to extract a very high number of airway branches, in fact more than any of the other 14 semi-automated and automatic methods evaluated within an airway extraction challenge.

## Acknowledgments

The authors thank their colleagues for many helpful suggestions and advices. Parts of this research were supported by the Japan Society for the Promotion of Science (JSPS) postdoctoral fellowship program for foreign researchers, a Grant-In-Aid for Scientific Research from JSPS, the program of formation of innovation center for fusion of advanced technologies "Establishment of early preventing medical treatment based on medical-engineering for analysis and diagnosis" funded by the Ministry of Education (MEXT), and a Grant-In-Aid for Cancer Research from the Ministry of Health, Labour and Welfare.

## References

1. Mori, K., Hasegawa, J., Toriwaki, J., Anno, H., Katada, K.: Automated extraction and visualization of bronchus from 3D CT images of lung. In: Proceedings of Computer Vision, Virtual Reality and Robotics in Medicine. (1995)
2. Sluimer, I., Schilham, A., Prokop, M., van Ginneken, B.: Computer analysis of computed tomography scans of the lung: A survey. IEEE Transactions on Medical Imaging **25**(4) (2006) 385–405
3. Tschirren, J., Hoffman, E., McLennan, G., Sonka, M.: Intrathoracic airway trees: Segmentation and airway morphology analysis from low-dose CT scans. IEEE Transactions on Medical Imaging **24**(12) (2005) 1529–1539
4. Graham, M.W., Gibbs, J.D., Higgins, W.E.: Robust system for human airway-tree segmentation. In: SPIE Medical Imaging. (2008)
5. Lo, P., de Bruijne, M.: Voxel classification based airway tree segmentation. In: SPIE Medical Imaging. (2008)
6. van Ginneken, B., Baggerman, W., van Rikxoort, E.M.: Robust segmentation and anatomical labeling of the airway tree from thoracic CT scans. In: MICCAI. (2008)
7. Kitasaka, T., Mori, K., Hasegawa, J., Toriwaki, J.: A method for extraction of bronchus regions from 3D chest X-ray CT images by analyzing structural features of the bronchus. FORMA **17**(4) (2002) 321–338
8. Kitasaka, T., Mori, K., Suenaga, Y., Hasegawa, J., Toriwaki, J.: A method for segmenting bronchial trees from 3D chest X-ray CT images. In: MICCAI. (2003)
9. Hu, S., Hoffman, E.A., Reinhardt, J.M.: Automatic lung segmentation for accurate quantitation of volumetric X-ray CT images. IEEE Transactions on Medical Imaging **20**(6) (2001) 490–498
10. Whitaker, R., Xue, X.: Variable-conductance, level-set curvature for image denoising. In: International Conference on Image Processing. Volume 3. (2001) 142–145
11. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**(7) (1990) 629–639
12. Mori, K., Kitasaka, T., Akimoto, S., Ebe, K., Wada, T.: Endoscope insertion support system and endoscope insertion support method (2009)
13. Lo, P., van Ginneken, B., Reinhardt, J., de Bruijne, M.: Extraction of airways from CT (EXACT'09). In: Second International Workshop on Pulmonary Image Analysis. (2009)
14. Schlathölter, T., Lorenz, C., Carlsen, I.C., Renisch, S., Deschamps, T.: Simultaneous segmentation and tree reconstruction of the airways for virtual bronchoscopy. In: SPIE Medical Imaging. (2002)