

Airway Segmentation Framework for Clinical Environments

Juerg Tschirren¹, Tarunashree Yavarna², and Joseph M. Reinhardt²

¹ VIDA Diagnostics, Inc.
100 Oakdale Campus, 225 TIC
Iowa City, IA 52242, USA,
juerg@vidadiagnostics.com,
<http://www.vidadiagnostics.com>

² The University of Iowa, Dept. of Biomedical Engineering,
Iowa City, IA 52242, USA

Abstract. A segmentation framework for the identification of human airway trees in high resolution computed tomography (CT) images is presented. This framework consists of a fully automated segmentation algorithm, supplemented by software editing tools that allow the user to correct the airway segmentation result if needed. The algorithm and tools presented in this paper have been successfully applied on more than 10,000 CT scans.

1 Introduction

The airway segmentation algorithm and manual airway editing tools presented here are part of VIDA Diagnostic's "Pulmonary Workstation 2.0" (PW2), a commercial software suite for the analysis of human lung CT scans. The software requirements for a commercial product such as PW2 differ substantially from those used in academic settings. The primary goal is to run a segmentation task in as little time as possible, and have it return an acceptable result in the majority of cases. An algorithm that finished in a few seconds and returns a usable result is strongly preferred over an algorithm with a run time of several minutes or longer, even if this slower algorithm returns significantly better results.

Beyond the automated methods it is important to have manual editing tools available to correct results as needed. No algorithm can guarantee 100% perfect results in all possible cases and the user needs to be able to correct segmentation results as needed, using efficient and intuitive tools.

In this paper we present an airway segmentation framework that consists of two major parts: 1) a fully automated airway segmentation algorithm, and 2) a set of software tools that allow the user to edit the airway segmentation result if needed.

The aim of the airway segmentation algorithm presented here is to identify the whereabouts of the airway lumen. The algorithm presented here does *not*

identify the exact location of the airway wall (as for example used to determine exact airway diameters). Exact inner and outer airway wall localization is done by the airway measurement algorithm, which we described elsewhere as it lies outside the scope of this paper.

2 Methods

2.1 Automated Airway Segmentation

The automated airway segmentation algorithm is split into two parts: 1) finding a seed point within the trachea, and 2) growing the airway tree. They are described separately below.

Trachea Finder The trachea seed point required to grow the airway tree is found using the following steps:

1. *Threshold axial slices.* Every axial slice s_a in the slice range $[15, \frac{N_{\text{axial}}}{2}]$ of the CT volume is transformed into a thresholded axial image \mathcal{T}_a using

$$\mathcal{T}_a(x, y) = \begin{cases} 255 & \text{if } |F(x, y)| > |B(x, y)| \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

with the foreground and background pixel sets defined as

$$\begin{aligned} F(x, y) &= \{(x', y') : \forall (x', y') \in N_9(x, y), d(s_a(x', y')) \leq t\} \\ B(x, y) &= \{(x', y') : \forall (x', y') \in N_9(x, y), d(s_a(x', y')) > t\}, \end{aligned} \quad (2)$$

respectively. $N_9(x, y)$ represents the 3×3 neighborhood at (x, y) , inclusive (x, y) , and $d(\cdot)$ returns the CT density of a voxel in s_a , measured in Hounsfield Units (HU).

2. *Find & classify foreground objects.* In every thresholded slice t_a a connected component analysis [8] is performed and the border polygon of every foreground object is found using the *boundary tracing* algorithm described in [8, p. 142]. The compactness $\sigma \in [0, 1]$ is computed for every object with

$$\sigma = \frac{4\pi A}{c^2} \quad (3)$$

with $\sigma = 1$ in case of a perfectly circular object, and the value of σ getting smaller the less circular the object. The area A is computed based on the number of object pixels and the circumference c is computed by adding up the lengths of the segments along the border polygon.

Only objects with $\sigma > 0.2$ and $40 \text{ mm}^2 < A < 500 \text{ mm}^2$ are used for further processing.

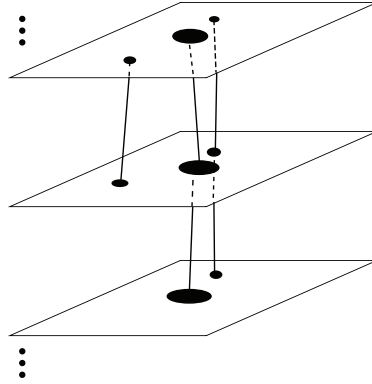


Fig. 1. Chains of foreground objects between individual axial slices are built by connecting elements that are close enough and similar enough to each other.

3. *Build foreground object chains.* The similarity ς between two foreground objects f_1 and f_2 is defined as

$$\varsigma(f_1, f_2) = 1 - \frac{d(f_1, f_2)}{r_{\text{circ } f_1}} \quad (4)$$

with the Euclidean distance d [voxels] between centroids

$$d(f_1, f_2) = \sqrt{(x_{\text{centroid } f_1} - x_{\text{centroid } f_2})^2 + (y_{\text{centroid } f_1} - y_{\text{centroid } f_2})^2} \quad (5)$$

and the circle-equivalent radius

$$r_{\text{circ } f_1} = \sqrt{\frac{A_{f_1}}{\pi}} \quad (6)$$

Only object pairs with $d(f_1, f_2) \leq 2$ are considered.

Given the set \mathcal{F} all foreground objects, find the object pairs (f_1, f_2) with the highest similarity and arrange them into one or more object chain(s) c (Figure 1), stored in the set \mathcal{C} ($c \in \mathcal{C}$) with

```

for all  $f_1 \in \mathcal{F}$  do
   $s_{\text{max}} \leftarrow 0$ 
   $f_{\text{max}} \leftarrow \text{null}$ 
  for all  $f_2 \in \mathcal{F}, f_1 \neq f_2$  do
     $s \leftarrow \varsigma(f_1, f_2)$ 
    if  $s > s_{\text{max}}$  then
       $s_{\text{max}} \leftarrow s$ 
       $f_{\text{max}} \leftarrow f_2$ 
    end if
  end for
  if  $s_{\text{max}} > 0.6$  then
     $\mathcal{C} = \mathcal{C} \cup (f_1, f_{\text{max}})$ 
  
```

end if
end for

4. *Find trachea chain.* For each $c \in \mathcal{C}$, compute the average radius r_{avg} with

$$r_{\text{avg}} = \frac{1}{N} \sum_{i=0}^{N-1} r_i \quad (7)$$

where N is the number of foreground objects in c , and the average variation from the volume center along the x-axis, x_{var} , defined as

$$x_{\text{var}} = \frac{1}{N} \sum_{i=0}^{N-1} \left| c_x - \frac{X}{2} \right| \quad (8)$$

where X is the volume size along the x -axis (number of voxels).

If r_{avg} and x_{var} apply to the same chain, label this chain as trachea. Otherwise compute the mean y value for all centroids along the chain

$$\bar{y}_{\text{centroid}} = \frac{1}{N} \sum_{i=0}^{N-1} c_y \quad (9)$$

for each of these two chains and label the one chain with the smaller $\bar{y}_{\text{centroid}}$ value as trachea. This is to prevent the esophagus from being labeled as trachea (assuming scan orientation such that the y -axis increases from anterior to posterior).

Growing Airway Tree Many different airway segmentation algorithms have been presented in the past. [1, 3, 10, 5, 2, 4] represent just a few of them. The core algorithm for the automatic airway segmentation is similar to the one presented by Mori et al. [1]. A queue-based breadth-first flood fill algorithm [6] is used to grow the airway lumen (Algorithm 1). This algorithm is executed iteratively by Algorithm 2, continuously increasing the threshold value for the region grow. At each iteration the number of voxels grown by the flood fill algorithm is recorded. A sudden big increase between two consecutive iteration steps n and $n + 1$ is considered a *leak*. We use the term “leak” for the local effect where grown airway lumen “mushrooms” into the surrounding lung parenchyma. This may happen due to the similar x-ray density values of lung tissue, compared with the airway lumen, and the relatively thin airway wall, particularly in small peripheral airways. Figure 2 shows an un-edited segmentation of an airway tree.

2.2 Manual Editing Tools

Deleting Airway Leaks Airway leaks can easily be removed with a few mouse clicks. First the airway tree is skeletonized using sequential 3D thinning as described in [7]. This gives us information about the topology of the tree and allows us to identify the individual segments and make them selectable in a 3D surface

Algorithm 1 Flood fill algorithm.

Require: $seed$ {Coordinate of seed voxel}
Require: t {Threshold value}

```

initial-color  $\leftarrow$  color( $seed$ )
color( $seed$ )  $\leftarrow$  final-color
 $n \leftarrow 1$ 
enqueue( $Q, seed$ )
while  $Q$  not empty do
   $h \leftarrow$  head( $Q$ )
  dequeue( $Q$ )
  for each  $n \in$  neighbors( $h, 6$ -neighborhood) do
    if color( $n$ ) = initial-color and density( $n$ ) <  $t$  then
      color( $n$ )  $\leftarrow$  final-color
       $n \leftarrow n + 1$ 
      enqueue( $Q, n$ )
    end if
  end for
end while
return  $n$ 

```

Algorithm 2 Iterative Growing. The maximal grow rate is set to $g_{\max} = 1.6$ and the maximum voxel count is set to $n_{\text{voxels max}} = 500,000$

Require: $seed$ {Coordinate of seed voxel}

```

 $t \leftarrow$  density( $seed$ )
 $n_{\text{voxels}} \leftarrow$  flood_fill( $c_{\text{seed}}, t$ )
repeat
   $t \leftarrow t + 1$ 
   $n_{\text{voxels prev}} \leftarrow n_{\text{voxels}}$ 
   $n_{\text{voxels}} \leftarrow$  flood_fill( $c_{\text{seed}}, t$ )
   $g \leftarrow \frac{n_{\text{voxels}}}{n_{\text{voxels prev}}}$ 
   $i \leftarrow i + 1$ 
  erase current segmentation result
until  $g > g_{\max}$  or  $n_{\text{voxels}} > n_{\text{voxels max}}$ 
flood_fill( $c_{\text{seed}}, t - 1$ )

```

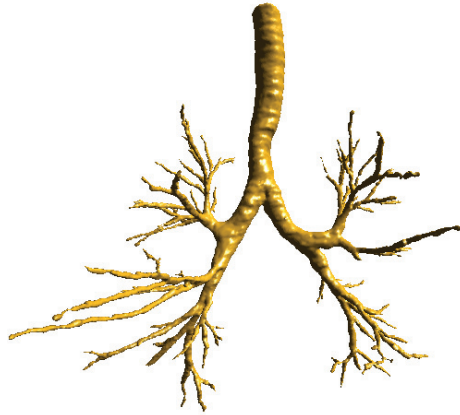
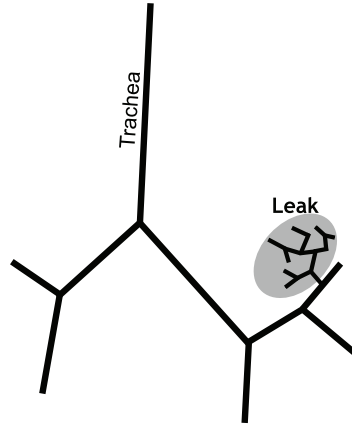


Fig. 2. Segmentation result of a CT scan at total lung capacity (TLC). 281 individual branches. No manual corrections. Segmentation time 17.5 s on an Intel Xeon™ CPU based machine running at a clock speed of 2.33 GHz. CT scan from VIDA's test database (not a subject from the EXACT09 contest database).

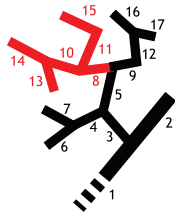
rendering. The user then selects an arbitrary segment within the leak and uses the mouse wheel to progress the selection up and down the tree as illustrated in the graph view in Figure 3. Turning the mouse wheel in one direction increases the number of selected segments, and turning the mouse wheel in the other direction de-selects the just selected segments. This allows the user to iteratively find the selection size that includes as many leak segments as possible without selecting actual airway segments. At any time the currently selected segments can be deleted by right-clicking on the selection. Figure 4 shows the actual progress in a sequence of screen shots taken in PW2.

Adding New Sub-Trees The user can add a missing sub-tree by placing a seed point inside the lumen, near the end of the current segmentation result. Figure 5(a) shows a screen shot from PW2 illustrating this. The user can select between three different threshold levels, which determines the growth rate. The computer then attempts to add the missing branch(es) starting at this seed point, using the algorithm described in Section 2.1 above.

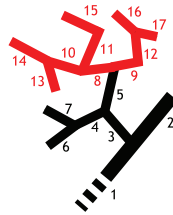
The region-grow algorithm alone can not always connect the newly added sub-tree back to the main tree. This is for example the case when a radio-dense object such as a mucus plug blocks the airway lumen. If connectivity is not automatically established we use Dijkstra's algorithm [9], starting from the user-place seed point, to find the shortest path back to the main tree. The cost function used favors low densities, and a penalty is added which progressively increases for longer paths.



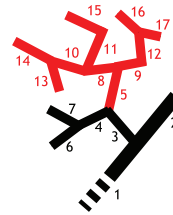
(a) Leak in left upper lobe.



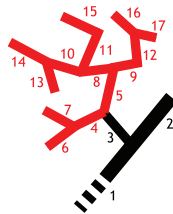
(b) User selects segment 8 (s8). All segments topologically underneath s8 are automatically selected.



(c) The first mouse-wheel increment expands the selection to the sibling of s8 (s9) and all segments topologically underneath it.



(d) The second mouse-wheel increment selects s8's parent segment s5.

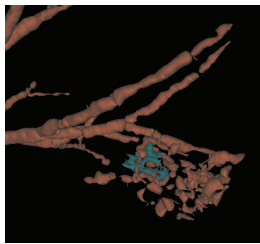


(e) The third mouse-wheel increment expands the selection to s5's sibling and the segments topologically underneath it.

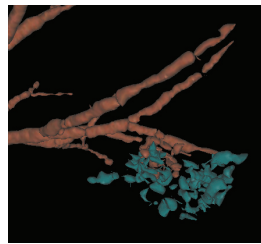


(f) After fourth and last mouse-wheel increment, s5's parent segment s3 is selected.

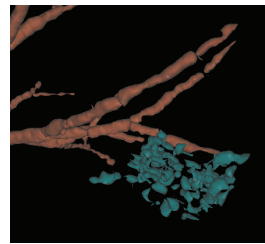
Fig. 3. Leak selection in graph view. (a): overall view. (b)–(f): detail view of leak with red/gray representing individual user selection steps. See main text for detailed explanation.



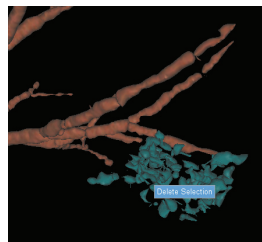
(a) User selected an arbitrary segment within a leak.



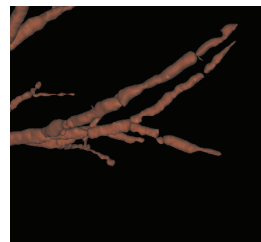
(b) After first mouse-wheel increment.



(c) After second mouse-wheel increment.

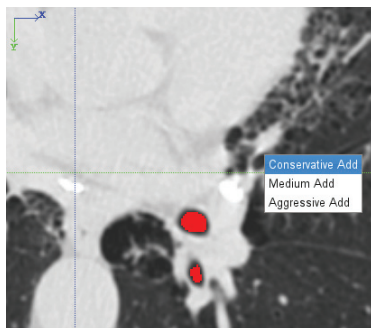


(d) After third mouse-wheel increment, all leak segments are selected. User right-clicks on selection and confirms leak removal through pop-up menu.

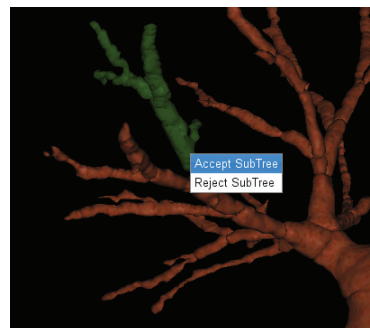


(e) Leak removed.

Fig. 4. Leak removal in Pulmonary Workstation 2.0. A leak can easily be removed with a few mouse clicks.



(a) User placed a seed point in CT view and chooses growth rate from pop-up menu.



(b) User confirms newly grown tree in 3D surface rendering through mouse click in pop-up menu.

Fig. 5. Manual growing of new sub-tree.

The added sub-tree is shown highlighted both in the CT viewer of PW2 as well as in a 3D surface rendering of the complete airway tree. The user can then examine the new addition and either accept or reject it with a simple mouse click [Figure 5(b)].

3 Validation

Twenty CT scans (identified as CASE21–CASE40 below) were made available by the organizers of the *Extraction of Airways from CT 2009 (EXACT09)* contest at the 2nd *International Workshop on Pulmonary Image Analysis*, MICCAI 2009, London. In a first step each case was segmented automatically by PW2. The standard built-in airway segmentation algorithm was used using the default parameters. Subsequently every case was examined by a human operator. Using the PW2 manual editing tools leaks were removed and sub-branches were added if applicable. The final segmentation results were sent to the EXACT09 organizers for evaluation where they were compared against a gold standard. This gold standard was not made available to the contestants prior to scoring.

4 Results

Table 1 was received from the EXACT09 organizers. The last column in this table, labeled “Manual correction time [min]”, was added by the authors of this paper, based on logs kept during the segmentation task.

5 Discussion

Table 1 shows that the method presented here achieved a relatively high number of identified airway branches and a low leakage rate. However, manually growing a lot of additional airway branches throughout the airway tree comes at a price, as can be seen in the last column of Table 1. The operator spent on average almost an hour per case trying to retrieve as many branches as possible.

The PW2 software was optimized for use with a specific imaging protocol. This protocol is based on a diagnostic-quality clinical chest CT. Deviations from the prescribed protocol in dose, reconstruction filter, slice collimation, pitch, lung volume, and other critical imaging parameters may reduce image quality and segmentation performance. The EXACT test cases span a wide range of image acquisition settings, and arguably, most cases are not high-quality diagnostic images. However, even for cases where the PW2 automatic segmentation produced a suboptimal segmentation, the flexible manual editing tools allowed the operator to add to and refine the segmentation if desired.

But from our observations in both clinical applications as well as in research studies where PW2 is used today we know the actual operator time spent per tree is, in most cases, significantly lower. This is because the automatically identified airway tree is often sufficient for the purpose of the specific analysis. And if more

Table 1. Evaluation measures 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 (%)	Manual correction time [min]
CASE21	116	58.3	69.6	63.0	0	0.0	0.00	10
CASE22	218	56.3	181.2	54.8	17	125.3	0.88	180
CASE23	159	56.0	110.6	42.5	22	141.8	1.09	90
CASE24	113	60.8	89.0	54.7	6	275.1	1.58	30
CASE25	169	72.2	145.7	57.8	13	214.6	1.08	50
CASE26	24	30.0	17.6	26.8	0	0.0	0.00	0
CASE27	89	88.1	71.1	87.8	15	265.1	3.03	70
CASE28	96	78.0	72.6	66.3	7	62.7	0.83	50
CASE29	129	70.1	91.8	66.5	10	207.8	2.20	90
CASE30	142	72.8	102.9	67.4	7	81.2	0.91	25
CASE31	147	68.7	111.2	63.3	5	76.3	0.62	50
CASE32	156	67.0	137.3	63.0	14	285.0	1.80	80
CASE33	138	82.1	118.5	80.6	9	79.3	1.00	105
CASE34	329	71.8	256.5	71.7	20	301.8	1.10	70
CASE35	234	68.0	177.3	57.3	18	67.8	0.41	60
CASE36	105	28.8	101.4	24.6	0	0.0	0.00	0
CASE37	138	74.6	127.2	71.6	10	121.8	0.83	90
CASE38	61	62.2	44.8	67.4	5	164.5	2.15	35
CASE39	135	26.0	113.4	27.7	5	188.0	1.86	0
CASE40	275	70.7	243.6	63.0	26	517.2	2.34	90
Mean	148.7	63.1	119.2	58.9	10.4	158.8	1.19	58.75
Std. dev.	71.0	17.1	59.8	16.9	7.5	128.0	0.84	44.10
Min	24	26.0	17.6	24.6	0	0.0	0.00	0.00
1st quartile	105	56.3	72.6	54.7	5	67.8	0.62	28.75
Median	138	68.4	110.9	63.0	10	133.5	1.04	55.00
3rd quartile	218	74.6	177.3	71.6	18	275.1	2.15	90.00
Max	329	88.1	256.5	87.8	26	517.2	3.03	180.00

peripheral airways need to be identified then these are often restricted to a few pre-defined paths. Therefore most small-sized airways need not be delineated. For the test data in the EXACT challenge, the operator was instructed to add all visible airway branches. This is generally not necessary or desirable for a clinical evaluation of the airways. For procedure planning purposes such as a biopsy or the placing of a stent normally only one single airway path is required. Clinical trials often only look at three or four sentinel airway paths.

In that sense the relative high operator times reported here are not representative for most clinical applications of the PW2 software and could be called a “contest artifact”. In the day to day PW2 operation most cases are analyzed much faster.

6 Conclusion

The airway segmentation algorithm presented here is part of VIDA Diagnostic’s “Pulmonary Workstation 2.0” (PW2) software suite, an FDA 510(k) approved software package for the analysis of CT scans of human lungs. PW2 is used in clinical settings for tasks such as the quantification of lung disease and bronchoscopic procedure planning. Today PW2 is also utilized as the central software tool in various clinical trials to determine the efficacy of new treatment methods, and in research studies that seek correlations between lung disease and specific genotypes and phenotypes.

So far over 10,000 CT datasets have been analyzed with the methods described here, and several dozen new cases are added every day.

References

1. Mori, K. and Suenaga, Y. and Toriwaki, J. Automated anatomical labeling of the bronchial branch and its application to the virtual bronchoscopy. *IEEE Transactions on Medical Imaging*. Vol. 19. 103–114 (2000).
2. van Ginneken, B. et al. Robust Segmentation and Anatomical Labeling of the Airway Tree from Thoracic CT Scans. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008*. Part 1. 219–226. (2008)
3. Schlathölter, T. et al. Simultaneous segmentation and tree reconstruction of the airways for virtual bronchoscopy. *Proc. SPIE*, vol. 4684. 103–113. (2002)
4. Lo, P. et al. Voxel classification-based airway tree segmentation. *Proc. SPIE*, vol. 6914 (2008)
5. Tschirren, J. et al. Intrathoracic airway trees: segmentation and airway morphology analysis from low-dose CT scans. *IEEE Trans. Med. Im.* 24(12), 1529–1539. 2005
6. Silvela, J. and Portillo, J. Breadth-first search and its application image processing problems. *IEEE Transactions on Image Processing*. Vol. 10, Issue 8. 1194–1199 (2001).
7. Palágyi, K. and Tschirren, J. and Sonka, M. Quantitative analysis of intrathoracic airway trees: methods and validation. In *Proc. 18th Int. Conf. Information Processing in Medical Imaging, IPMI 2003, Ambleside, UK, Lecture Notes in Computer Science 2732, Springer, 222–233, 2003.*

8. Sonka et al. *Image Processing, Analysis, and Machine Vision*. 2nd ed. Brooks/Cole Publishing. (1998).
9. Cormen, T. et al. *Introduction to Algorithms*. McGraw-Hill (2003).
10. Kiraly, Atilla and Higgins, William E., and McLennan, Geoffrey and Hoffman, Eric A. and Reinhardt, Joseph M. 3D human airway segmentation for clinical virtual bronchoscopy. *Academic Radiology*. Vol. 9, Number 10. 1153–1168 (2002).