A Bottom-up approach for labeling of human airway trees

Pechin Lo[1], Eva M. van Rikxoort[1], Jonathan Goldin[1], Fereidoun Abtin[1], Marleen de Bruijne[2,3], and Matthew Brown[1]

[1] UCLA Thoracic Imaging Research Group, Department of Radiology, University of California, Los Angeles, USA, pechinlo@mednet.ucla.edu,
[2] Image Group, Department of Computer Science, University of Copenhagen, Denmark,
[3] Biomedical Imaging Group Rotterdam, Departments of Radiology & Medical Informatics, Erasmus MC, Rotterdam, The Netherlands.

Abstract. In this paper, an airway labeling algorithm that allows for gaps between the labeled branches is introduced. A bottom-up approach for arriving to an optimal set of branches and their associated labels is used in the proposed method. A K nearest neighbor based appearance model is used to differentiate the different anatomical branches. The proposed method was applied on 33 computed tomography scans of different subjects, where an average of 24 anatomical branches were correctly detected out of a total of 29 anatomical branches. Additionally the proposed method was also evaluated on trees with simulated errors, such as missing branches and having falsely detected branches, where we showed that such errors have little or no effect on the proposed method.

1 Introduction

Airway labeling is the task of assigning anatomical labels to a given airway tree. These anatomical labels correspond to the branches in, roughly, the first six generations of the human airway tree, where similar topology is observed across subjects [1]. Airway labeling algorithms have been commonly used in virtual bronchoscopy to aid navigation, as well as in the segmentation of the lobes and pulmonary segments, as shown in [2]. Another important usage of airway labeling algorithm is in quantification of airways, where airway labeling algorithms are used to match airway trees from different subjects, thus making it possible to directly compare measurements obtained from the airway branches, e.g. dimensions of lumen and wall.

Despite the importance of airway labeling, relatively little work has been done in this area as compared to airway segmentation. Tschirren et al. [1] uses manually labeled trees for constructing the relationship between the labels, and assumes that the geometrical features of the different anatomical branches are independent Gaussian distributions. The geometrical features used for describing the different anatomical branches include segment length, spatial orientation, distance between parent and child branches, and angle difference between branch segment. Labels are assigned by finding the maximum clique in a weighted association graph based approach that is constructed based on the learned normal distributions. Van Ginneken et al. [3] uses a similar normal distribution based model as in [1], but proposed to iteratively look for the most probable branches with a labeled parent. Mori et al. [4] proposed to label airway branches by comparing orientation of their children to the ones in the training set, where the labels assigned are the ones with the children that give the best matching.

In this paper, we propose an airway labeling algorithm that allows for "gaps" or unlabeled branches between the labeled branches. One of the advantage of allowing gaps is that it enables anatomic variations to be handled naturally, without the need for intermediate labels that may differ between subjects. Besides natural variations, the problem of extra generations or intermediate branches caused by falsely detected branches (false positives) from segmentation algorithms can also be solved by simply treating them as gaps.

Because of the natural anatomical variations in the topology of the airway trees between different subjects, it is quite logical that the underlying distribution of the geometrical features of the anatomical branches should be multimodal as well, such that features from a particular type of variation should cluster together. For this reason, a K nearest neighbor (KNN) model is used to model the geometrical characteristic of the anatomical branches for the proposed method, in contrast to e.g. a Gaussian based model [1] that assumes a single mode distribution.

The proposed method also features a searching scheme that goes through all combination of possible solutions before arriving to the final best solution, thus making it unlikely to be trapped in a local minima in the solution space. It may seem as if the idea of going through all combinations can be very complicated and impractical to compute. However, because of the various constraints and certain properties unique to the problem, it is possible to subdivide the problem into multiple small sub-problems, where the solution can be obtained efficiently via a bottom-up approach.

This paper is organized as follows: We start by explaining the methodology of the proposed airway labeling algorithm in Section 2, where details of the KNN based appearance model and the bottom-up labeling scheme is described. Experiment results based on manually assigned labels from 33 CT scans from different subjects are presented in Section 3. The paper finally ends with discussion and conclusions in Section 4.

2 Methodology

The proposed method assumes that segmented airway trees from both the target scan and the training set scans are provided, with the individual branches extracted for each of the airway trees. Information about the centerline and radius are available for each of the branches, as well as anatomical labels for the branches in the training set. With these data available, the proposed method starts by constructing models of the anatomical branches that we would like

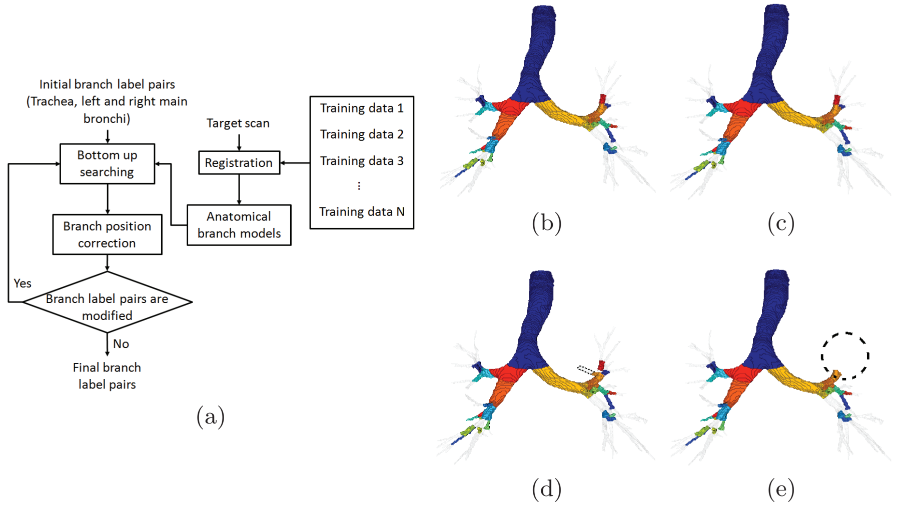

Fig. 1. (a) Block diagram of the labeling process. (b) Renderings of the reference standard for a particular tree, with the labels assigned shown in different colors, and unassigned branches shown in translucent white. (c) Renderings of the result from the proposed method on the untampered tree. (d) Rendering of the result with an added falsely detected branch, highlighted with dashed outline. (e) Rendering of the result with the children of an intermediate branch removed, with the region where branches are removed highlighted in the circle.

to label from the training set. An iterative labeling scheme, which consists of a bottom-up searching procedure and a branch position correction procedure at each iteration, is used to assign labels to the most plausible branches according to the constructed models. The iterative labeling scheme stops once the labeled branches in the current iteration are the same as the ones in the previous iteration. Figure 1(a) presents a block diagram of the proposed method.

We define $\mathcal{L} = \{l_1, l_2, ..., l_n\}$ as the set containing all the anatomical labels that we are interested in. We represent a tree in the form of a set $\mathcal{B} = \{b_1, b_2, ..., b_m\}$, where the elements b_i are the individual branches of the segmented tree.

As mentioned earlier in the introduction, a unique approach of this work is that there is no need for a parent and child relationship between the labeled branches. In order to indicate a certain sequence or relationship between the labeled branches, the terms “precede” and “succeed” will be used in the following text instead. To give an example of the new terminology, when we say that a labeled branch b_1 is preceded by a labeled branch b_2, we mean that b_1 is a descendant of b_2 and that the nearest ancestor of b_1 with a label assigned is b_2. The same concept also applies to “succeed”.

2.1 KNN based appearance models for the anatomical branches

The proposed KNN based appearance model is based on the mean distances of K nearest neighbors. An advantage of the KNN distance measure is that it accounts for clusters in features space, thus as long as a point in feature space is near to a cluster, the distance measure will be relatively small. This is in contrast to, for example, taking the distance the from the mean of all samples in feature space, where there is a possibility that the mean could in actual be far away from the individual clusters, causing the distance measure to not reflect the actual distribution of the data.

In order to take into account the distribution of the data in feature space, the Mahalanobis distance is used as a distance measure for the KNN distance. Given Φ as the covariance the samples from the training set, we define the probability for a feature vector $\mathbf{x}$ to belong to this particular distribution as

$$P(\mathbf{x}) = \exp\left(-\frac{1}{K|\mathbf{x}|}\sum_{i=1}^{K}\left(D\left(\mathbf{x},\mathbf{q}_i;\Phi\right)\right)^2\right), \tag{1}$$

where $(D(\mathbf{x},\mathbf{q}_i;\Phi))^2 = (\mathbf{x}-\mathbf{q}_i)^T(\Phi)^{-1}(\mathbf{x}-\mathbf{q}_i)$ is the Mahalanobis distance, $\mathbf{q}_1$, $\mathbf{q}_2$, ..., $\mathbf{q}_K$ are the K nearest neighbors of $\mathbf{x}$ from the training samples based on $D(.,.;\Phi)$, and $|.|$ denotes the cardinality of a given set.

Two appearance models using the KNN based model are constructed to decide which of the different anatomical labels in $\mathcal{L}$ a given branch is more similar to. The two models consist of a local appearance model that describes the appearance with respect to the whole tree, and a relative appearance model that describes the difference in appearance between branches from labels that precede each other.

Prior to the construction of the appearance model, the scans in the training set are first rigidly registered to the target scan to establish a standard frame of reference for all the scans. The transformation, from the rigid registration, is then applied to the segmented airway trees associated to each of the registered scans in the training set. For a scan j in the training set, we define $\mathcal{B}_j = \{b_{j,1}, b_{j,2}, ..., b_{j,m_j}\}$ as the tree for scan j after transformation is applied, where $b_{j,i}$, with $i = 1, 2, ..., m_j$, is a branches in $\mathcal{B}$. For convenience, from here on we will refer to a branch in $\mathcal{B}_j$ with the anatomical label $l \in \mathcal{L}$ as $b_{j,l}$.

After completing the registration process, we can now compute the following features from any given branch b in a segmented airway tree:

- The starting point, $\mathbf{x}(b) \in \Re^3$, of the centerline of b,
- the distance, $d_l(b)$, between the starting point of b to the starting point of the root of the tree measured along the centerlines,
- a unit vector, $\mathbf{v}(b) \in \Re^3$, representing orientation of branch b,
- assuming an elliptical profile at the cross-section (perpendicular to the centerline) of a branch, the average radius of a branch is stored in the form of a vector, $\mathbf{r}(b) \in \Re^2$, which contains the average radii measured along the major and minor axis of b.

The first appearance model is the local appearance model, that describes the features of an anatomical label by looking only at the features computed from branches in the training set belonging to this label, with respect to the root of their respective tree. Given b_{root} as the root of the tree, the following features are used to describe the local appearance of a branch b:

- Starting coordinates with respect to the root, $\mathbf{x}_a(b, b_{root}) = \mathbf{x}(b_{root}) - \mathbf{x}(b)$,
- Euclidean distance and the distance measured along the centerlines with respect to the root, $\mathbf{d}_a(b, b_{root}) = [\|\mathbf{x}_a(b, b_{root})\|_2 \; (d_l(b) - d_l(b_{root}))]$,
- orientation $\mathbf{v}(b)$,
- average radius measured at the major and minor axis $\mathbf{r}(b)$.

The computed features are then stored in the form of a feature vector $\mathbf{f}(b, b_{root}) = [\mathbf{x}_a(b, b_{root}) \; \mathbf{d}_a(b, b_{root}) \; \mathbf{v}(b) \; \mathbf{r}(b)]^T$. The covariance, Φ_l, associated with label l is computed from the feature vector of the branches belonging to l in the training set, with the assumption that $\mathbf{x}_a$, $\mathbf{d}_a$, $\mathbf{v}$ and $\mathbf{r}$ are independent of each other. Given a new feature vector $\mathbf{g}$ obtained from some branch in a tree and following (1), the probability of the new feature belonging to label l is computed as follows:

$$P_{local}(l|\mathbf{g}) = \exp\left(-\frac{1}{K|\mathbf{g}|}\sum_{i=1}^{K}\left(D\left(\mathbf{g}, \mathbf{f}_i; \Phi_l\right)\right)^2\right), \tag{2}$$

where $\mathbf{f}_1$, $\mathbf{f}_2$, ..., $\mathbf{f}_K$ are the K nearest neighbors of $\mathbf{g}$ from the feature vectors collected from the training set.

The second appearance model is the relative appearance model that describes the difference in the features between branches from two labels that precedes each other. Features for the relative appearance model are computed based on a current labeled branch b_c and its preceding labeled branch b_p, and is as follows:

- Difference in distance from root, $\mathbf{d}_r(b_c, b_p) = [\|\mathbf{x}(b_p) - \mathbf{x}(b_c)\|_2 \; (d_l(b_c) - d_l(b_p))]$,
- difference in orientation, $\mathbf{v}_r(b_c, b_p) = \mathbf{v}(b_p) - \mathbf{v}(b_c)$,
- difference in radius, $\mathbf{r}_r(b_c, b_p) = \mathbf{r}(b_p) - \mathbf{r}(b_c)$.

The computed features are then stored in the form of a feature vector $\mathbf{f}'(b_c, b_p) = [\mathbf{d}_r(b_c, b_p) \; \mathbf{v}_r(b_c, b_p) \; \mathbf{r}_r(b_c, b_p)]^T$. Similar to the local appearance model, the covariance, $\mathbf{F}_{l_c,l_p}$, for branches belonging to l_c that precede by branches belonging to l_p is constructed from the training set by assuming independence between $\mathbf{d}_r$, $\mathbf{v}_r$ and $\mathbf{r}_r$. given a new feature vector $\mathbf{g}'$ computed from two branches that precedes each other, the probability of the new feature belonging to label l_c with a preceding label l_p can be computed as:

$$P_{rel}(l_c|l_p, \mathbf{g}') = \exp\left(-\frac{1}{K|\mathbf{g}'|}\sum_{i=1}^{K}\left(\left(D\left(\mathbf{g}', \mathbf{f}'_i; \Phi_{l_c,l_p}\right)\right)\right)^2\right), \tag{3}$$

where $\mathbf{f}'_1$, $\mathbf{f}'_2$, ..., $\mathbf{f}'_K$ are the K nearest neighbors from the training set.

2.2 A bottom-up searching scheme

Given a branch with a known label, the aim of the searching scheme is to obtain a combination of branches and labels among the descendants of the given branch that results in the best overall confidence according to the appearance models, subjected to the following constraints: (1) The succession and precedence among the branches with labels assigned must exists in the training data; (2) A label can only be assigned to a single branch and vice versa; (3) Labels assigned to branches at the "leaf" level, or labeled branches that are not succeeded by any other labeled branches, cannot be removed if the confidence is sufficiently high. The overall confidence is defined as the average confidence of all the assigned labels.

We use the notation (b, l) to represent the notion of assigning a label $l \in \mathcal{L}$ to a branch $b \in \mathcal{B}$, which we refer to as a "branch label pair". To indicate that a branch label pair (b_c, l_c) is succeeded by another branch label pair (b_p, l_p), we use the notation (b_c, l_c, b_p, l_p), which we refer to as "branch relationship pair". Based on the constructed models given in (2) and (3), we define the confidence for a given branch relationship pair (b_c, l_c, b_p, l_p) as

$$\mathrm{conf}(b_c, l_c, b_p, l_p) = P_{local}(l_c|\mathbf{f}(b_c, b_{root})) + P_{rel}(l_c|l_p, \mathbf{f}'(b_c, b_p)) \tag{4}$$

where b_{root} is the root branch of the current tree.

Rephrasing the overall problem, the aim of the searching scheme is to find a set of branch relationship pairs from all possible combinations, subjected to the constraints mentioned earlier, which results in the highest average confidence. More generally, provided with a branch relationship pair $\eta_{c,p} = (b_c, l_c, b_p, l_p)$, we can define the optimal set of branch relationship pairs, $\mathbf{W}(\eta_{c,p})$ for the subtree of (b_c, l_c) as

$$\operatorname*{arg\,max^*}_{\mathbf{W}(\eta_{c,p})\subset\mathbf{H}_{c,p}} \left(\frac{1}{|\mathbf{W}(\eta_{c,p})|} \sum_{\eta_{x,y}\in\mathbf{W}(\eta_{c,p})} \mathrm{conf}(\eta_{x,y}) \right), \tag{5}$$

where $\mathbf{H}_{c,p}$ is the set containing all valid combinations of branch relationship pairs within the subtree of (b_c, l_c) and $\arg\max^*(.)$ is an argument of the maximum that takes into account of the constraints mentioned in the beginning of this section. Additionally, the overall confidence of the subtree for $\eta_{c,p}$ can be defined as

$$c(\eta_{c,p}) = \frac{1}{|\mathbf{W}(\eta_{c,p})| + 1} \left(\mathrm{conf}(\eta_{c,p}) + \sum_{\eta_{x,y}\in\mathbf{W}(\eta_{c,p})} \mathrm{conf}(\eta_{x,y}) \right). \tag{6}$$

Let $\mathbf{N}(\eta_{c,p})$ be a set of branch relationship pairs in $\mathbf{W}(\eta_{c,p})$ that directly succeed $\eta_{c,p}$. It is quite obvious that $\mathbf{N}(\eta_{c,p})$ can be solved as follows:

$$\operatorname*{arg\,max^*}_{\mathbf{N}(\eta_{c,p})\subseteq\mathbf{h}_{c,p}} \left(\sum_{\eta_{x,c}\in\mathbf{N}(\eta_{c,p})} c(\eta_{x,c}) \right), \tag{7}$$

where $\mathbf{h}_{c,p}$ is the set containing all possible branch relationship pairs that succeeds $\eta_{c,p}$. Subsequently, (6) can now be rewritten into

$$c(\eta_{c,p}) = \frac{1}{|\mathbf{W}(\eta_{c,p})| + 1} \left(\mathrm{conf}(\eta_{c,p}) + \sum_{\eta_{x,c} \in \mathbf{N}(\eta_{c,p})} (|\mathbf{W}(\eta_{x,c})| + 1) c(\eta_{x,c}) \right), \quad (8)$$

and (5) into

$$\mathbf{W}(\eta_{c,p}) = \bigcup_{\eta_{x,c} \in \mathbf{N}(\eta_{c,p})} \mathbf{W}(\eta_{x,c}). \quad (9)$$

What the series of equations in (7), (8) and (9) showed is that $\mathbf{W}(\eta_{c,p})$ can actually be solved in a recursive manner, where each iteration only involves the current branch relationship pair, $\eta_{c,p}$, being evaluated and all combination of succeeding branch relationship pair $\mathbf{h}_{c,p}$.

Additionally, since $\mathbf{W}$ is no longer needed for computing the confidence as in (6), branch label pairs can also be stored instead of branch relationship pairs, which is a simpler data structure that also reflects on how a tree is labeled.

In the following, we will describe the algorithm to perform the search and show in details how (7) is solved. The searching scheme consists of two algorithms, a first algorithm that generates all possible valid combination of branch relationship pairs and a second algorithm that uses a bottom-up approach to obtain the optimal set of branch label pairs.

Generation of all possible valid combination Given an initial branch relationship pair $(b_{init}, l_{init}, 0, 0)$, the algorithm starts by generating all valid (that existed in the training set) branch relationship pairs that directly succeed it. The same process is then repeated for all the generated branch relationship pairs and continues until no more valid branch relationship pairs can be generated. To keep the number of combination to a manageable size, we limit the distance between succeeding branch label pairs to no more than D_{search} generation apart. The initial and generated branch relationship pairs are all stored in the form of a set $\mathbf{S}$, which is to be used in the next part of the searching scheme.

Bottom-up approach for obtaining the optimal set of branch label pairs The aim of this algorithm is to solve (7), and compute (8) and (9). The algorithm starts by processing the branch relationship pairs in $\mathbf{S}$ that are in the leaf level, where c in (8) and $\mathbf{W}$ in (9) can be computed directly without the need to solve (7). When computing branch relationship pairs at leaf level, a value of negative one is assigned to c if their confidence is lower than T_c to indicate rejection.

In the next iteration, the algorithm moves on to the branch relationship pairs that precede the previously processed branch relationship pairs. During this iteration, if a branch relationship pair is still at the leaf level because all succeeding branch relationship pairs are rejected, the same treatment as mentioned above is applied. For the branch relationship pair that is no longer at leaf level, we now need to solve (7). Let (b_c, l_c, b_p, l_p) be the current branch relationship that is being evaluated, (7) can be solved by iteratively selecting a succeeding branch relationship pair (b_m, l_m, b_c, l_c) with the highest c that also fulfills the following criteria:

- b_m and l_m are not in any of the branches from the selected branch relationship pairs and their associated $\mathbf{W}$,
- b_m is neither a descendant nor an ancestor of the branches from the selected branch relationship pairs.

This selection process ends once no more branch relationship pair can be selected. Additionally, to ensure a sufficient number of branch relationship pairs are selected, the labels of the selected branches and their associated $\mathbf{W}$ are compared to a set $\mathbf{L}_{freq}$, which contains labels that frequently (>50% occurence in the training set) succeeds l_c. If the set of selected branch relationship pairs and their associated $\mathbf{W}$ do not contain all the labels in $\mathbf{L}_{freq}$, then the selection process is repeated on permutations of the current selected branch relationship pairs being marked for exclusion. If the branch relationship pairs that contains all labels in $\mathbf{L}_{freq}$ still cannot be found, then all the selected branch relationship pairs will be marked, and whole process is repeated again on the remaining unmarked branch relationship pairs. The process is repeated until an optimal set of branch relationship pairs containing all the labels in $\mathbf{L}_{freq}$ are found, or if all succeeding branch relationship pairs are marked, the results from the first selection process will be used.

The algorithm iterates through all the branch relations pairs in $\mathbf{S}$ until it reaches the initial branch parent pair $(b_{init}, l_{init}, 0, 0)$, where the solution for the optimal set of branch label pairs is then $\mathbf{W}(b_{init}, l_{init}, 0, 0)$.

2.3 Branch position correction

A problem with the bottom-up searching scheme is that the branch selected for a particular anatomical label may not be the root of the actual anatomical branch. Therefore there is a need for a correction process to move the labeled branch as near as possible to the preceding labeled branch.

Given $\mathbf{W}$ as a set of branch label pair for a tree, the branch position correction procedure starts by processing the branch label pair belonging to the trachea, and then move on to its children until the leaf branch label pairs are reached. Let $(b, l) \in \mathbf{W}$ be the branch label pair that is currently being processed. The processing for (b, l) will be skipped if it only has a single succeeding branch label pair. If it has more than one branch label pair, then for each succeeding branch label pair, (b_i, l_i), we allow it to keep moving up the generation as long as the following conditions are fulfilled:

- $\text{conf}(b_i^-, l_i, b, l) \geq T_c$ only if (b_i, l_i) is at the leaf level,
- b_i^- is not an ancestor nor a descendant of the other succeeding branches,
- $(b_i^-, l_w) \notin \mathbf{W}, \forall l_w \in \mathcal{L}$,

where b_i^- is a parent branch that it is moving to. If a move is allowed, b_i will be updated with the new branch that it moved to in $\mathbf{W}$. Subsequently, all descendants of b_i in $\mathbf{W}$ will be removed from $\mathbf{W}$.

Once all the branch label pairs in $\mathbf{W}$ are processed, the searching procedure is repeated again using the branch label pairs in $\mathbf{W}$ at the leaf level as the initial set of branch label pairs.

2.4 Detection of trachea and main bronchi

The bottom-up searching scheme described above requires a starting branch with a known label. In this work, both the left and right main bronchi are used as starting branches for the searching scheme.

The process for obtaining the left and right main bronchi starts by identifying N_{trac} candidate branches with the highest probability for belonging to the trachea, according to (2), where each branch is a b_{root} itself. For each of the trachea candidates, two branches with the highest confidence according to (4) are selected, subject to the constraints that they are neither ancestors nor descendants of each other. Let b_{trac} be the candidate branch for the trachea, b_{lmb} and b_{rmb} be the selected branch for the left and right main bronchi respectively, we define the overall confidence score of these three branches as

$$\begin{aligned} q(b_{trac}, b_{lmb}, b_{rmb}) =& 2P_{local}(b_{trac}, Trac) \\ &+ \mathrm{conf}(b_{lmb}, LMB, b_{trac}, Trac) \\ &+ \mathrm{conf}(b_{rmb}, RMB, b_{trac}, Trac). \end{aligned}$$

The candidate branch, along with its associated left and right main bronchi with the highest q is selected. The final left and right main bronchi is obtained after applying the branch position correction algorithm on those from the selected candidate trachea, with T_c set to zero.

3 Experiment and Results

A total of 33 clinical computed tomography (CT) scans of different subjects from a clinical trial were used in this work. The in plane resolution of the scans range from 0.6 to 1 mm, with the slice thickness ranging from 0.5 to 1 mm. A total of 29 anatomical airway branches up till the segmental level (Trachea, LMB, RMB, LLIB, LLLB, LSUB, LULB, RILB, RLLB, RMLB, RULB, LB1+2, LB3, LB4, LB5, LB6, LB7+8, LB9, LB10, RB1, RB2, RB3, RB4, RB5, RB6, RB7, RB8, RB9, RB10) were manually identified in the scans by two experienced chest radiologists.

Segmentation of the airway trees in the scans were obtained using the method presented in [5, 6], which combines a voxel classification based appearance model and similarity between orientations of the airways with nearby vessels. For splitting a segmented airway tree into its individual branches and also obtaining the centerlines, a fast marching based algorithm that detects bifurcation as splitting fronts as presented in [7] is used. The lumen (defined by the segmented airway tree) radii measured along the major and minor axis were obtained for each point by applying principal component analysis on the segmented region that lies within a plane perpendicular to the centerline.

It should be noted that, our experiments were conducted only on the branches extracted by the segmentation algorithm. Branches that were identified by the radiologists but were not in the segmentation were excluded from the experiments. Because of the way anatomical labels for airway branches are defined, it is possible for multiple branches in a tree to share the same label. For such cases, only the label of first encountered branch is retained, with the rest treated as gaps.

Two fold cross validation was used for evaluating the performance of the proposed method, where the scans were divided into group A and group B that consist of 16 and 17 scans respectively. The scans from group A is then used as training set, with scans from group B as testing set in the first fold, and vice versa in the second fold.

3.1 Parameters and settings

The `elastix` [8] registration package was used to performed the rigid registration, where the registration was applied with Gaussian image pyramid (without down sampling) and a mutual information based cost function[4]. Additionally, to save computation time, the scans were also down sampled to $1.5 \times 1.5 \times 1.5$ mm prior to registration.

Three sets of appearance models, including both local and relative appearance model, were constructed in this work. The first set, containing the trachea, left and right main bronchi, was constructed using the trachea as root. This set of models was used for obtaining the branches belonging to the trachea, and the left and right main bronchi, as described in Section 2.4. The second and third set were constructed using the left and right main bronchi as root respectively, and contains only anatomical labels that come after the left and right main bronchi respectively. Both sets were used to obtain the labels of the subtrees of the left and right main bronchi respectively, as described in Section 2.2. A value of $K = 4$ was used to compute both P_{local} and P_{rel} in (2) and (3). When computing the covariances of the appearance models, the eigenvalues of the covariance matrix were capped to a minimum value of 0.01% of the mean.

For extracting the trachea, left and right main bronchi, the number of trachea candidates, N_{trac}, was set to 20. In the bottom-up searching scheme, the maximum distance between a labeled branch and its preceding labeled branch, D_{search}, was set to 5. The threshold for the minimum confidence an end branch can have, T_c, was determined automatically by performing leave-one-out experiment on the training set, where the threshold that assigns the most number of labels correctly was selected.

[4] Parameters settings are available at
http://elastix.bigr.nl/wiki/images/1/1a/Par0003.affine.txt

3.2 Results

Two evaluation measures were used to evaluate the performance measures of the proposed method, which are true positive rate (TPR) and precision. TPR is defined as the number of labels assigned correctly divided by the total number of labels in the reference. Precision is defined as the number of labels assigned correctly over the total number of labels assigned. Due to the nature of the reference and the method, which contains gaps between labeled branches, a label is said to be correctly assigned if it is assigned to either the same branch as in the reference or to one of the branches in the "gaps" between the actual labeled branch and its succeeding labeled branches in the reference.

Average TPR and precision of the proposed method in the two fold cross validation experiments is 0.83 and 0.87 respectively. If we only limit the evaluation to the labels at the segmental level, we have an average TPR of 0.77 and an average precision of 0.83.

To study the effects of falsely detected and missing branches on the proposed method, a series of experiments were conducted on a randomly selected scan from the 33 scans, with the remaining 32 scans served as training set. The rendering of the reference standard for the selected scan is shown in Figure 1(b). From the segmented airway tree of the selected scan, we generated 8 tampered trees with a fake branch added to one of the anatomical branches in each tree to simulate a falsely detected branch. The fake branch added is a cylinder of 20 mm in length, departing at a random angle no more than 100 degree from the originating branch and with a radius that is determined randomly according to the radius of the originating branch. To simulate missing branches, another 8 tampered trees were generated by removing the children of an intermediate branch in each tree. Without tampering, both the TPR and precision of the of proposed method for the selected scan is 1, with all 29 labels assigned correctly, as shown in Figure 1(c). The average TPR and precision remain the same for the cases where a fake branch is added. For the case with children of a branch removed, the average TPR remain unchanged, however we do observe a slight drop in average precision, which is 0.99 and 0.98 for all branches and branches at the segmental level respectively. Figure 1(d) and 1(e) show renderings of the results of a case from each of the two types of tampered trees.

4 Discussion and Conclusions

Promising results were obtained from the proposed method, with an average TPR of 0.83 in the two fold cross validation, which amounts to an average of 24 out of 29 branches being detected. Although a lower TPR of 0.77 was observed for branches at the segmental level, it is still an average of 14 branches detected out of a total of 18 branches. The overall precision of the proposed method is 0.87, which is at most 5 out of 29 branches labeled wrongly on average.

The experiments were performed using a 2.67 GHz PC. It took between 1 to 2 minutes to register a scan in training set to the target scan. After the registration, the whole labeling process took no more than 15 seconds per scan.

Although it was only tested on the segmented tree from a single scan, it seems that fake branches have no effect on the results of the proposed method, which is as expected. Additionally, removal of children of a branch has also very little effect on results the proposed method, with the main error lying in the proposed method assigning one or two extra labels that are wrong in a few cases.

The overall TPR and precision presented here seems lower than other works [4, 1, 3]. However, it should be noted that we only have 29 labels in total, which is less than those used in other works, where the number can go up to more than 40 labels, with most of the extra labels belonging to the intermediate branches.

To summarize, we proposed an airway labeling algorithm that has the following features: (1) Allow gaps between labeled branches to better handle topological variations; (2) A KNN based appearance model to take into account a clustered feature space distribution; (3) A bottom-up searching scheme that results in a globally optimal solution, with decisions made at a subtree level and rejection occurring only at the leaf level. The proposed method was evaluated on 33 CT scans from different subjects and on segmented trees with simulated errors with promising results.

References

1. Tschirren, J., McLennan, G., Pálágyi, K., Hoffman, E., Sonka, M.: Matching and anatomical labeling of human airway tree. IEEE Trans Med Imaging **24**(12) (Dec. 2005) 1540–1547
2. Mori, K., Nakada, Y., Kitasaka, T., Suenaga, Y., Takabatake, H., Mori, M., Natori, H.: Lung lobe and segmental lobe extraction from 3D chest CT datasets based on figure decomposition and Voronoi division. In: Medical Imaging 2008: Image Processing. Volume 6914., SPIE (2008) 69144K
3. van Ginneken, B., Baggerman, W., van Rikxoort, E.: Robust segmentation and anatomical labeling of the airway tree from thoracic CT scans. In: Medical Image Computing and Computer-Assisted Intervention. Volume 5241 of Lecture Notes in Computer Science. (2008) 219–226
4. Mori, K., Ema, S., Kitasaka, T., Mekada, Y., Ide, I., Murase, H., Suenaga, Y., Takabatake, H., Mori, M., Natori, H.: Automated nomenclature of bronchial branches extracted from ct images and its application to biopsy path planning in virtual bronchoscopy. Med Image Comput Comput Assist Interv **8**(Pt 2) (2005) 854–861
5. Lo, P., Sporring, J., de Bruijne, M.: Multiscale vessel-guided airway tree segmentation. In: Proc. of Second International Workshop on Pulmonary Image Analysis. (2009) 323–332
6. Lo, P., Sporring, J., Ashraf, H., Pedersen, J.J.H., de Bruijne, M.: Vessel-guided airway tree segmentation: A voxel classification approach. Med Image Anal **14**(4) (Aug 2010) 527–538
7. 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) 175–189
8. Klein, S., Staring, M., Murphy, K., Viergever, M.A., Pluim, J.: elastix: A toolbox for intensity-based medical image registration. IEEE Trans Med Imaging **29**(1) (2010) 196–205