

Semi-automated segmentation of pulmonary lobes in chest CT scans using evolving surfaces

Pechin Lo¹, Eva M. van Rikxoort², Jonathan Goldin¹, and Matthew S. Brown¹

¹ Center for Computer Vision and Imaging Biomarkers, Department of Radiological Sciences, David Geffen School of Medicine, University of California, Los Angeles, USA, pechinlo@mednet.ucla.edu,

² Diagnostic Image Analysis Group, Radboud University Nijmegen Medical Centre, The Netherlands

Abstract. Automated segmentation of the pulmonary lobes from chest CT scans is a challenging problem that is yet to be solved reliably. Therefore there is a need for a semi-automated solution in the case where the automated solution fails. We present an approach that can be used for correcting an existing lobe segmentation or segmenting the lobes from scratch in a semi-automatic manner. The method is based on an iterative approach that evolves a surface based on a voxel based fissure confidence function, smooth prior and user input points. An advantage of the proposed method is that it takes into account both inputs from user and the appearance of fissures in the image, which in turn reduces the number of user interactions required. The proposed method was trained and tuned on 18 CT scans, and tested on 22 CT scans from different subjects with either idiopathic pulmonary fibrosis or severe emphysema. On average, the proposed method requires 37 user drawn line segments, which are mostly short, to segment all lobes accurately. We did not notice a large difference in the number of required line segments between starting from scratch or correcting lobe segmented from an automated method, as it usually requires only two lines in two different view plane from the user to obtain a relatively accurate fissure from scratch.

1 Introduction

Segmentation of pulmonary lobes in chest computer tomography (CT) scans is a prerequisite for many analyses of lung diseases in clinical trials [1, 2]. Various automated solutions have been proposed [3–6]. As observed from the results in LObe and Lung Analysis 2011 (refer to lola11.com/), the problem of automated lobe segmentation is still far from solved. Therefore, there is a need for a semi-automated method for correction purposes.

Despite the need for semi-automated solutions for correcting lobe segmentation, relatively few publications on this topic is available in the literature [7, 6]. Ross et al. [7] proposed the use of thin plate splines that are controlled by user input points for segmenting the fissures. Lassen et al. [6] proposed to model fissures in the form of a 3D surface in which the surface can be raised or lowered

by user inputted lines and Laplacian smoothing was used to ensure the overall smoothness of the resulting surface.

In our previous work [8], we have presented the use of evolving surfaces for segmenting the lobes automatically. In this paper, we will show how the evolving surface method can be adapted for semi-automated lobe segmentation, either for correcting an existing lobe segmentation or to segment lobes from scratch. In addition to user input points, our proposed method also takes into account fissure-like structures in the image via a fissure confidence function. This results in more accurate fissure segmentation with less user interactions, especially for cases with obvious fissures.

2 Methodology

Similar to [8], the proposed method consists of two main components, which are the computation of fissure confidence function and the surface evolution algorithm. A human user interacts with the surface evolution algorithm by providing a set of user input points, which is used to further constrain the surface evolution algorithm. The user can then gradually add more points to the set until the desired result is obtained.

We start by describing the fissure confidence function that is based on a supervised fissure enhancement filter. This is followed by the surface evolution algorithm, which is an iterative process that modifies a given surface such that the overall fissure confidence is maximized, with the constraint that the resulting surface must be smooth and near to user input points. Finally, the way the proposed method is initialized is described.

2.1 Fissure confidence function

The fissure confidence function is based on the supervised enhancement filter presented in van Rikxoort et al. [9]. The filter consists of a two stage K nearest neighbor (KNN) classifier, that is trained to distinguish between fissure and non-fissure voxels. In the first stage, a set of Gaussian derivative and Hessian based features are computed at different scales from the CT scans for all training samples, which belong to either the fissure class or non-fissure class. The computed features are then used to train the stage one KNN classifier, which estimates the probability of a voxel being a fissure as the fraction of samples that are fissures among the K nearest neighbors. In the second stage, Gaussian derivative and Hessian based features of the training samples are computed from the first stage probability image. The probability image based features are then combined with the CT scan based features computed in the first stage. The combined features are then used again to train a stage two KNN classifier, which results in the final probability estimates of a voxel being a fissure.

Although the results of the supervised fissure enhancement filter are usually good, they may be noisy at times, e.g., misclassifying voxels adjacent to a fissure, resulting in a thick slab of detected fissure. In order to remove such noise and

to better localize actual fissure voxels, the following fissure confidence function, \mathcal{C} , for a voxel \mathbf{x} is used

$$\mathcal{C}(\mathbf{x}) = \begin{cases} \mathcal{P}(\mathbf{x}; \sigma) + \left(1 - \left|\frac{\lambda_2(\mathbf{x}; \sigma)}{\lambda_1(\mathbf{x}; \sigma)}\right|\right), & \text{if } \lambda_1(\mathbf{x}; \sigma) < 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $\mathcal{P}(\cdot; \sigma)$ is the probability image observed under a Gaussian kernel of scale σ , and $\lambda_1(\mathbf{x}; \sigma) \geq \lambda_2(\mathbf{x}; \sigma)$ are the largest and the second largest eigenvalues from the Hessian matrix computed at \mathbf{x} from $\mathcal{P}(\cdot; \sigma)$. Intuitively, the confidence of a voxel will only have a non-zero value if its probability is higher than the voxels nearby. In order to have a high confidence, a voxel must have a high probability of belonging to the fissure and its surroundings in the probability image resemble a plate like structure.

2.2 Surface evolution algorithm

A fissure is represented using a 3D surface, which is modeled using a height map that resides on a 2D reference plane that has a normal \mathbf{n} and passes through a point \mathbf{x}_0 . All points $\mathbf{p} \in \mathbb{R}^2$ on the 2D reference plane contain the the height of the surface in the normal direction from the reference plane, which can be mapped to a surface voxel coordinate via a function $f: \mathbb{R}^2 \rightarrow \mathbb{R}^3$.

The surface evolution algorithm uses a multi-scale approach, where the evolution process starts at the coarsest scale and ends at the finest scale. During the evolution process at a particular scale σ , the surface is iteratively evolved such that the total confidence measure computed at σ is maximized, with the constraint that the surface must be smooth and that it must be approximately near a set of user inputted points \mathbf{P} . Each iteration in the evolution process consists of three steps, which are displacement, smoothing and reconstruction.

In the displacement step, the height of all the points in the height map are adjusted independently, such that the total fissure confidence is maximized. This is achieved by searching locally, within a radius of 10 voxels, for the nearest local maxima in the confidence.

Once the new heights of all the points in the height map are determined, the smoothing step is performed by applying a modified Laplacian smoothing algorithm on the heights, where the update equation for the height of a point \mathbf{p} is defined as

$$h_{t+1}(\mathbf{p}) = (1 - \omega(\mathbf{p}))h_t(\mathbf{p}) + \frac{\omega(\mathbf{p})}{|\Omega_{\mathbf{p}}|} \sum_{\mathbf{q} \in \Omega_{\mathbf{p}}} h_t(\mathbf{q}) \quad (2)$$

where $\Omega_{\mathbf{p}}$ is a set containing the immediate four neighbors of \mathbf{p} and $|\Omega_{\mathbf{p}}|$ is the cardinality of $\Omega_{\mathbf{p}}$. The weight ω , which controls the amount of smoothing occurred in each iteration of the modified Laplacian smoothing algorithm, is designed such that less smoothing (lower weight) occur at points with high confidence compared to those with lower confidence, and is defined as

$$\omega(\mathbf{p}) = \min \left(\omega_{\max} \left(1 - \frac{\mathcal{C}(f(\mathbf{p}))}{\max(\mathbf{S})} \right), \omega_{\min} \right) \quad (3)$$

for $f(\mathbf{p}) \notin \mathbf{P}$ in the height map, where \mathbf{S} is the set containing the confidence of all the points on the surface prior to smoothing, and ω_{\min} and ω_{\max} are the minimum and maximum smoothing weight respectively. For $f(\mathbf{p}) \in \mathbf{P}$, ω is set to ω_{\min} .

The reconstruction step adjust the reference plane of the height map so that it is optimal for the representation of the surface. By grouping voxel coordinates of all points on the surface that have a non-zero fissure confidence \mathbf{X} and the user input points \mathbf{P} , the weighted mean coordinates and weighted covariance matrix are computed, where the weights of \mathbf{X} corresponds to their fissure confidence (in Equation 1) and the weight of \mathbf{P} is a constant value of 2. The \mathbf{x}_0 of the new reference plane is then the mean, and \mathbf{n} is the eigenvector corresponding to the smallest eigenvalue of the covariance matrix. The coordinates of \mathbf{X} and \mathbf{P} are then mapped back to the height map constructed with the new reference plane, with the height from \mathbf{P} overwriting those from \mathbf{X} . For those points on the height map that do not correspond to any of the extracted coordinates, their heights are interpolated via a linear tensioning scheme described in Dressler [10].

2.3 Initialization of surface evolution algorithm

The proposed method can be initialized either with a given lobe segmentation or with user input points alone. Initialization of the proposed method with a given lobe segmentation is straight forward.

For initialization with only a set of user input points \mathbf{P} , we first estimate the 2D reference plane for representing the surface via the mean and the covariance of \mathbf{P} , similar to Section 2.2. The points in \mathbf{P} are then projected onto the newly formed surface. Using the projected points as seed, their heights are propagated to neighboring points, where a new height for each of these points is obtained by searching within a neighborhood of 10 voxels for the nearest point that forms a maxima in terms of the fissure confidence. The same height propagation and searching process is then repeated for the neighboring points and so on until the heights of all points of the surface are obtained. The resulting surface is then used for initializing the surface evolution algorithm in Section 2.2.

3 Experiments and results

Training of the supervised fissure enhancement filter and tuning of the parameters for the proposed method was performed using a total of 18 chest CT scans obtained from a research database, with slice thickness and in-plane resolution ranging from 0.50 to 1.25 mm and 0.55 to 0.78 mm respectively. The method was tested on chest CT scans from another research database, consisting of 22 scans from different patients with either idiopathic pulmonary fibrosis or severe emphysema (forced expiratory volume in one second $< 45\%$). The slice thickness and in-plane resolution of the test set ranged from 0.6 to 3.0 mm and 0.59 to 0.80 mm respectively.

As a preprocessing step, the lungs of all test scans were segmented using the method presented in Brown et al [11]. Approximate nearest neighbor searching [12] was used to implement the KNN classifiers of the supervised fissure enhancement filter, where the number of nearest neighbor K was set to fifteen [9] and an error bound ϵ of 0.5 was used. To reduce computational time, the multiple scales used for computing the confidence and performing the evolution process were implemented in the form of a Gaussian pyramid. The probability image from the supervised fissure enhancement filter was first filtered with a Gaussian kernel of $\sigma = 1$ voxel, resulting in the image at scale level one. By filtering the image with the same Gaussian kernel and subsampling the resulting image at a factor of two, the image at the second scale level is obtained. The same process is repeated to obtain images at higher scale level. A total of four scale levels were used in this work. The finite difference method was used to approximate Equation 1 using the images from the Gaussian pyramid.

To further reduce the computation time, the surface evolution algorithm was modified to stop at the second finest level instead of the finest level. A simple stopping criterion of stopping after N iterations was used for the evolution process. The value N was set to $10n_s$, where $n_s = 2, 3, 4$ indicates scale level in the evolution process, with $n_s = 2$ being the second finest scale. The number of smoothing iteration for the smoothing process was set to 100 for $n_s = 4$, and was divided by two whenever n_s is decreased (proceeding to finer scale). The minimum weight ω_{\min} and maximum weight ω_{\max} were set to 0.01 and 0.3 respectively.

Semi-automated segmentation of the lobes using the proposed method were performed by the first author and approved by a board certified radiologist. Initialization from scratch was used when a fissure detected from the automated segmentation were too different from the truth. User input points required by the proposed method were provided by drawing lines on either the axial, coronal or sagittal viewing plane. Table 1 shows the number of interactions required for the test cases, measured in the number of line segments drawn. Excluding the computation time of the supervised enhancement filter, segmenting the lobes interactively for a test case takes on average 20 minutes (including loading and processing), depending on how easy it is to locate the fissures visually. Figure 1 shows surface rendering of the segmented lobes and the line segments provided for the fissures in a left and a right lung.

Table 1. The number of line segments drawn by the user.

	Left major fissure	Right major fissure	Right minor fissure	All fissures
Average	13.5	13.7	9.5	36.8
Standard deviation	5.0	4.8	4.1	7.8
Min	3	5	3	18
Max	20	25	19	47

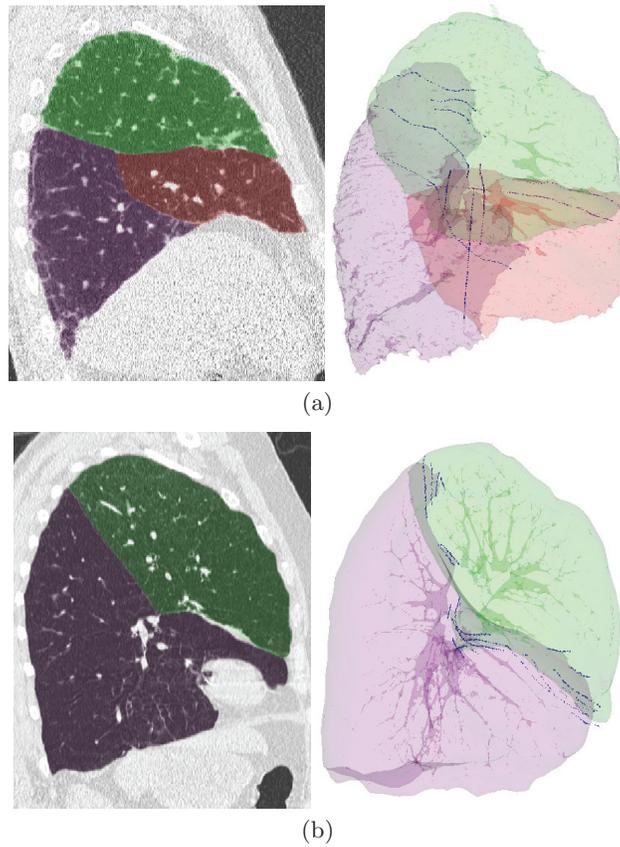


Fig. 1. Original CT scan overlaid with the segmented lobes (left), and the surface rendering of the segmented lobes and user input points, with the upper lobe in green, lower lobe in purple, middle lobe in red, input points for major fissure in blue and input points for minor fissure in pink.

4 Discussion and conclusion

We have presented a generic approach that is suitable for segmenting lung lobes both semi-automatically and automatically (as shown in previous work [8]). The advantage of the proposed method as a tool for semi-automatic lobe segmentation is that it is capable of taking into account user inputs and image appearance, and thus reducing the number of user inputted points, as seen from the Table 1 where it can take as low as three and no more than 25 line segments to get an accurate segmentation of a fissure.

It should be noted that the majority of these line segments are very short, less than half the length of the fissures in the view plane they are in, as observed in Figure 1. Also, most of these short segments are meant to correct errors that are less than one centimeter from the actual fissure, and thus they may be omitted

if the task at hand does not require accurate lobe segmentation, e.g. lobe volume analysis.

We did not notice a large difference in the number of required line segments between starting from scratch or correcting lobe segmented from an automated method. For the test cases where we start from scratch, two lines in two different view planes are all that is needed to obtain a fairly accurate fissure.

Future work will be to improve runtime performance of the proposed method, as currently it takes about 5 to 7 seconds to execute the evolution algorithm after user points are changed. Among the possible solution would be to precompute the fissure confidence instead of computing it on the fly. Also, the way the proposed method is currently implemented, there is a loading and preprocessing time for each fissure before the user can start the editing it. Taking advantage of multi-threading technology by moving certain processing into background will remove such wait time and thus further shorten user interaction time.

References

1. Coxson, H.O., Nasute Fauerbach, P.V., Storness-Bliss, C., Mller, N.L., Cogswell, S., Dillard, D.H., Finger, C.L., Springmeyer, S.C.: Computed tomography assessment of lung volume changes after bronchial valve treatment. *European Respiratory Journal* **32**(6) (Dec 2008) 1443–1450
2. Sciruba, F.C., Ernst, A., Herth, F.J.F., Strange, C., Criner, G.J., Marquette, C.H., Kovitz, K.L., Chiacchierini, R.P., Goldin, J., McLennan, G., Group, V.E.N.T.S.R.: A randomized study of endobronchial valves for advanced emphysema. *N Engl J Med* **363**(13) (Sep 2010) 1233–1244
3. Pu, J., Zheng, B., Leader, J.K., Fuhrman, C., Knollmann, F., Klym, A., Gur, D.: Pulmonary lobe segmentation in CT examinations using implicit surface fitting. *IEEE Transactions on Medical Imaging* **28**(12) (Dec 2009) 1986–1996
4. Ross, J.C., San José Estépar, R., Kindlmann, G., Daz, A., Westin, C.F., Silverman, E.K., Washko, G.R.: Automatic lung lobe segmentation using particles, thin plate splines, and maximum a posteriori estimation. *Medical Image Computing and Computer Assisted Intervention* **13**(Pt 3) (2010) 163–171
5. van Rikxoort, E.M., Prokop, M., de Hoop, B., Viergever, M.A., Pluim, J., van Ginneken, B.: Automatic segmentation of pulmonary lobes robust against incomplete fissures. *IEEE Transactions on Medical Imaging* **29**(6) (2010) 1286–1296
6. Lassen, B., Kuhnigk, J.M., Schmidt, M., S, S.K., Peitgen, H.O.: Lung and lung lobe segmentation methods at Fraunhofer MEVIS. In: *Proc. Fourth International Workshop on Pulmonary Image Analysis*. (2011) 185–199
7. Ross, J.C., San José Estépar, R., Díaz, A., Westin, C.F., Kikinis, R., Silverman, E.K., Washko, G.R.: Lung extraction, lobe segmentation and hierarchical region assessment for quantitative analysis on high resolution computed tomography images. *Med Image Comput Comput Assist Interv* **12**(Pt 2) (2009) 690–698
8. Lo, P., van Rikxoort, E.M., Abtin, F., Ahmad, S., Ordoorkhani, A., Goldin, J., Brown, M.S.: Automated segmentation of pulmonary lobes in chest CT scans using evolving surfaces. In: *Proc. SPIE 8669, Medical Imaging 2013: Image Processing*. (March 2013) 86693R–86693R–7
9. van Rikxoort, E., van Ginneken, B., Klik, M., Prokop, M.: Supervised enhancement filters: application to fissure detection in chest CT scans. *IEEE Transactions on Medical Imaging* **27**(1) (2008) 1–10

10. Dressler, M.: Art of Surface Interpolation. PhD thesis (2009)
11. Brown, M.S., McNitt-Gray, M.F., Mankovich, N.J., Goldin, J.G., Hiller, J., Wilson, L.S., Aberie, D.R.: Method for segmenting chest CT image data using an anatomical model: preliminary results. *IEEE Transactions on Medical Imaging* **16**(6) (1997) 828–839
12. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM* **45**(6) (1998) 891–923