IET **Journals**

The Institution of
Engineering and Technology

BronchoX: Bronchoscopy Exploration Software for Biopsy Intervention Planning

Esmitt Ramírez[1], Carles Sánchez[1], Agnés Borràs[1], Marta Diez-Ferrer[2], Antoni Rosell[2], Debora Gil[1,3]*

[1] *Computer Vision Center, Autonomous University of Barcelona, Bellaterra 08193, Spain*
[2] *Bellvitge University Hospital, L'Hospitalet de Llobregat, Barcelona 08907, Spain*
[3] *Serra Húnter fellow*
* *E-mail: esmitt.ramirez@cvc.uab.es*

Abstract: Virtual Bronchoscopy (VB) is a non-invasive exploration tool for intervention planning and navigation of possible pulmonary lesions (PLs). A VB software involves the location of a PL and the calculation of a route, starting from the trachea, to reach it. The selection of a VB software might be a complex process, and there is no consensus in the community of medical software developers in which is the best-suited system to use or framework to choose. This paper presents BronchoX (**Broncho**scopy E**X**ploration), a VB software to plan biopsy interventions that generate physician-readable instructions to reach the pulmonary lesions. Our solution is open-source, multiplatform and extensible for future functionalities, designed by our multidisciplinary research and development group. BronchoX is a compound of different algorithms for segmentation, visualization, and navigation of the respiratory tract. Performed results are focus on test the effectiveness of our proposal as an exploration software, also to measure its accuracy as a guiding system to reach pulmonary lesions. Then, 40 different virtual planning paths were created to guide physicians until distal bronchioles. These results provide a functional software for bronchoscopy exploration and demonstrate how following simple instructions is possible to reach distal lesions from the trachea.

1 Introduction

Bronchoscopy is an endoscopic technique that allows to visualize the inside of lung airways by inserting an instrument with a camera (i.e. bronchoscope) through the nose or mouth. Bronchoscopy examinations allow biopsy of pulmonary nodules with minimum risk for patients. A main constraint of flexible bronchoscopy is the difficulty to determine the best pathway to reach peripheral lesions.

Virtual Bronchoscopy (VB) [1] is a non-invasive imaging-based procedure to examine airways, and it is used to plan biopsy interventions. VB inspects lungs using a CT volume of the patient, and it allows planning the procedure to be followed during intervention. Biopsy planning involves both the location of the peripheral pulmonary lesion and identification of the path across airways to reach it. This path would be the best suited to perform the medical procedure using the available instrumentation. In the surgery room, the intention is to follow the planning previously carried out to locate the pulmonary lesion, preferably following friendly instructions. This process can be supported by a computer assisted software, which is a useful tool to help physicians.

Currently, there are some commercial options for VB planning and visualization, such Archimedes Planner/System and LungPoint Planner/VBN, both developed by Bronchus (http://www.broncus.com), or Spin System, developed by Medical Veran Technologies (http://www.veranmedical.com). The major drawbacks for adopting these systems are their set up and establishment process into clinical environments, and their expensive licenses. Therefore, exploring the available open-source medical software to be used as VB planner would be very useful.

The development process of an assistance software for bronchoscopists requires the study of several options, to choose the proper medical framework that best fits the clinical requirements. However, there is not a consensus about which software should be used for a VB, since it involves diverse tasks that include matching of planning with intervention videos, nodule segmentation and location, 2D/3D visualization, and others.

This kind of software might be developed using an existing framework by adding new functionalities (as modules or extensions) or not. Undoubtedly, using a framework involves considering the steep learning curve to develop applications, the portability for different platforms, the available documentation, and the extensibility to develop new features over it. These aspects may create doubts in which is the adequate framework for a particular need, or how to implement a particular algorithm in a chosen framework.

Bearing this in mind, in this paper we present an open source, multi-platform (Windows, Mac, and Linux) planning software for VB. The planning software allows the visualization and navigation of pulmonary airways. It performs the segmentation of the airways and its codification as a binary tree. Moreover, it allows building a path until a pulmonary lesion, starting from the trachea, through the segmented airways following an airways centerline. This path is built using a virtual fly-through camera, projecting images at all bifurcation points. This image-based approach allows the generation of instructions to be used during bronchoscopy as a roadmap.

2 Medical Imaging Frameworks

There are plenty of image processing and visualization frameworks to develop medical applications. In this research, we only focus in the open-source medical imaging libraries, APIs or frameworks extendable by plugins or modules. In general, a medical imaging software intend should be used by a multidisciplinary team, integrated by physicians and scientists. Remarkable examples are SCIRun [2], Medical Imaging Interaction Toolkit (MITK - http://mitk.org), VolView (www.kitware.com/volview), Matlab (https://mathworks.com) and MeVisLab (www.mevislab.de). These have been used as basis for several developments in the medical imaging field.

Similarly, there are medical imaging packages that cover tasks such as image processing and analysis (ITK - www.itk.org), visualization (VTK - www.vtk.org), tracking and related tasks in image-guided surgery (IGSTK [3]), real-time image and video processing (OpenCV - https://opencv.org), or real-time simulations (SOFA - www.sofa-framework.org). These packages work on different levels having their strengths and weakness, being used for general
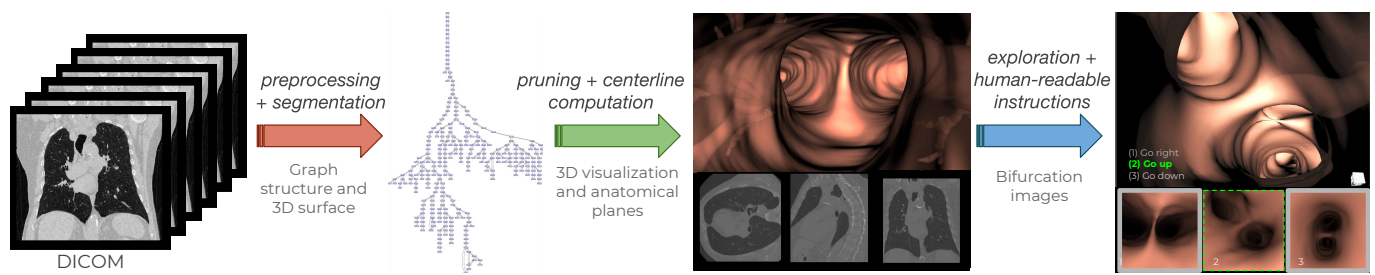
Fig. 1: Pipeline of the virtual bronchoscopy planning system. Using patient DICOM data, a graph structure is computed from a segmentation of airways. Graph allows the encoding of any path across airways as a list of instructions at each branching point. Path instructions are visually rendered as a mosaic of the whole route for instruction guiding.

or specific purposes in a particular task. Further studies of image processing tools for medical application can be found in [4, 5].

Other noticeable software platforms for medical imaging are 3D Slicer[6] and OsiriX [7]. 3D Slicer is an open-source software for medical imaging and 3D visualization built over two decades ago. Osirix is a suite of medical image processing and visualization software, supporting the DICOM standard. Both are widely used in medical community offering an easy integration with other hardware/software modules . They allow developing more complex and personalized applications.

Higgins [8] argues that VB applications should offer the navigation and visualization of pulmonary structures. Also, these applications are might be valuable as supporting tools for physicians, and applications would offer improvements for the clinical practice: positional information of lesions, projections and different views of structures, and being capable to extract data to be analyzed later.

The considerations suggested by Higgins represent a foundation for some studies in the development of VB solutions. In 2017 Nardelli et al. [9] use an architecture integrating different libraries/APIs: an electromagnetic tracking system handled with Matlab, and a video grabber connected to 3D Slicer using PLUS (Public Software Library for Ultrasound Imaging) [10]; both components joined with the OpenIGTLink (http://openigtlink.org) library. Also, Python and C++ programming language were used to write its procedures, and ITK for the centerline extraction. Another example is the usage of the OsiriX software, where Fiorelli et al. [11] present a virtual bronchoscopy tool to improve the accuracy of trans-bronchial needle aspiration for mediastinal staging, running in a tablet device.

Alternatively, some VB solutions use platforms built from scratch to ensure an optimal fulfilling of their requirements. Namely, CustusX [12] is a navigation system for image-guided intervention developed by the Norwegian National Competence Centre for Ultrasound and Image-Guided Therapy (http://usigt.org). Since 2003, CustusX has been used by clinical and technological researchers to medical imaging and navigation. It could be used as a navigation system or as a toolkit to develop new applications. In particular, Jens et al. [13] present a visualization method for navigated bronchoscopy with CustusX, including a technique called ACCuSurf (Anchored to Centerline Curved Surface) using a curved 3D surface from the trachea through the smaller airways.

Although using medical frameworks are powerful choices to start in the development from prototypes until in-vivo surgical applications, they are complex and often difficult to learn. In fact, Johnson [14] reported that frameworks require a good documentation and longer training than other options. Also, frameworks might be hard to develop, requiring better programmers than regular application developers.

Accordingly, the development of a VB functionality without using a framework is an option to consider. Numerous attempts have been made to develop algorithms for registration, segmentation and planning [15–17], mostly using C++ as core programming language and OpenGL for the visualization. Nevertheless, one of the main drawbacks to adopt these systems is the lack of information on how to use them. In 2007, an image-guided bronchoscopy for all planning stages was developed [18], and it was used during surgery to improve the procedure success rate. However, this solution is not shown as

reproducible. Similarly, other interesting proposals [19, 20] do not report the details of their implementations.

A few programming languages like C or C++, offer performance and speed and they can nearly interface with other languages. Nevertheless, the development of medical applications using this class of language might involve a large developing time. Then, developing a short-time software prototype without sacrificing the performance, emerges as an excellent option to develop.

The next section presents our proposal to create a planning software for bronchoscopy, using particular algorithms developed for our multidisciplinary group; covering task such as lung lesion selection, visualization, airway segmentation, and simple guiding instructions for physicians. Section 4 shows our results and discussions. Finally, Section 5 concludes the paper and future work of our research.

3 Bronchoscopy Planning Software

BronchoX (**Bronco**scopy **EX**ploration) is our proposal for a biopsy intervention planning software, with the intention to be used prior the bronchoscopy. BronchoX integrates different aspects to consider previous a surgical intervention, being a multiplatform and computer-efficient aided tool.

The software allows loading, visualizing and processing CT volumes in DICOM format. Pulmonary airways are segmented, and they are encoded in a 3D geometry using a tree-based structure (see Section 3.2). In addition, BronchoX allows to physicians to navigate interactively for the CT anatomical planes (axial, coronal and sagittal), and select the pulmonary lesion (interest point). Once a target point is selected, BronchoX constructs a path to the closest bronchi or bronchiole, (see Section 3.3) and generates a set of human-readable instructions to be used during the intervention (more details in Section 3.4).

Figure 1 summarizes the pipeline of BronchoX. First, the user selects the corresponding study for a patient in DICOM format (CT volume). Next, the pre-processing and airway segmentation algorithms are executed generating a 3D geometrical mesh of airways. This mesh is encoded into a rooted binary tree which represents the airways bronchi geometry set by the structure of the segmentation centerline. Thus, the visualization of the 3D geometry and 2D anatomical slices are available. Finally, the instructions-based visual roadmap from the trachea to the pulmonary lesion is showed to the user and exported for being used during an intervention.

3.1 Architecture

The layer-based architecture uses a set of well-known stable libraries in order to have a simple and reproducible software pipeline, to develop algorithms for VB planning and guiding. The design of BronchoX is based on C++ programming language, which is used for its efficiency to handle computational resources.

We opted for QT to develop the GUI, which will be directly used by physicians . This selection does not entail implications in the object-oriented programming paradigm. The GUI is fluid and intuitive for physicians offering icons as visual elements. Also, Qt

contains cross-platform components to create compliant software ensuring patient safety.

The Figure 2 shows the architecture, where C++ programming language and QT framework are above the host operating system. The libraries OpenCV, ITK and VTK are on the top of BronchoX implementation performing the rendering and processing of the 3D objects.
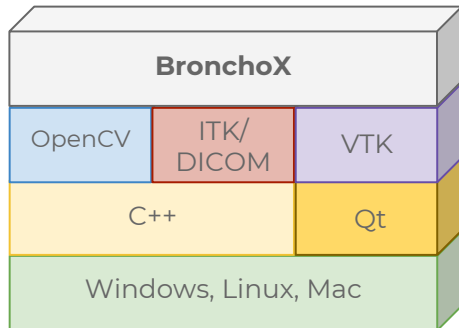

Fig. 2: Layer-based architecture proposed in BronchoX.

The proposed architecture allows BronchoX to function whether as a high-level library for developers or as a programming language directly. In fact, it is possible to incorporate algorithms or libraries with the C++ integration capability (e.g. Matlab or Python scripts). The combination of algorithms offered by VTK, ITK and OpenCV are favourable to implement particular functionalities for bronchoscopy: 3D reconstruction, DICOM handling, image filtering, and others. Further, OpenCV provides modern features such as augmented reality and deep learning.

3.2 Anatomy Segmentation & Representation

Our segmentation strategy is based on the thresholding of bronchi local appearance maps computed convolving CT volumes with own-designed tubular filters, which are define using second derivatives of Gaussian kernels. In order to account for difference in caliber and appearance between main and distal airways, we adopt a multiresolution approach.

The segmented anatomy is encoded to considering any possible path across airways. Airways are tubular structures and their geometry is determined by the centerline given by bronchi lumen center. These centerlines have a tree structure given by bronchi branching levels. Then, to quantify the anatomical consistency with segmentations, we analyze the geometry of their skeleton. To do so, the segmentation skeleton is encoded in a graph that represents its branching geometry by nodes and edges. The nodes of the graph correspond to the skeleton branching points and its edges represent branch connectivity. The trachea entry point allows directing the graph using the Depth First Search (DFS) algorithm. The directed graph is a (binary) tree where levels correspond to bronchial levels, and leafs correspond to the most distal points achieved by the segmentation.

The directed graph of the final segmentation skeleton is encoded using two adjacency matrices: one binary matrix defining node tree connectivity, and one matrix of 3D segments that keeps the list of 3D skeleton points that connect each pair of adjacent nodes. This scheme is flexible and allows us to develop a matching between a CT-video bronchial structure and the coded airways using anatomical landmarks [21].

3.3 Path Encoding

Each level of the binary tree represents a bronchial depth d. A navigation path is defined as a sequence of segments $S_1, S_2, \ldots, S_n$, connecting a leaf node in S_n to the trachea (root node). Each segment S_d contains the sequence of skeleton points, $p_1^d, p_2^d, \ldots, p_{kd}^d$,

that joins two bifurcation points at consecutive levels $d-1$ and d, being p_{kd}^d the bronchial bifurcation point at level d.

For encoding the path, each segment is labeled according to its 2D position in the projected images. These positions are obtained by the simulation of virtual guiding using CT scans of patients. As illustrated in Figure 3, for each bifurcation point p_{kd}^d along a path, a virtual camera is located with the viewing direction to its children segments, specifically σ_1 and σ_2. The skeleton points of σ_1, σ_2 are projected from that point of view to label σ_1, σ_2 according to its position in the projected image.
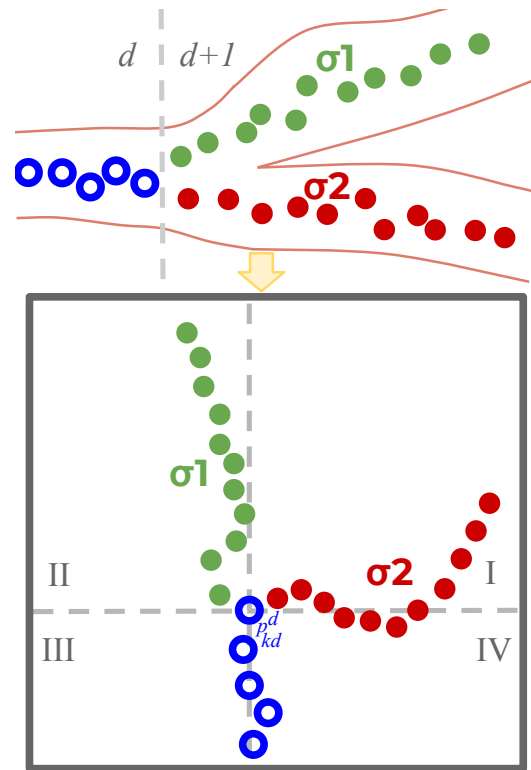

Fig. 3: Codification of a path, using the 2D position of airways segments in the projection.

Each project image is split into four quadrants (labeled I, II, III and IV) centered at the projection of p_{kd}^d. Each projected point belongs to a quadrant, and σ_1, σ_2 are assigned to a quadrant to which their points belong to most frequently.

Figure 3 illustrates the segment quadrant assignment. The gray dotted lines show the quadrant division and σ_1, σ_2 points are plot in different colors. We observe that σ_1 has points in quadrant I and IV, but it is assigned to quadrant I which is where majority of σ_1 points belong.

3.4 Navigation Instructions

The classification of each segment allows constructing a set of ordered instructions using the quadrants of the segments, connecting the trachea to a peripheral lesion. The based-quadrant instructions are described by their position in images: I is named as **up-right**, II as **up-left**, III as **down-left**, and IV as **down-right**. The labels of segments given by their quadrant are further simplified: the common word in the projected segments at each bifurcation is excluded. For instance, if two segments are in quadrants I/II and III/IV, the instruction whether is **go up** or **go down** respectively; if they are in quadrants I/IV and II/III, the instruction whether is **go right** or **go down**. If two segments are in the same quadrant, we order them by their angle inside the quadrant they belong, after, the same approach to get a single instruction is applied.

The instructions are visually presented to physicians as a mosaic of images representing each bifurcation point. Consequently, physicians can navigate through each one using the provided GUI of BronchoX, as shown in Figure 4. Each bifurcation is enumerated according its bronchial level, presenting the navigation instructions in a simple way.
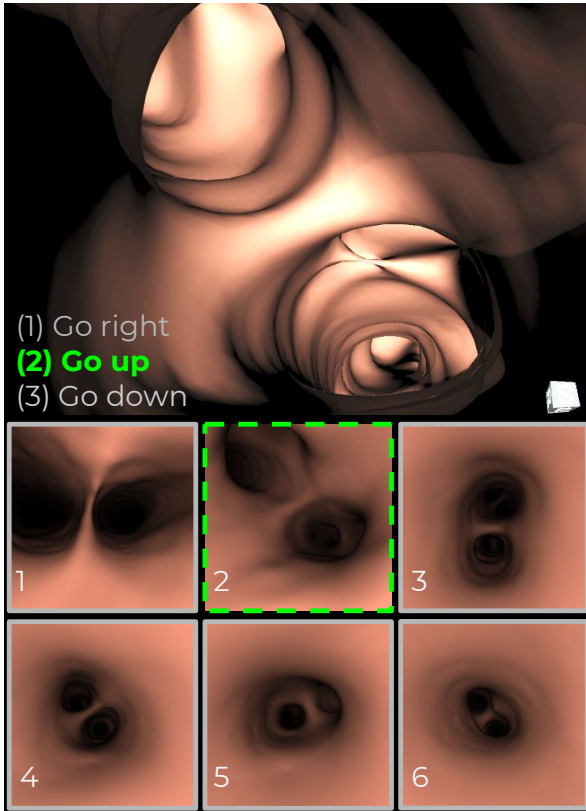

Fig. 4: Instructions presented to physicians as a mosaic of images.

4 Results and Discussion

BronchoX is developed using open-source available software packages, also the user interface is built using Qt forms, and the 3D airway model is an OBJ geometrical file. About the 3D model, this is visualized with 70% transparency using the dual depth peeling algorithm [22], with an occlusion ratio = 0.1, and a maximum of 40 layers. The code is available as open source at https://gitlab.com/esmitt/BronchoX, where the latest version can be found and where all developments can be followed.

The segmentation was applied to the 40 CT scans data of the MICCAI Challenge EXACT'09 [23], acquired in different conditions including variable slice thickness (0.5 - 1.0 mm), in-plane voxel sizes (0.55-0.78 mm) and radiation dose (120/140 kVp, 10.0 - 411.5 mAs). Evaluation on EXACT'09 cases shows that our method achieves competitive performance.

From a technical point of view, is a self-contained software. It could execute scripts written in Matlab, which are used to perform the segmentation process. Also, the VTK, ITK, and QT libraries are integrated into the same code, offering a CMake configuration to be compiled in different configurations (i.e. different compilers and operating system).

In order to validate the reliability of our proposal, physicians verified the effectiveness of a set of given instructions to reach distal bronchioles. For this, we generated different virtual explorations using CT volumes of 10 anonymized patients. CT scans were acquired with a 320-detector row Aquilion ONE, Toshiba CT

scan, and a sample thickness of 0.5 mm. For each patient, four virtual explorations covering four lobes were defined: left and right upper lobes (LUL and RUL), and left and right lower lobes (LLL and RLL). For each path, a sequence of instructions was validated by three experts, trying to reproduce the path using the given instructions. The generated paths are between the sixth and twelfth bronchial level.

We defined a false instruction rate (FIR) variable, representing when an expert could not reproduce the given path. This data is modeled as a mixed model using R, version 3.2.5. A Poisson model was adjusted to include the segmental lobe as factor ($Lobe$), and a random subject effect (Pat) to consider the intra-individual variability among cases, and a random effect to model inter-observer variability as:

$$log(FIR_{ijk}) = \beta_0 + \beta_1 Lobe + Pat_i + Obs_j + \epsilon_{ijk}$$

where $Pat_i \sim N(0, \sigma_{Pat})$ denotes the random effect that models intra-patient variability, $Obs_j \sim N(0, \sigma_{Obs})$ the random effect for inter-observer variability and the factor $Lobe$ with values RLL, LLL, RUL, LUL. Model assumptions were validated by means of residual analysis and influential values. Also, model coefficients, p values and 95% confidence interval (CI) for significance in main effects were computed. The CIs were back transformed to the original scale for their interpretation. Thus, a p value < 0.05 was considered statistically significant.

Descriptive statistics (average and standard deviation, SD) and model adjustment for FIR are shown in Table 1, in a percentage way. The increase in FIR for the lower left lung lobe is mostly due for confusing instructions at the third generation, just after the LUL-LLL branching point. Although the 3D geometry around the 3rd generation presents two branching points (i.e. two levels), they are not appreciated in the projected images due to a short distance between them.

Table 1 False instruction rate (FIR) values for the 4 lobes (descriptive & model)

% FIR	descriptive		model		
	mean	**SD**	**coeff**	**p-val**	**CI**
RLL	2.8	4.6	1	-	(0.6, 5.2)
LLL	13	11.7	1.3	< 0.01	(5.2, 20.6)
RUL	1.7	4.0	-0.35	0.06	(0.1, 3.4)
LUL	6.4	8.7	0.64	0.08	(2.1, 10.5)

In projected images, the LLL lumen is hard to identify visually, then three airway lumens that correspond to the projection of LLL next generation are visible. Therefore, from the point of view of the operator (i.e bronchoscopist), there are three possible airways to follow in the same level, whereas in our codification consist of two consecutive levels with two airways each one. Figure 5 illustrates this phenomenon, where three lumens appear in the projection for a particular bronchial level.

The upper part of figure 5 corresponds with a captured view at depth level $d = 3$, and the lower part when $d = 4$ for one example path. In the same projection, there are both a child from level 3 (blue - dotted ellipse), and children of level 4 (purple - continuous ellipse). This might generate confusing instructions to be followed by operators. This occurrence happens at short distance between a bifurcation point and the segment's children of the next level.

5 Conclusions

An exploration bronchoscopy software for biopsy intervention planning and navigation is introduced in this paper. The developed VB software, named BronchoX, has an architecture designed to create personalized applications; these applications could be implemented as supporting software in the planning stage and subsequently on surgery stage. Similarly, it is possible to add new features over the C++/Qt layer to improve the solution. For example, the addition of
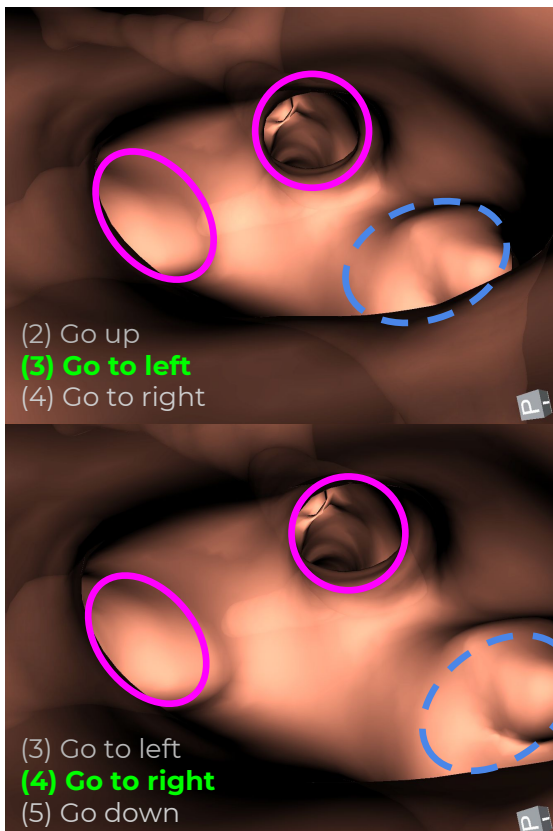
Fig. 5: Example of the phenomenon, when three lumens appear in a projection (third bronchial level).

the Pulse Physiology Engine* might simulate the comprehensive patient physiology in the respiratory system. Besides, the integration of external devices or libraries as the Intel Real Sense† or any AR/VR library is possible.

BronchoX has many interesting functionalities for VB navigation. It is possible to identify a lesion using the anatomical planes and to compute the path to a lesion's closest point. The path is encoded as a sequence of instructions at each airway bifurcation to traverse. Instructions are given in a natural language, allowing the easy identification of the bronchi that physicians should follow. This feature offers a great advantage in order to follow the planned roadmap.

The results based on mixed models are very promising, they provide a quantitative analysis of performance considering both population and anatomical factors. The statistical analysis gives a bias in instructions for the left lower lobe, being introduced by the close spatially consecutive levels. As a result, these levels are visualized as a single level with three lumens, when certainly is composed by two levels. Then, unifying those levels into one single instruction instead of two, seems as a practical solution.

Future work will concentrate on introducing a hands-free guiding system to be used into operating room, to guide during in-vivo explorations. This might be realized with a specific hardware, also it will be integrated as part of our layer-based architecture. Besides, we want to integrate the video-bronchoscopy landmarks [24] into BronchoX to obtain a synchronization between the virtual and in-vivo video patient traversal.

We have confidence that our work might be a starting point for supporting researchers into development of new algorithms using our open-source solution. BronchoX is the first step in the development of a bronchoscopy navigation and exploration system for physicians, and it will be as foundation to develop further studies in virtual bronchoscopy.

6 Acknowledgments

7 Funding and Declaration of Interests

8 References

1 De Wever, W. and Bogaert, J. and Verschakelen, J. A. Virtual Bronchoscopy: Accuracy and Usefulness—An Overview. *Seminars in Ultrasound, CT and MRI*, 26(5):364 – 373, 2005. Update of Chest Imaging-Part I.

2 Steven G. Parker and Christopher R. Johnson. Scirun: A scientific programming environment for computational steering. In *Proceedings of the 1995 ACM/IEEE Conference on Supercomputing*, Supercomputing '95, New York, NY, USA, 1995. ACM.

3 Andinet Enquobahrie, Patrick Cheng, Kevin Gary, Luis Ibanez, David Gobbi, Frank Lindseth, Ziv Yaniv, Stephen Aylward, Julien Jomier, and Kevin Cleary. The image-guided surgery toolkit igstk: An open source c++ software toolkit. *Journal of Digital Imaging*, 20 Suppl 1:21–33, 12 2007.

4 Lay-Khoon Lee and Siau-Chuin Liew. A Survey of Medical Image Processing Tools . In *4th International Conference on Software Engineering and Computer Systems (ICSECS)*, pages 171–176. IEEE, 2015.

5 Garima Sharma. A Study of Open Source Toolkits of Image Processing for Healthcare Industry. *Global Journal of Researches in Engineering*, 17:27–33, 2017.

6 Ron Kikinis, Steve D. Pieper, and Kirby G. Vosburgh. *3D Slicer: A Platform for Subject-Specific Image Analysis, Visualization, and Clinical Support*, pages 277–289. Springer New York, New York, NY, 2014.

7 Antoine Rosset, Luca Spadola, and Osman Ratib. OsiriX: An Open-Source Software for Navigating in Multidimensional DICOM Images. *Journal of Digital Imaging*, 17(3):205–216, September 2004.

8 W. E. Higgins, K. Ramaswamy, R. D. Swift, G. McLennan, and E. A. Hoffman. Virtual bronchoscopy for three–dimensional pulmonary image assessment: state of the art and future needs. *RadioGraphics*, 18(3):761–778, May 1998.

9 Pietro Nardelli, Alexander Jaeger, Conor O'Shea, Kashif A. Khan, Marcus P. Kennedy, and Pádraig Cantillon-Murphy. Pre-clinical validation of virtual bronchoscopy using 3D Slicer. *International Journal of Computer Assisted Radiology and Surgery*, 12(1):25–38, June 2017.

10 A. Lasso, T. Heffter, A. Rankin, C. Pinter, T. Ungi, and G. Fichtinger. Plus: Opensource toolkit for ultrasound-guided intervention systems. *IEEE Transactions on Biomedical Engineering*, pages 2527–2537, October 2014.

11 Alfonso Fiorelli, Antonio Raucci, Reginelli Alfonso Cascone, Roberto and, Davide Di Natale, Carlo Santoriello, Antonio Capuozzo, Roberto Grassi, Nicola Serra, Mario Polverino, and Mario Santini. Three-dimensional virtual bronchoscopy using a tablet computer to guide real-time transbronchial needle aspiration. *Interactive CardioVascular and Thoracic Surgery*, 24(4):567–575, April 2017.

12 Christian Askeland, Ole Vegard Solberg, Janne Beate Lervik Bakeng, Ingerid Reinertsen, Geir Arne Tangen, Erlend Fagertun Hofstad, Daniel Høyer Iversen, Cecilie Våpenstad, Tormod Selbekk, Thomas Langø, Toril A. Nagelhus Hernes, Håkon Olav Leira, Geirmund Unsgård, and Frank Lindseth. CustusX: an open-source research platform for image-guided therapy. *International Journal of Computer Assisted Radiology and Surgery*, 11(4):505–519, Apr 2016.

13 Pall Jens Reynisson, Erlend Fagertun Hofstad, Håkon Olav Leira, Christian Askeland, Thomas Langø, Hanne Sorger, Frank Lindseth, Tore Amundsen, and Toril Anita Nagelhus. A new visualization method for navigated bronchoscopy. *Minimally Invasive Therapy & Allied Technologies*, 27(2):119–126, 2018.

14 Ralph E. Johnson. Frameworks = (components + patterns). *Commun. ACM*, 40(10):39–42, October 1997.

15 Scott Merritt, Rahul Khare, Rebecca Bascom, and William E. Higgins. Interactive CT-Video Registration for the Continuous Guidance of Bronchoscopy. *IEEE Transactions On Medical Imaging*, 32(8):1376–1396, August 2013.

16 Jason D. Gibbs, Michael W. Graham, Rebecca Bascom, Duane C. Cornish, Rahul Khare, and William E. Higgins. Optimal Procedure Planning and Guidance System for Peripheral Bronchoscopy. *IEEE Transactions On Biomedical Engineering*, 61(3):638–657, March 2014.

---

17 Rahul Khare, Rebecca Bascom, and William E. Higgins. Hands-Free System for Bronchoscopy Planning and Guidance. *IEEE Transactions On Biomedical Engineering*, 62(12):2794–2811, December 2015.

18 J. P. Helferty, A. J. Sherbondy, A. P. Kiraly, and W. E. Higgins. (computer-based system for the virtual-endoscopic guidance of bronchoscopy.

19 Daniel Barnes, José Gutiérrez, Mariana Benegas Rosario, Teresa M. De Caralt, José Ramirez, Ivan Vollmer, and Marcelo Sanchez. Central airway pathology: clinic features, CT findings with pathologic and virtual endoscopy correlation. *Insights Imaging*, 8(2):255–270, December 2017.

20 Wilbert G. Aguilar, Vanessa Abad, Hugo Ruiz, Jenner Aguilar, and Fabián Aguilar-Castillo. RRT-Based Path Planning for Virtual Bronchoscopy Simulator. In Lucio Tommaso De Paolis, Patrick Bourdot, and Antonio Mongelli, editors, *Augmented Reality, Virtual Reality, and Computer Graphics*, pages 155–165, Cham, 2017. Springer International Publishing.

21 Carles Sánchez, Antonio Esteban-Lansaque, Agnes Borrás, Marta Diez-Ferrer, Antoni Rosell, and Debora Gil. Towards a videobronchoscopy localization system from airway centre tracking. In *VISIGRAPP (4: VISAPP)*, pages 352–359, 2017.

22 Louis Bavoil and Kevin Myers. Order Independent Transparency with Dual Depth Peeling. Technical report, Nvidia, 2008.

23 J.M. Reinhardt Y Tarunashree P.A. de Jong B Irving C Fetita M Ortner R Pinho J. Sijbers M Feuerstein A Fabijanska C Bauer R Beichel C.S. Mendoza R. Wiemker J. Lee A.P. Reeves S. Born O Weinheimer E.M. van Rikxoort J Tschirren K Mori B Odry D.P. Naidich I.J. Hartmann E.A. Hoffman M. Prokop J.H. Pedersen P. Lo, B. van Ginneken and M. de Bruijne. Extraction of airways from ct (exact09). *TMI*, 31:2093–2107, 2012.

24 Carles Sánchez, Debora Gil, and et al. Navigation path retrieval from videobronchoscopy using bronchial branches. In *CLIP, LNCS 9401*, 2015.