# Toward Reliable Multi-Generational Analysis of Anatomical Trees in 3D High-Resolution CT Images

K. C. Yu, E. L. Ritman, A. P. Kiraly, S. Y. Wan, M. Zamir, and W. E. Higgins

Penn State University, University Park, PA 16802 USA

[ER]Mayo Foundation, Rochester, MN 55905 USA

[SW]Chang Gung University, Taoyuan 333, Taiwan

[MZ]University of Western Ontario, London, Ontario, Canada N6A 5B7

## ABSTRACT

Modern micro-CT and multidetector helical CT scanners can produce high-resolution 3D digital images of various anatomical tree structures, such as the coronary or hepatic vasculature and the airway tree. The sheer size and complexity of these trees make it essentially impossible to define them interactively. Automatic approaches, using techniques such as image segmentation, thinning, and centerline definition, have been proposed for a few specific problems. None of these approaches, however, can guarantee extracting geometrically accurate multigenerational tree structures. This limits their utility for detailed quantitative analysis of a tree. This paper proposes an approach for accurately defining 3D trees depicted in large 3D CT images. Our approach utilizes a three-stage analysis paradigm: (1) Apply an automated technique to make a "first cut" at defining the tree. (2) Analyze the automatically defined tree to identify possible errors. (3) Use a series of interactive tools to examine and correct each of the identified errors. At the end of this analysis, in principle, a more useful tree will be defined. Our paper will present a preliminary description of this paradigm and give some early results with 3D micro-CT images.

**Keywords:** micro-CT, 3D imaging, arterial trees, vascular networks, data mining

## 1. INTRODUCTION

Modern micro-CT[1–3] and multidetector helical CT scanners[4–6] can produce high-resolution 3D digital images of various anatomical tree structures, such as the coronary or hepatic vasculature[7–13] and the airway tree.[14,15] The sheer size and complexity of these trees make it essentially impossible to define them interactively. Automatic approaches,[9,13] using techniques such as image segmentation, thinning, and centerline definition, have been proposed for a few specific problems. Also, some automated analysis work has only focused on a principal pathway,[16–18] where a principal pathway is designated as one selected path from the beginning of the tree (the "root") to the end of one terminating branch.

*While existing techniques give a high percentage of apparently correct branches, none of them can guarantee geometrically accurate multigenerational tree structures.* This limits their utility for detailed quantitative analysis of a tree. Many problems arise in the trees produced by previously proposed approaches[10,13]: (1) branches are missed; (2) branches break, creating overly short branches and forming new false branches; (3) extra spurious branches arise, causing false bifurcations; (4) anatomically implausible loops occur. See Figure 1. These problems result in a tree that has an incorrect geometrical structure. Also, errors arising in the first few generations cause errors to propagate to all other generations. Again, the size and complexity of the images involved make it difficult to quantify the extent of these difficulties. Also, some errors are caused by image-reconstruction artifacts and flaws in specimen preparation.

These observations lead to our basic philosophy in reaching a solution to this problem: (1) It is unrealistic and counterproductive to rely strictly on improved scanning technology and improved automated image-processing algorithms for defining an accurate tree; (2) automated techniques, despite their imperfections, are essential in providing a complete description of a tree; (3) judicious human interaction is essential for arriving at an accurate tree. Our proposed analysis approach, based on this philosophy, utilizes a three-stage approach:

1. Apply an automated technique for making a first iteration at defining the tree. This step will produce a segmented tree and an associated tree description.

Send correspondence to WEH; email: weh2@psu.edu

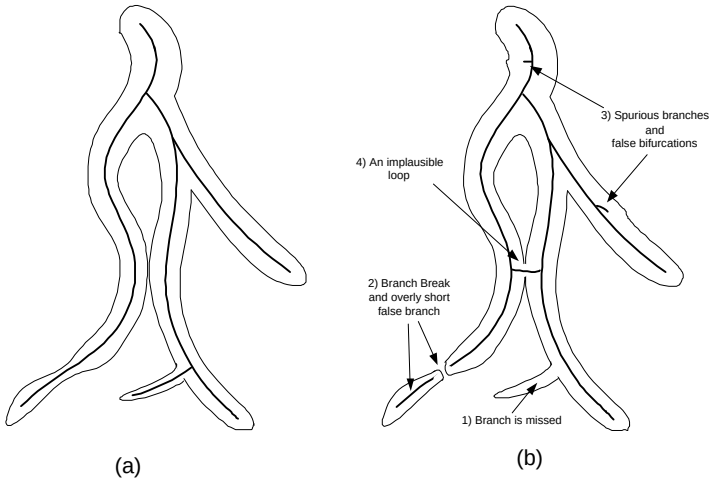



**Figure 1.** Schematic figure illustrating the problems that arise in defining 3D anatomical trees: (a) ideal tree; (b) typical tree defined by existing image-analysis techniques. Outer solid outline represents the segmented tree, while the interior solid lines represent the defined axial tree structure.

2. Analyze the automatically defined tree to identify possible errors, such as broken branches, short terminal branches, loops, overly close bifurcations, and trifurcations.
3. Use a series of interactive tools, summarized briefly below, to examine and correct each of the identified errors. At this point, other unidentified errors could also be fixed.

At the end of this analysis, in principle, a more accurate tree will be defined.

This paper describes our early efforts toward devising this approach. We focus primarily on 3D micro-CT images, but we believe many of our techniques have general applicability. Section 2 describes the approach in more detail, while Section 3 gives preliminary results. Finally, Section 4 points out that much remains to be done and highlights remaining issues.

## 2. METHODS

The discussion below gives more detail on the proposed three-stage tree analysis system. This effort builds on the previous efforts of Wan *et al.*[9,13] The software, built to run on a Microsoft Windows 2000 or XP platform, is written in Visual C++. It employs the Microsoft Foundation Class (MFC) library for basic graphical user interface development, the Visualization Toolkit (VTK) for 3D visualization, and OpenGL for additional graphics support (required by VTK).[19–21] In addition some high-level graphical interface components are incorporated from the Business Components Gallery (BCG), an MFC extension library that enhances user interface development in the Windows.[22] The remainder of this section describes the analysis procedure.

### 2.1. Stage 1: Define the Raw Tree

The first stage of processing involves automatically computing a raw uncorrected tree. Currently, we incorporate the automatic tree analysis method of Wan *et al.*[9,13] and the tree-refinement method of Kiraly.[23] Given a gray-scale 3D image, the method proceeds through four main processing stages: segmentation, skeletonization, tree refinement, and surface generation. The segmentation phase includes three operations: sigma filtering (to reduce noise), symmetric region growing (to define the initial raw tree), and cavity deletion (to fill interior holes in the segmentation that greatly affect the topology). During skeletonization, a 3D homotopic thinning algorithm is run to generate a single-layer thick axial structure of the segmented tree. Next, a tree refinement method is run that automatically removes some false branches and generates a smooth axial-tree structure. This axial structure represents the raw tree. A final surface generation step is run to create polygonal (triangle) surface representation data, suitable for rendering the segmented tree and axial structure. Consult the references for complete detail.

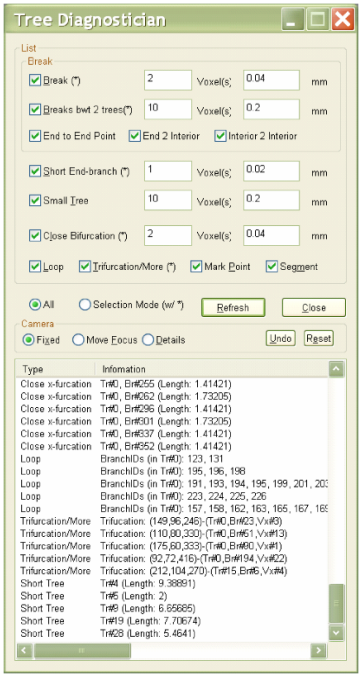



**Figure 2.** Example output for the Tree Diagnostician for image "control2." (See Section 3.) The top portion of the panel shows parameters used to identify defects. The bottom portion lists the possible tree defects satisfying the defect criteria. In the example, each pair of branch endpoints separated by 2 or fewer voxels is listed as break, branches of length 1 voxel are listed as short branches, bifurcations separated by 2 or fewer voxels are listed as close bifurcations, loops are identified, and branch points spawning 3 or more children (trifurcations/more) are listed.

### 2.2. Stage 2: Identify Possible Tree Errors

For Stage 2, a tool referred to as the Tree Diagnostician automatically identifies candidate tree defects, such as broken branches, loops, etc. This tool performs a series of straightforward analysis steps on the previously defined raw tree to identify the potential defects. For example, a candidate broken branch (a "break") can be defined as a location where two tree end points are fewer than two voxels apart. Figure 2 gives a complete example. The user can then cycle through each defect on the list and attempt to correct them, as discussed in the next subsection.

### 2.3. Stage 3: Examine and Correct Tree Errors

During Stage 3, the user examines each defect flagged by the Tree Diagnostician with a series of interactive computer-graphics tools. To interrogate and correct a particular defect, the user double clicks on the defect as listed in the Tree Diagnostician. The system then zooms in on the region of the candidate defect in a 3D Rendering Tool. Other graphical tools can also be invoked to study the geometry and image data about the defect and possibly correct it. The following tools have been built for this interrogation/correction process:

1. 3D Rendering Tool — Depicts a 3D rendering of the segmented tree and axial structure.
2. Locator Tools — Enable location of sites of interest.
3. Site Bounding Box — Provides a local view of data about a flagged defect.
4. Editing Tools — Provide a means for fixing and eradicating defects.
5. 2D Tree Map — Gives a 2D graph view of a tree.

Figure 3 gives an overview of these tools, while Figure 4 give a system composite.

The Locator Tools help mark a tree site in order, for example, to fix a broken branch. The Locator Tools are comprised of a Skeleton Picker (for picking existing tree sites), a 3D Site Locator, a 3D Cursor for moving about a defect region, and an Intersection Locator. The Skeleton Picker allows the user to pick and modify existing

branches. The 3D Site Locator draws upon two different angles of view to enable a graphical "camera" to locate a unique point in 3D space. The 3D Cursor allows the user to move about in 3D space and to confine attention to a focal plane, or parallel planes in $x$, $y$, or $z$. The Intersection Locator finds the center of intersection between a graphical line and skeleton details depicted in the segmented tree image (Figure 3c).

When focusing on a candidate defect site, the system provides a Bounding-Box mechanism for viewing data in the neighborhood of the site (Figure 3a). Movements can be made in transverse, sagittal, and coronal directions within the bounding box. Standard 2D slice images and projection images are then computed, corresponding to these movements. These views are useful for finding sites to connect broken branches or for deleting unwanted branches.

Among the Tree Editing tools are: 1) a Skeleton Editor, which enables point and connection editing (Figure 3b); 2) a Point Editor, which when used in conjunction with the Locator Tools, allows the addition or removal of specific skeletal points; and 3) a Connection Editor, which permits similar editing of two selected points (line segments can be deleted or added). Other Tree Editing Tools enable tree root selection and tree pruning.

Finally, a 2D Tree Map presents the 3D tree as a graphical 2D rendition (Figure 3d).[24] This view makes it much easier to focus on particular generations and sections of the tree, without being confounded by the geometrical complexity of 3D Viewing. It functions in a manner similar to Windows Explorer. Portions of the tree can be expanded or collapsed to facilitate viewing various parts of the tree. It can also display branch width and length information. The 2D Tree Tool also has a facility for selecting the tree root and for pruning. It can also be used to highlight branches in the other viewing tools.

Much of our philosophy in the tools developed here originate from the field of visual data mining.[24–27] Visual data mining techniques have shown value in exploring large complex databases. The basic idea of visual data mining is to present the data in some visual form which allows the user to get insight into the data, draw conclusions, and directly interact with the data. As we do for our system, visual data mining typically follows a three-step process[25]: Overview first, zoom and filter, and then details-on-demand. Figure 4 shows a partial montage of the system. The given view fits onto a 2048×1536 screen. In the labotatory, we generally use two monitors to give more tools available simultaneously.

## 3. RESULTS

### 3.1. Image Control2

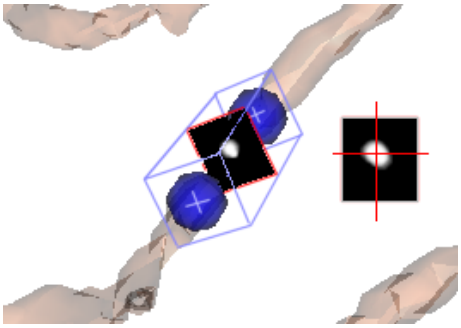
Our first test was with the 3D micro-CT image "control2," used in the earlier work of Wan *et al.*[9] Figure 5 (a) displays a raw 3D tree structure produced by automatic techniques. Figure 5 (b) shows the result after a serious editing to eliminate all possible errors. This is a difficult case with many defects identified by the Tree Diagnostician in the raw tree (Figure 2). Also, veins and arteries merge at points. Using the proposed tools, we fixed these merged branches. Also, five loops and four false trifurcations were removed. Finally, all close bifurcation-point regions were verified. Given these edits, we are now nearly perfectly confident that all branches within the first eight generations are correct. This effort took on the order of two hours of user interaction. We point out that this task is essentially impossible with any other methodology. One interesting result of this test is that the number of "apparent" generations increased from 14 to 25. This result brings up a further question on the proper definition of the concept of a generation. We further realize that it is unrealistic to get all generations depicted in an image correctly, as branches will be missing or other raw image defects will cause difficulties.

### 3.2. Image H61

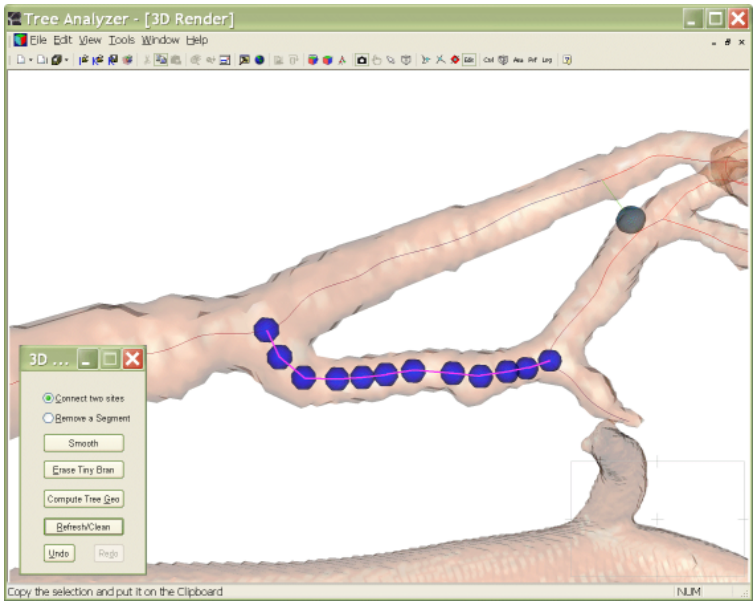
Figure 6 gives results for the micro-CT image H61. For this case, the Tree Diagnistician found 14 loops, 12 trifurcations, and one small disconnected tree. It took a human operator over two hours to remove these defects. Also, the large main tree and small disconnected tree were connected at the root. This step was necessary, since the original scanned specimen was embedded in clay. This caused the proximal portion of the tree to be obscured in the resulting scan, causing a break into two "separate" trees. Previously, a skilled human observer tabulated the existing branches for this case in the original cast. Our results match these human-observed results (the human observer found branches up to generation 11) to a fair degree, with the following issues:

1. The human observer delineated certain very small branches that did not show up in the image data. The computer could not find these. This then caused substantial propagating "mislabelings" of all other successive lower-generation branches in the computer's results.

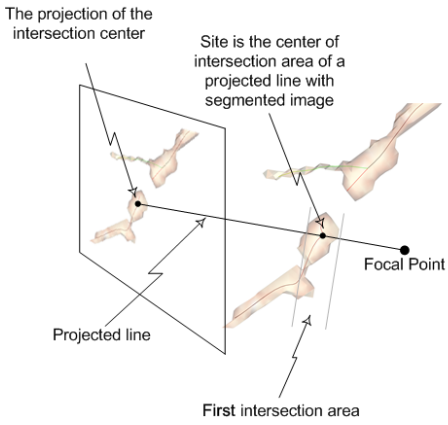
2. The computer identified extra branches at generations 12-15 that the human did not measure.



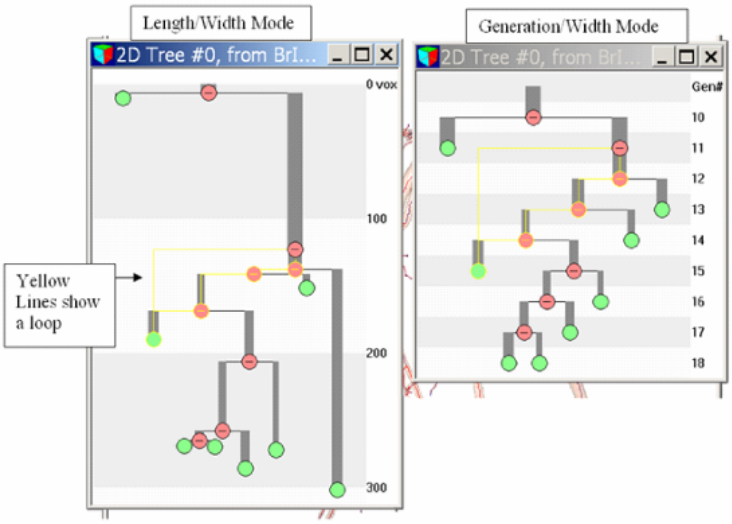
(a) Bounding box visualization




(b) Tree Editing tool




(c) 3D intersection analysis




(d) 2D tree map

**Figure 3.** Montage of interactive tools for analyzing a 3D tree. (a) 3D Bounding box focused on a site of interest. The local 2D oblique cross-section at the active site is shown. In this case, there is a gap between two branches (a branch break). The 2D slice views are used to specify connecting points. (b) The Tree Editing tool can be used in conjunction with the 3D Bounding Box to mark a series of axial (skeletal) points (small spheres). These points are then connected to fix a broken branch. (c) Depiction of the 3D geometry in the visualization tools and coordinating the visualization with user mouse interactions is complex. The example figure illustrates how 3D intersections are defined. (d) Two forms of the 2D Tree Map. The left window shows the lengths of branches in voxels with their width information. The right window displays its generation hierarchy structure. Both windows enable the width mode, and each branch shows its related width in the displayed branches. Colored lines in both windows show there is a loop in the depicted tree.

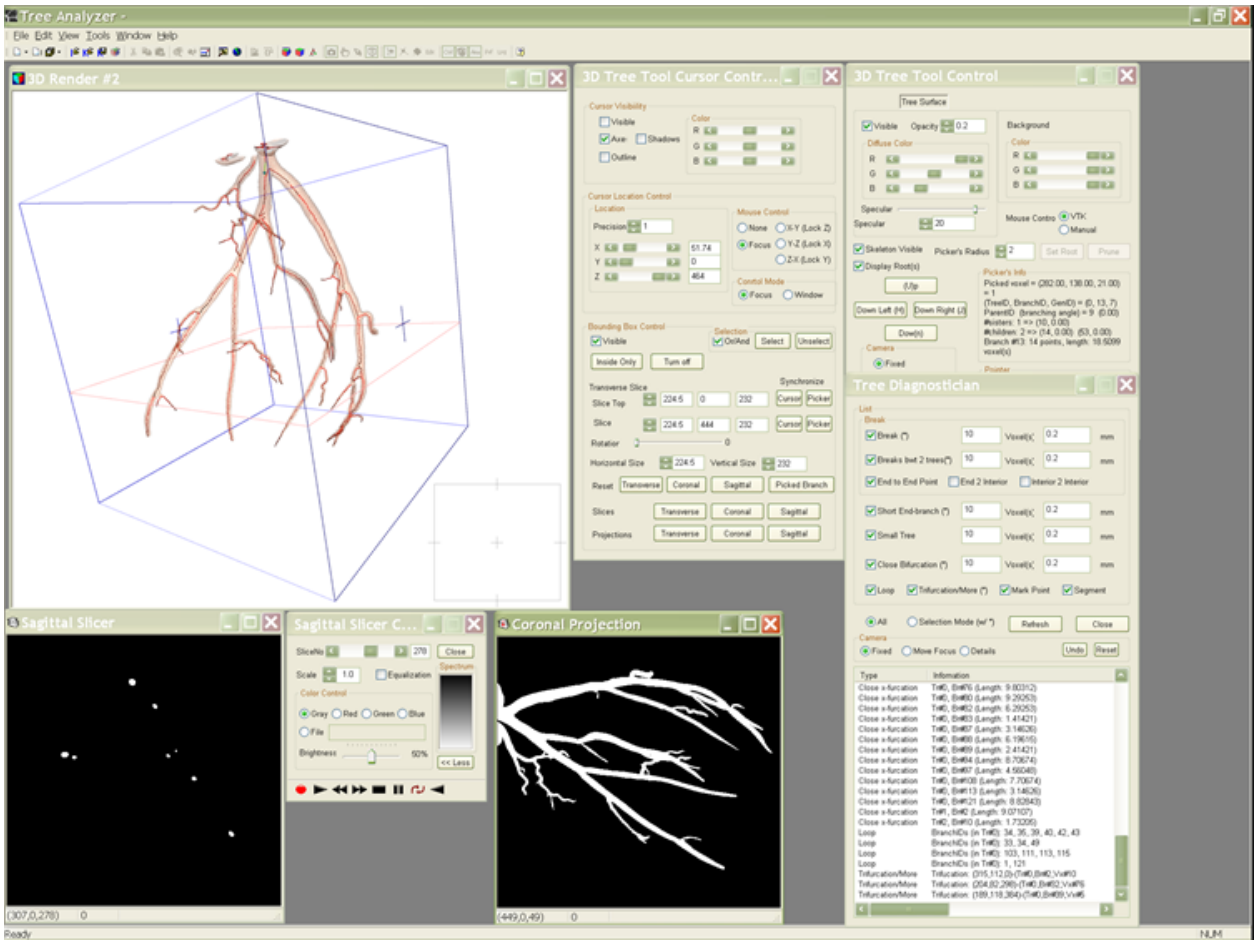



**Figure 4.** Composite view during tree editing of image H61. The 3D Render tool shows the segmented tree and extracted axial tree structure. Sagittal Slicer and Coronal Projection views focus on a particular bounding box location. Finally, other controls are visible for various manipulations.

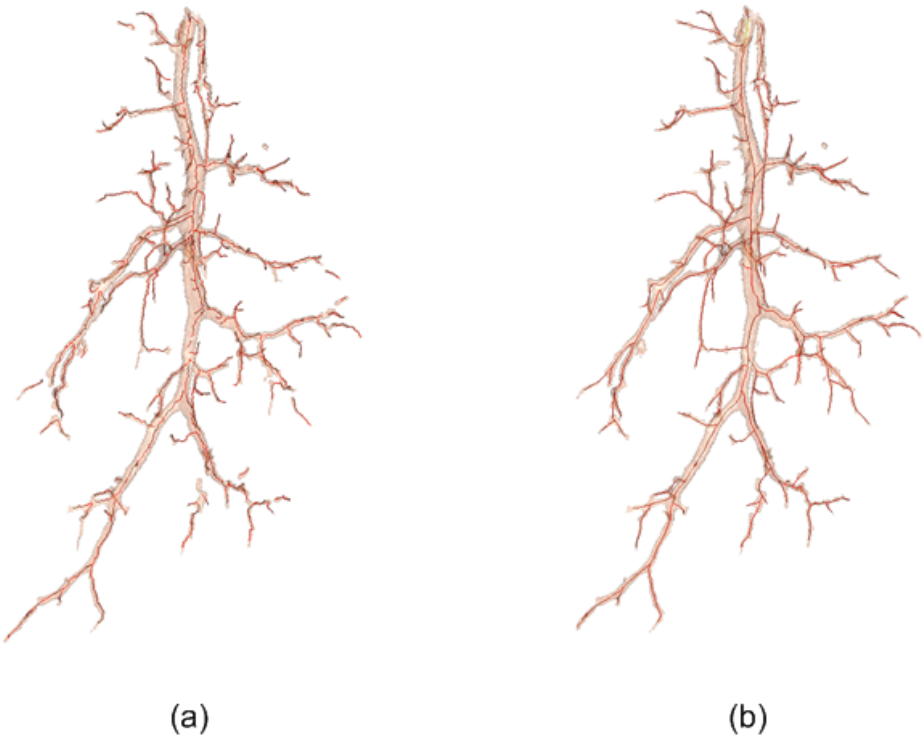



**Figure 5.** Control2 results. (a) Initial raw 3D tree structure and rendered segmentation. (b) Results after correcting tree. Original image is 400×400×375, with voxel size given by $\Delta x = \Delta y = \Delta z = 21\mu$m. The image contains a single hepatic lobe of a rat liver, where the bile ducts were first selectively opacified with a contrast agent before micro-CT scanning.[9]

Figure 7 shows the 2D Tree Map of the final edited result. Further work is needed in generating adequate groundtruth for determining the accuracy of the trees we can now generate.

## 4. DISCUSSION

This paper introduced a complete procedure for defining correct branching-tree structures depicted in large 3D CT images. We proposed many new graphical tools that allow the user to interrogate and fix various tree defects. Early results with these tools show their potential. While we have many tools for interrogating and correcting a tree, the process is still tedious. User interaction will be improved to help users work more easily and comfortably. In particular, a more intuitive graphical user interface is needed. Further, a more systematic use of the tools is required.

Currently, all defect correction is done with manual editing tools. We plan to introduce semi-automatic tools for correcting some of these defects. For example, a semi-automatic procedure can be defined for connecting breaks. The overall goal of this effort is to enable precise geometric tree definition, so that quantitative assessments can be made on trees. It is our aim to bring in such reliable quantitative measurements.

## ACKNOWLEDGMENTS

This work was partially supported by grants #EB00305 and #CA74325 from the NIH.

## REFERENCES

1. S. M. Jorgensen, O. Demirkaya, and E. L. Ritman, "Three dimensional imaging of vasculature and parenchyma in intact rodent organs with X–ray micro–CT," *Am J. Physiol (Heart, Circ Physiol 44)*, vol. 275, pp. H1103–H1114, 1998.
2. A. Garcia-Sanz, A. Rodriguez-Barbero, M. D. Bentley, E. L. Ritman, and J. C. Romero, "Three-dimensional micro-computed tomography of renal vasculature in rats," *Hypertension*, vol. 31, no. Part 2, pp. 440–444, 1998.

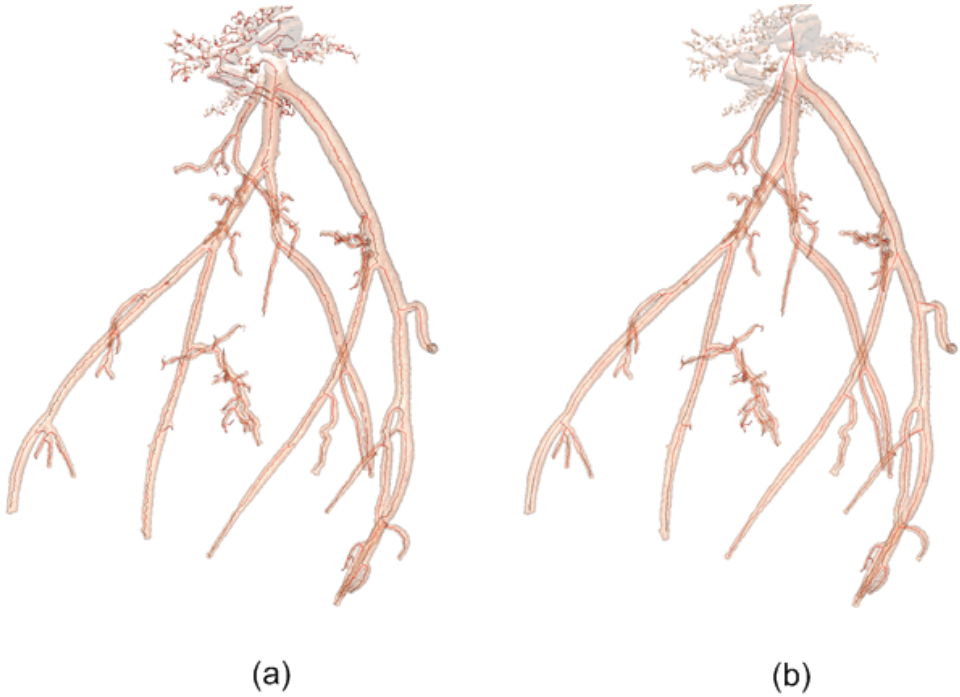



**Figure 6.** H61 results. (a) Initial raw 3D tree structure and rendered segmentation. (b) Results after Tree Editing. Original micro-CT image is 450×445×465, with voxel size given by $\Delta x = \Delta y = \Delta z = 20.1\mu$m. The image was derived from a cast of an opacified coronary arterial tree.[8]

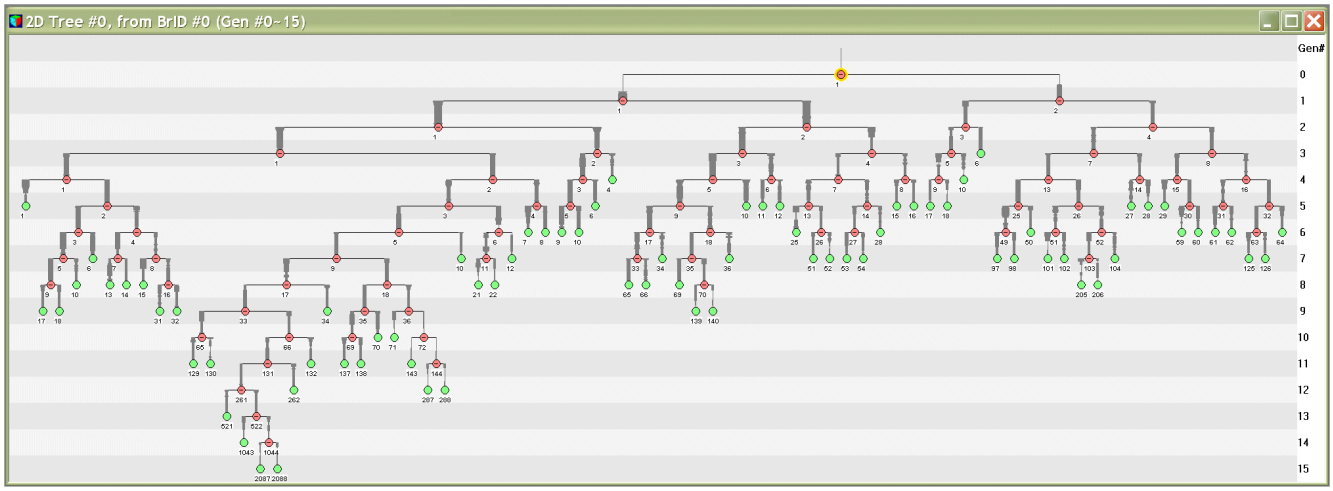



**Figure 7.** 2D Tree Map of H61 after correction. As shown, the final tree has fifteen generations. The numbers next to each branch indicate the branch index for that branch for a given generation. For example, at generation 5, branch indices 1-6, 9-18, and 25-32 are detected. Thus, 24 out of 32 fifth-generation branches are nominally labeled. Note, however, that it is not necessarily the case that this tree has 32 branches at generation 5 in perfect circumstances.

3. A. Feldkamp, S. A. Goldstein, A. M. Parfitt, G. Jesion, and M. Kleerekoper, "The direct examination of three-dimensional bone architecture in vitro by computed tomography," *Bone Min. Res.*, vol. 4, pp. 3–11, 1989.
4. W. Kalender, *Computed Tomography: Fundamentals, System Technology, Image Quality, Applications*. Munich: Publicis MCD Verlag, 2000.
5. P. J. L. Riviere and X. Pan, "Pitch dependence of longitudinal sampling and aliasing effects in multi-slice helical computed tomography (CT)," *Physics in Medicine and Biology*, vol. 47, no. 15, pp. 2797–810, Aug. 2002.
6. K. Taguchi, "Multislice helical computed tomography: Image reconstruction algorithms, implementations, and clinical evaluations," *Medical Physics*, vol. 29, no. 4, p. 643, Apr. 2002.
7. M. Zamir and H. Chee, "Branching characteristics of human coronary arteries," *Canadian J. Physiol. Pharmacology*, vol. 64, pp. 661–668, 1986.
8. M. Zamir, "Tree structure and branching characteristics of the right coronary artery in a right-dominant heart," *Canadian J. Cardiology*, vol. 12, no. 6, pp. 593–599, 1996.
9. S. Y. Wan, A. P. Kiraly, E. L. Ritman, and W. E. Higgins, "Extraction of the hepatic vasculature in rats using 3D micro-CT images," *IEEE Transactions on Medical Imaging*, vol. 19, no. 9, pp. 964–971, Sept. 2000.
10. P. J. Yim, "Gray-scale skeletonization of small vessels in magnetic resonance angiography," *IEEE Transactions on Medical Imaging*, vol. 19, no. 6, pp. 568–576, 2000.
11. F. K. H. Quek and C. Kirbas, "Vessel extraction in medical images by wave-propagation and traceback," *IEEE Transactions on Medical Imaging*, vol. 20, no. 2, pp. 117–131, Feb. 2001.
12. D. Selle and H. Peitgen, "Analysis of the morphology and structure of vessel systems using skeletonization," *SPIE Medical Imaging 2001: Physiology and Funct. from Multidim. Images, A. Clough and C. T. Chen, eds.*, vol. 4321, pp. 271–281, Feb. 18-20, 2001.
13. S. Y. Wan, E. L. Ritman, and W. E. Higgins, "Multi-generational analysis and visualization of the vascular tree in 3D micro-CT images," *Computers in Biology and Medicine*, vol. 32, no. 2, pp. 55–71, Feb 2002.
14. E. A. Kazerooni, "High resolution CT of the lungs," *American Journal of Roentgenology*, no. 3, pp. 501–519, Sept. 2001.
15. J. M. Reinhardt, N. D. D'Souza, and E. A. Hoffman, "Accurate measurement of intrathoracic airways," *IEEE Trans. Medical Imaging*, vol. 16, pp. 820–827, Dec. 1997.
16. R. H. Johnson, K. L. Karau, R. C. Molthen, and C. A. Dawson, "Exploiting self–similarity of arterial tree branches to reduce the complexity of analysis," *SPIE Medical Imaging 1999: Physiology and Function from Multidimensional Images*, vol. 3660, pp. 351–361, C. T. Chen and A. V. Clough (ed.), 1999.
17. R. H. Johnson, K. L. Karau, R. C. Molthen, S. T. Haworth, and C. A. Dawson, "Micro-ct image-derived metrics to quantify arterial wall distensibility reduction in a rat model of pulmonary hypertension," in *Proceedings of SPIE Medical Imaging 2000: Physiology and Function from Multidimensional Images*, vol. 3978, A. Clough and C.T Chen, eds., 2000.
18. K. L. Karau, R. C. Molthen, R. H. Johnson, A. H. Dhyani, S. T. Haworth, and C. A. Dawson, "Pulmonary arterial remodeling revealed by microfocal X-ray tomography," *SPIE Medical Imaging 2001: Physiology and Function from Multidimensional Images*, vol. 4321, pp. 18–20, Feb. 2001.
19. W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit, 2nd. Ed.* Upper Saddle River, New Jersey: Prentice Hall, 1998.
20. D. J. Kruglinski, *Inside Visual C++*. Redmond, WA: Microsoft Press, 1997.
21. E. Angel, *Interactive Computer Graphics: A Top-Down Approach with OpenGL*. Reading, Massachusetts: Addison-Wesley, 2000.
22. BCG, "Business components gallery (bcg)." World wide web URL: http://www.bcgsoft.com/.
23. A. Kiraly and W. E. Higgins, "Analysis of branching tubular structures in 3D digital images," *IEEE International Conference on Image Processing 2002*, pp. II–333 — II–336, 2002.
24. J. Abello and J. Korn, "Mgv: a system for visualizing massive multidigraphs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 21 –38, Jan.-Mar. 2002.
25. D. A. Keim, "Information visualization and visual data mining," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 1 –8, Jan.-Mar. 2002.
26. I. Herman, G. Melancon, and M. S. Marshall, "Graph visualization and navigation in information visualization: A survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pp. 24 –43, Jan.-Mar. 2000.
27. P. C. Wong, "Visual data mining," *IEEE Computer Graphics and Applications*, vol. 19, no. 5, pp. 20 –21, Sep.-Oct. 1999.