Centerline Reformations of Complex Vascular Structures

Gabriel Mistelbauer[*]
Vienna University of Technology, Austria

Andrej Varchola[†]
Vienna University of Technology, Austria

Hamed Bouzari[‡]
Austrian Academy of Sciences

Juraj Starinsky[§]
Comenius University, Bratislava

Arnold Köchl[¶]
KFJ Hospital Vienna, Austria

Rüdiger Schernthaner[‖]
Medical University of Vienna, Austria

Dominik Fleischmann[**]
Stanford University

Meister Eduard Gröller[††]
Vienna University of Technology, Austria

Milos Sramek[‡‡]
Austrian Academy of Sciences

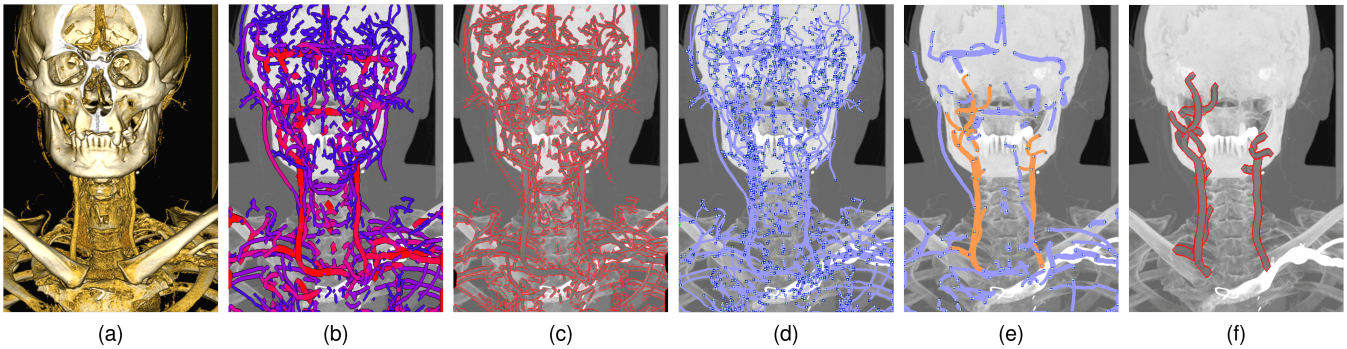


Figure 1: (a) 3D visualization of a CT data set. (b) Automatically detected cervical vessels visualized with size-based color-coding (thick vessels in red, thin ones in blue). (c) Vessel lumen of the whole vasculature with halos (red) and context (Maximum Intensity Projection), rendered by the proposed Centerline Reformation technique. (d) Automatically detected vessels (blue). (e) Vessels filtered according to length and thickness in order to select specific vessels (orange). (f) Carotis delineated with halos (red) and embedded into a context (Maximum Intensity Projection).

Abstract

Visualization of vascular structures is a common and frequently performed task in the field of medical imaging. There exist well established and applicable methods such as Maximum Intensity Projection (MIP) and Curved Planar Reformation (CPR). However, when calcified vessel walls are investigated, occlusion hinders exploration of the vessel interior with MIP. In contrast, CPR offers the possibility to visualize the vessel lumen by cutting a single vessel along its centerline. Extending the idea of CPR, we propose a novel technique, called Centerline Reformation (CR), which is capable of visualizing the lumen of spatially arbitrarily oriented vessels not necessarily connected in a tree structure. In order to visually emphasize depth, overlap and occlusion, halos can optionally envelope the vessel lumen. The required vessel centerlines are obtained from volumetric data by performing a scale-space based feature extraction. We present the application of the proposed technique in a focus and context setup. Further, we demonstrate how it facilitates the investigation of dense vascular structures, particularly cervical vessels or vessel data featuring peripheral arterial occlusive diseases or pulmonary embolisms. Finally, feedback from domain experts is given.

Index Terms: Computer Graphics [I.3.3]: Picture/Image Generation—Display algorithms Computer Graphics [I.4.6]: Segmentation—Edge and feature detection

1 Introduction

The precise analysis and visualization of vascular structures play a crucial role in therapy-centric visualization tasks. Understanding vessel morphology and spatial relationships can lead to better diagnosis and planning of further treatment. Vascular trees are often very complex, with hundreds of branchings, which might occlude each other in rendering. Their reconstruction and visualization is therefore a challenging problem [31].

Medical volume data with a focus on vessels typically comes from clinical Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) angiography. Although the vascular structures are usually contrast enhanced, their intensities are not perfectly separated from other structures such as bones. In the case of pathologies that occur on vessel walls, such as plaque and calcifications, it is difficult to assess the vessel lumen, due to occlusion effects. Because of various limitations, in many cases conventional visualizations like iso-surface extraction, Maximum Intensity Projection (MIP), or Direct Volume Rendering (DVR) with simple transfer functions do not provide diagnostically satisfactory images. Curved Planar Reformation (CPR) [13] is an advanced visualization method for vessel diagnosis, where a 2D visualization of the vessel lumen is obtained by reformatting a curved cut along the vessel centerline.

The main contribution of this work is Centerline Reformation (CR), as a new visualization method with the objective to visualize complex vessel trees for diagnostic purposes, as presented in Fig. 1(c) on the example of cervical vessels. Our method was motivated by the CPR technique described by Kanitsar et al. [13] but goes beyond its restrictions. The core novelty of the technique is the way how the lumen of vessels is reformatted. A rotation independent approach is used, which is based on centerline reconstruction

[*] e-mail: gmistelbauer@cg.tuwien.ac.at
[†] e-mail: varchola@cg.tuwien.ac.at
[‡] e-mail: hamed.bouzari@oeaw.ac.at
[§] e-mail: juraj.starinsky@gmail.com
[¶] e-mail: a.koechl@wienkav.at
[‖] e-mail: ruediger.schernthaner@meduniwien.ac.at
[**] e-mail: d.fleischmann@stanford.edu
[††] e-mail: groeller@cg.tuwien.ac.at
[‡‡] e-mail: milos.sramek@oeaw.ac.at

and wavefront propagation of the centerline information within the vessel extent (radius). This new approach offers the possibility to reformat and visualize the lumen of spatially arbitrarily oriented vessels and arbitrarily complex vascular structures. The centerlines and radii are determined in a preprocessing step and are used for the lumen reformation. Their knowledge allows extending the visualization by halos, by vessel filtering according to various criteria and user interaction. Furthermore, the visualization can be augmented in the focus and context sense, for example by the VesselGlyph technique [26, 27].

The remainder of this paper is organized as follows. The subsequent section gives related work concerning multi-scale analysis and CPR. Section 3 describes the applied vessel-detection pipeline. Section 4 explains our novel CR technique. Section 5 describes implementation details. In Section 6 results and discussions are presented and a clinical evaluation of CR is given in Section 7. Concluding remarks and future aspects are described in Section 8.

2 Related Work

Volume visualization is an important part of Computed Tomography Angiography (CTA), which is a non-invasive diagnostic procedure to evaluate patients in hospitals. CT scanners can produce an angiogram with a simple injection of a contrast agent into a peripheral vein. Quantification of vascular anomalies generally involves the definition of a centerline, which is a line-like representation of a 3D object [28], either directly by path tracing, or indirectly, by first segmenting the lumen [32].

2.1 Vessel Detection

A pipeline for vessel extraction is proposed by Selle et al. [23] and Hahn et al. [8]. Frangi et al. [7] describe a vessel-enhancement filter based on the eigenvalue analysis of the Hessian matrix in scale-space. Vessel-enhancement filters improve the visibility of vessel structures, which are geometrically described as curved tubular objects [20]. The so called *multi-scale analysis* is performed by first convolving the image with Gaussian derivative filters on different scales and subsequently analyzing the second-order derivatives, i.e., the Hessian matrix, at every image element [19, 15]. This approach is applied in our vessel-detection pipeline in order to retrieve vessels of various sizes. The high computational cost of the Hessian analysis on multiple scales in the 3D data can be reduced by performing local instead of global operations [1]. Lesage et al. [18] give an overview of vessel-lumen segmentation-techniques, including multi-scale analysis. Joshi et al. [12] describe a non-parametric entropy-based vessel detection that handles bifurcations better.

A comprehensive curve-skeleton survey can be found in the work of Cornea et al. [4, 3]. Thinning methods iteratively remove points from the boundary of an object while preserving topology. This is the method we apply to retrieve the skeleton of vessels obtained from the multi-scale analysis. In distance-field methods, ridges in the distance field correspond to voxels that are locally centered within the object [21]. Tagliasacchi et al. [29] describe a method that uses the rotational symmetry axis of a point set to obtain its skeleton. Zhu et al. [35] propose smooth harmonic skeletons of a triangulated representation of the vessel surface.

Jones et al. [11] give an overview of several techniques for computing distance fields. They distinguish between two approaches based on sweeping and on wavefront calculation. The first one calculates the distance field by sweeping through the data multiple times, whereas the second one propagates the distances from a starting point to its vicinity. Sethian [24, 25] describes the Fast Marching Method (FMM), which is a wavefront method and used by our proposed CR technique for lumen rendering. Distances are computed by expanding the wavefront in the normal direction of a set of points. Telea et al. [30] augment the FMM method for computing skeletons and centerlines.

Streaming approaches have been investigated to reduce the memory load of the vessel-detection pipeline. Law et al. [16] give key principles for streamable data and describe a block-based streaming technique by partitioning a volume into blocks. Varchola et al. [33] describe slice-based streaming of volumetric data, which is the approach applied in this paper.

2.2 Vessel Visualization

Reformation is a visualization technique to display interpolated values along a cutting surface. Multi-planar reformation is a traditional technique which enables orthogonal and later also oblique cutting planes through the data.

Reformation in CPR is performed over a non-planar surface. Kanitsar et al. [13] distinguish three types of CPRs, namely projected, stretched and straightened CPRs. In order to generate the curved reformation along a vessel centerline, lines-of-interest have to be determined, as shown in Fig. 2. They can have any orientation, although this might lead to problems, such as overlappings, when vessels are curved or their radii are too large. Otherwise, if only horizontal or vertical lines-of-interest are used, artifacts as, for example, thinning of the vessel might occur.

The Multipath Curved Planar Reformation (mpCPR) technique has been initially designed to investigate the peripheral arterial occlusive disease of the arteries of the human lower extremities [14]. These peripheral vessels split first into two branches (femoral arteries left and right) and each later on into another three branches (posterior, anterior and peroneal arteries), leading to six main branches. The mpCPR method [14, 22] exploits the specific geometry and spatial orientation of the vessel-tree to partition the image according to the branchings. These image parts are rendered separately using CPR and are combined to the final image, which leads to artifacts at the boundaries of the partitions.

Jianu et al. [10] explore the brain connectivity by projecting 3D fiber tracts into 2D and subsequently cluster them, with each cluster having one centroid and many non-centroid tracts. They account for visibility by depth sorting the centroid tracts according to the depth of the center of the corresponding 3D segments. In our case, we use the distance along the graph of all projected vessels together with the depth of a projected centerline voxel to determine proper visibility. This leads to a smooth lumen visualization at the connections of consecutive segments.

Straka et al. [26, 27] proposed the VesselGlyph to establish focus and context rendering for CPR and mpCPR. It provides the possibility to visualize vessels with a context created by methods like MIP or DVR. This is also supported by our technique.

Interrante et al. [9] mention halos as an enhancement of depth perception in 3D flow visualizations. Wenger et al. [34] give a volumetric approach for halos in order to enhance the visualizations of blood flows or neural pathways resulting from Diffusion Tensor Imaging. Everts et al. [6] give a geometrical technique for enveloping neural pathways with halos. We adapt this method in order to augment the vessel lumen with halos when investigating dense vascular structures.

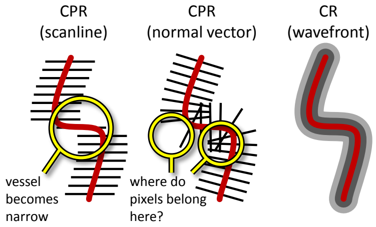


Figure 2: Comparison between CPR using parallel scanlines (left) and normal vectors (center) as lines-of-interest. The proposed CR technique spreads wavefronts from the centerline outwards (right).

3 Vessel-Detection Pipeline

Since vessel centerlines are required to visualize the lumen, we have implemented a vessel-detection pipeline which is based on the work of Hahn et al. [8]. The pipeline uses multi-scale analysis of the data set to enhance tubular structures, followed by segmentation based on Hysteresis thresholding (HT), skeletonization and finally conversion of the skeleton into a graph representation. This representation offers the possibility to filter vessels according to specific constraints during interactive visualization and lets the user select vessels by visual queries utilizing brushing. The pipeline is additionally capable of handling large volumetric data sets by employing data streaming approaches [33].

In order to relieve clinicians of the time consuming manual vessel segmentation, our pipeline provides them with a semi-automatic segmentation. The user input has been reduced to the necessary minimum which consists in specifying a range of vessel sizes (sigma values) and two thresholds for HT that are used for vessel segmentation. In order to incorporate changes of these parameters, the pipeline needs to be re-run. Since clinicians usually do not want to specify and modify parameters, a default choice adapted for their daily purposes was experimentally defined.

The workflow of the vessel-detection pipeline is outlined in Fig. 3. It consists of three major steps, which are automatically executed: preprocessing, scale-space analysis and centerline generation. Subsequently, they will be discussed in more detail.

3.1 Preprocessing

To support input from various acquisition modalities, preprocessing is a necessary step to feed the pipeline with data in a proper format and with certain expected properties. Owing to the fact that various scanners produce data with a different range of values, this step consists primarily of contrast enhancement, such that vessels remain the structures with the highest density. This may require remapping of the density values by a windowing function, in order to suppress other types of tissues with potentially higher density, such as bones [26]. Since various types of tissues are not distinguishable solely based on their density, unwanted tissues are usually only partially suppressed.

3.2 Scale-Space Analysis

Scale-space analysis is used for vessel detection, since only one parameter needs to be specified, namely the Gaussian standard deviation σ. The fact that it corresponds to the size of the detected vessels simplifies the user input for clinicians. Since vessels are tubular-shaped structures we enhance them with the vesselness operator which is based on an eigenvalue analysis of the Hessian matrix [7]. Other and more advanced vesselness operators are applicable here as well, e.g. the one by Pock et al. [20]. A vesselness operator enhances tubular-shaped structures of a certain scale and suppresses others. Thus, in order to enhance all vessels of interest captured in the data, a batch of filters has to be applied with values of the Gaussian σ covering the desired range of vessels.

Vessel segmentation is based on the Hysteresis thresholding (HT) from the Canny's edge detection framework [5]. This operation is applied to the respective results of the Hessian enhanced volumetric data sets. The results are binary volumes where the foreground corresponds to the detected vessels of the appropriate scale, as shown in Fig. 3 in the right image of the scale-space analysis plate. The advantage of HT resides in the suppression of spurious and not well pronounced structures and noise.

3.3 Centerline Generation

After all scales have been processed, they are represented as binary volumetric data sets. A mask of all vessels is obtained as a union of the vessel masks of the individual scales. Simultaneously, an auxiliary volume is created which stores for each foreground voxel the Hessian scale with the maximum response. If later a position is identified as a centerline voxel, this value is regarded as the *vessel-radius* of the centerline at this voxel.

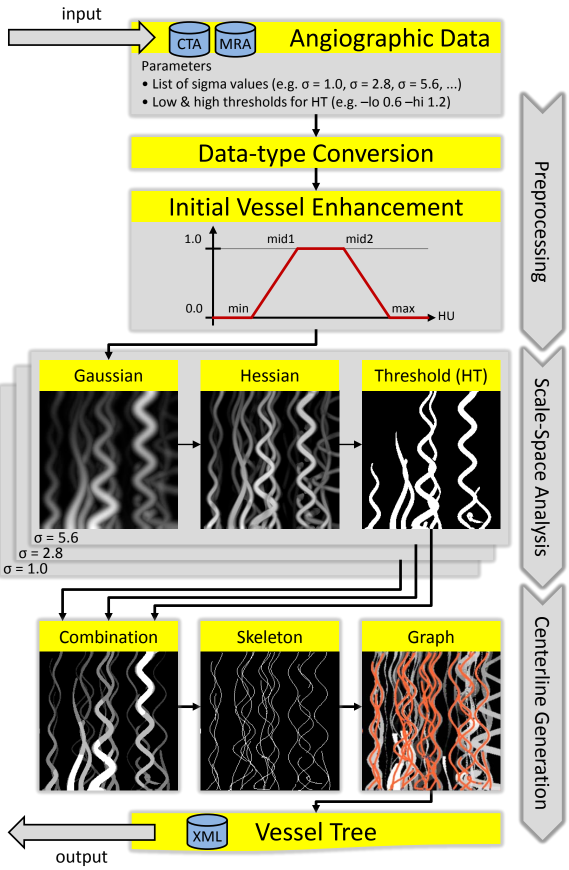


Figure 3: Workflow of the vessel-detection pipeline illustrated on an artificial data set. The arrows on the right side indicate the automatically processed parts of the pipeline. In the preprocessing step the data set is converted to the desired representation and vessels might be initially enhanced using a windowing function. During scale-space analysis, the Gaussian and Hessian filters followed by HT are applied for each scale in parallel (three scales are illustrated here). After combining all scales, the vessel centerlines are detected by skeletonization and finally converted into a graph representation.

The centerlines of the previously segmented and combined vessels are determined by skeletonization in 3D using a thinning technique [17]. Although more elaborate methods exist, we used this one due to implementation simplicity. As result a one-voxel thin 26-connected 3D skeleton is obtained.

After analyzing the topology of the skeleton we represent it as a graph, which is done in a separate step called graph conversion. An acyclic graph $G = (V, E)$, the vessel-tree, is created, where the set of vertices V consists of branch-points (with ≥ 3 neighbors) and end-points (with exactly one neighbor). All other points have two neighbors and correspond to exactly one edge in the edge set E. We additionally compute the *thickness* of an edge as the average of the radii of all its points and the *length* as the number of all its points. These two properties are used for interactive filtering of vessels in order to select a specific subset of the whole vasculature.

4 Centerline Reformation

The major contribution of this paper is the Centerline Reformation (CR) for reformation and visualization of the lumen of vessels and

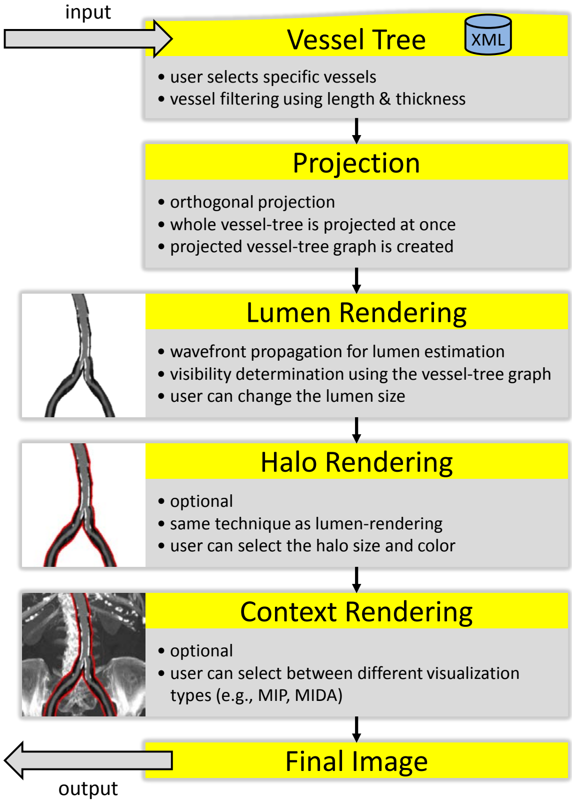

input
Vessel Tree
XML
• user selects specific vessels
• vessel filtering using length & thickness
Projection
• orthogonal projection
• whole vessel-tree is projected at once
• projected vessel-tree graph is created
Lumen Rendering
• wavefront propagation for lumen estimation
• visibility determination using the vessel-tree graph
• user can change the lumen size
Halo Rendering
• optional
• same technique as lumen-rendering
• user can select the halo size and color
Context Rendering
• optional
• user can select between different visualization types (e.g., MIP, MIDA)
Final Image
output

Figure 4: Workflow of CR. The vasculature is projected into the image space. Then the lumen is rendered and optionally enveloped by halos to enhance depth perception. The final image is composed by adding a context visualization if desired.

other vascular or tube-like structures given as volume data. The technique belongs to the type of projected CPRs, as categorized by Kanitsar et al. [13], and therefore uses no flattening or straightening.

The algorithm can be divided into four main parts, as shown in Fig. 4. First, the whole vessel-tree is projected into the viewing plane. In the second step, the projected vessel-tree is grown until the extent of each vessel is reached. Third, halos can be optionally rendered around the vessel lumen. Finally and fourth, the resulting image is composed by optionally augmenting the rendered lumen or halo with a context visualization. In the following sections each step will be explained in more detail.

4.1 Projection

The first part of our algorithm is the projection. Input is a user-selected subset $sel(G)$ of the 3D vessel-tree G with centerlines e_i as its edges and $v_j \in e_i$ as their corresponding voxels. We project this subset into 2D using orthogonal projection

$$sel(G) \mapsto G^{2D} \qquad (1)$$

with $e_i^{2D} \in G^{2D}$ as the projected centerlines, where each of them is a list of pixels $p_j = (x_j, y_j)$. Additionally, the adjacency information of G is used to build the graph G^{2D} in the 2D projection space. Then, we compute for every e_i^{2D} independently the arc-length, $l_{arc}(p_j)$, for every pixel $p_j \in e_i^{2D}$, by the following recursive equation

$$l_{arc}(p_j) = l_{arc}(p_{j-1}) + d(p_j, p_{j-1}), \qquad l_{arc}(p_0) = 0 \qquad (2)$$

where $d(p,q)$ is the distance between two pixels, depending on the currently used distance metric (Manhattan or Euclidean) and neighborhood (4 or 8 connected). This is correlated to the arc-length parametrization of the edges e_i^{2D} and is similar to the approach proposed by Telea et al. [30]. They parametrize the boundary of an object according to the arc-length and subsequently move inwards to compute the skeleton. Since we have the centerline of a vessel and want to obtain the lumen, we move the other way. Starting from the projected vessel-tree, we propagate wavefronts outwards, until the extent of each vessel is reached.

Because multiple vessels might overlap in the 2D projection space, we use two buffers, a top-level buffer $B_T(p_j)$ and a candidate buffer $B_C(p_j)$ with pixels $p_j = (x_j, y_j)$, in order to render the vessel overlap areas correctly. The top-level buffer contains the lumen of vessels that are currently visible. All the other, currently hidden, vessels are stored in the candidate buffer, because they might become visible when growing further. Both buffers have the dimension of the final image and store the following properties for every pixel, either only once in B_T, or in a list in B_C:

- a reference to the edge e_i^{2D} the pixel belongs to
- the depth value of pixel $p_j \in e_i^{2D}$
- the arc-length $l_{arc}(p_j)$ of pixel $p_j \in e_i^{2D}$
- the voxel $v_k \in e_i$ the pixel belongs to
- the vessel-radius
- the type of the pixel (vessel, halo, background)

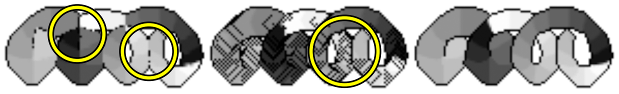
Whenever we refer to a pixel in one of these buffers in the remainder of this paper, we mean the pixel's 2D position together with the pixel's associated properties. The lists at every pixel of buffer B_C are additionally depth-sorted and contain the information of all candidate wavefronts, which are currently hidden, but might become visible when spreading further. A pixel of a currently hidden candidate wavefront can be rejected from spreading further, if its preceding pixel in the candidate list has a larger vessel-radius, because the former pixel will always remain behind the latter one. This reduces the total number of propagated wavefront pixels and, hence, increases overall performance. The computation of all candidate pixels of the initial wavefront, which is the projected vessel graph G^{2D}, is outlined in the subsequent code-snippet. There, U is a list of pixels where B_C needs to be investigated during the lumen rendering, because one pixel of each of these candidate lists might be closer to the viewing plane than the current top-level pixel.

```
$U \leftarrow \emptyset$                                   list of pixels, whose candidate lists in $B_C$ need to be investigated
foreach $e^{2D} \in G^{2D} : \forall p \in e^{2D}$ do        for all pixels of all projected edges
    computeAndAddProperties($p$)                          compute every associated property
    posInList $\leftarrow B_C(p) \cup \{p\}$                      insert pixel into candidates
    if radius($B_C(p)$[posInList $-$ 1]) $\geq$ radius($p$) then
        $B_C(p) \leftarrow B_C(p) \setminus \{p\}$               reject candidate because of the smaller radius
    else if $\#B_C(p) = 1$ then                              only one candidate in list so far?
        $U \leftarrow U \cup \{(x(p), y(p))\}$                      add the pixel's position
```

It is sufficient to store the position of pixel p_j for every candidate in the list $B_C(p_j)$ only once in U. Finally we get all candidate pixels of the initial wavefront, $\forall p_i \in U : B_C(p_i) \neq \emptyset$.

4.2 Lumen Rendering

Our approach to reformat the lumen of the selected vessels is based on the Fast Marching Method (FMM) for propagating wavefronts, together with their properties, to neighboring pixels. We parametrize the edges of G^{2D} along their arc-length and extend this distance parametrization to the whole graph G^{2D}. Finally, we use



(a) no arc-length, no depth buffer (b) no arc-length, depth buffer (c) arc-length, depth buffer

Figure 5: Comparison when using the arc-length parametrization or depth buffering for visibility. Since grey value gradients are stored in the data, they should be clearly reflected in the lumen visualization. Artifacts (highlighted with yellow circles) occur in (a) because pixels are never changed and in (b) since too many pixels are falsely overwritten. (c) shows the result of our proposed method.

this parametrization to determine the visibility of the vessels during lumen rendering.

Our method is motivated by the observation that never altering a pixel once it has been spread to its neighbors, as done by the FMM, leads to visibility problems. These are illustrated in Fig. 5(a) (highlighted with yellow circles). The reason is that a pixel of the same wavefront would not be changed, even if another one is closer to the viewing plane, since they are not depth-buffered. If only the depth information is used, the results would look like in Fig. 5(b), because always the pixels closest to the viewing plane are taken, regardless if they will overwrite the lumen of pixels behind them. Hence using the depth information only is not sufficient to render the lumen correctly while accounting for proper visibility. Therefore, we have to find a more sophisticated way when to alter an already grown pixel.

Before we can describe this, we need to define the distance between two points of the graph G^{2D} as illustrated in Fig. 6. The distance between two points of the same edge $e^{2D} \in G^{2D}$, called edge-distance, d_e, is computed as the arc-length difference of the two points

$$d_e(p,q) = |l_{arc}(p) - l_{arc}(q)| \qquad (3)$$

and the distance between two points of different edges, namely the inter-edge distance, d_{ie}, as

$$d_{ie}(p,q) = \begin{cases} \infty & \text{if } \nexists path(p,q) \\ \sum_{i=0}^{n-1} d_e(N_i, N_{i+1}), N_0 = p, N_n = q & \text{if } \exists path(p,q) \end{cases} \qquad (4)$$

where $N_1 \dots N_{n-1}$ are the nodes of the graph G^{2D} along the path between the points p and q. The distance is computed by searching for a path in a depth-first fashion. If such a path exists, it is unique, because G^{2D} is acyclic. We can now define a distance metric over the graph G^{2D}, using these two distances, as

$$d_{graph}(p,q) = \begin{cases} d_e(p,q) & \text{if } edge(p) = edge(q) \\ d_{ie}(p,q) & \text{if } edge(p) \neq edge(q) \end{cases} \qquad (5)$$

for any two points of G^{2D}. This distance metric is utilized during wavefront propagation to determine visibility of colliding fronts.

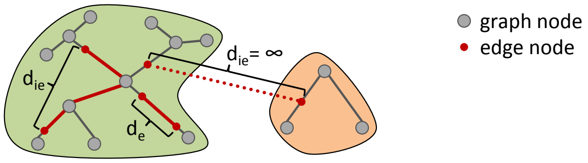


Figure 6: Distances d_{graph} between two points along the projected vessel graph G^{2D}, which consists of two disconnected graphs (highlighted in green and orange).

The growing process can be divided into two parts. First, in an update pass, all candidates of the list $B_C(p_j)$ of all pixels $p_j \in U$ are investigated and checked, whether they become visible and might contribute to $B_T(p_j)$. Since the list at $B_C(p_j)$ is depth-sorted, only the first element becomes visible if it is closer to the viewing plane than $B_T(p_j)$. Then, all candidates, including the first one, spread to their neighbors, unless they are discarded owing to a larger vessel-radius of a preceding candidate. The following code outlines how to determine the pixels that will spread to their neighbors by using the candidate pixel list U.

```
$P \leftarrow \emptyset$      resulting list of pixels with their properties that need to be propagated
foreach $p \in U$ do      for all pixels that need to be updated
    first $\leftarrow B_C(p)[0]$      take first candidate
    top $\leftarrow B_T(p)$      closest one is current top-level
    if depth(first) $<$ depth(top) then      first candidate closer than top?
        $P \leftarrow P \cup \{B_T(p) \leftarrow$ top $\leftarrow$ first$\}$      assign first to buffer and add to list
        $B_C(p) \leftarrow B_C(p) \setminus \{$first$\}$      remove first from candidates
    foreach $c \in B_C(p)$ do      for all remaining candidates
        if radius$(c) \geq$ radius(top) then      radius of candidate $\geq$ top?
            $P \leftarrow P \cup \{$top $\leftarrow c\}$      candidate becomes top and is added
    $B_C(p) \leftarrow \emptyset$      finally clear candidate list at pixel p in $B_C$
```

When the update pass is finished, the second part, namely the propagation, is performed in a breadth-first processing of G^{2D} starting with the propagation list P that has been obtained from the initial wavefront. Every pixel of the list P is spread to its neighbors, together with its properties. The vessel-radius is decreased according to the chosen distance metric and when it reaches zero, the propagation is stopped. The user is capable to offset the radius by a constant. This is particularly useful when the vessel-radius, which has been detected by our vessel-detection pipeline, is not sufficient to investigate possible suspicious regions on the vessel walls.

Since we propagate the wavefronts by simply iterating through their pixels, this might lead to different results when using another order, e.g. reverse iterate. To alleviate this, if multiple pixels spread to the same neighbor and if they are close together according to d_{graph}, all their properties are interpolated. Fig. 7 outlines how the wavefronts are propagated and when interpolation is taking place. A user-defined threshold determines when two pixels are considered close together. In order to relieve clinicians from manually selecting this threshold, a default value for their application purposes is given. During propagation, all candidates belong to the same wavefront, since the candidate buffer is cleared at every pixel after the update pass. However, one must be aware, that interpolation might break the depth-sorted invariance of the candidate lists. Therefore, all interpolated pixels are removed and added back later. The propagation is outlined in the following algorithm, where P is

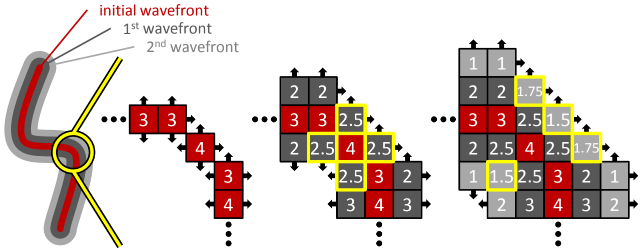


Figure 7: Illustration of pixels spreading to their neighbors using 4-neighborhood and Manhattan metric. The numbers indicate the vessel-radius and interpolated pixels are outlined in yellow.

the current propagation list and U is the list of pixels for the next update pass.

```
$U \leftarrow \emptyset$    list of pixels whose candidates need to be updated
foreach $p \in P : \forall n \in N_{neighborhood}(p)$ do    for all neighbors of all pixels
    $p \mapsto n$    spread p to its neighbor n (radius−−, if halo update depth, ...)
    if $d_{graph}(n, B_T(n)) >$ threshold then    neighbor away from top?
        $A \leftarrow \emptyset$, neighborSkipped $\leftarrow false$    A is a list of changed candidates
        foreach $c \in B_C(n)$ do    loop through every candidate c of $B_C$ at pixel n
            if $d_{graph}(n, c) <$ threshold then    neighbor close to candidate?
                if halo$(n) \wedge \neg$halo$(c)$ then    skip halo if candidate is lumen
                    neighborSkipped $\leftarrow true$    halo does not overwrite lumen
                else if $\neg$halo$(n) \wedge$ halo$(c)$ then    neighbor lumen, cand. halo?
                    $B_C(p) \leftarrow B_C(p) \setminus \{c\}$    remove candidate from list
                    $A \leftarrow A \cup n$    lumen can overwrite nearby halo
                else if (type$(n) =$ type$(c)$) then    both same type
                    $B_C(p) \leftarrow B_C(p) \setminus \{c\}$    remove candidate from list
                    $A \leftarrow A \cup$ lerp$(c, n)$    interpolate candidate with neighbor
        if $\neg$neighborSkipped $\wedge (\#A = 0)$ then    neighbor not interpolated?
            $B_C(n) \leftarrow B_C(n) \cup \{n\}$    add neighbor to candidates
            if $\#B_C(n) = 1$ then $U \leftarrow U \cup \{(x(n), y(n))\}$
        else $B_C(n) \leftarrow B_C(n) \cup A$    add interpolated candidates back
    else if $(\neg$halo$(n) \wedge$ halo$(B_T(n)))$ then    neighbor lumen, top halo?
        $B_T(n) \leftarrow n$    lumen can overwrite nearby halo
    else if (type$(n) =$ type$(B_T(n))$) then    both same type
        $B_T(n) \leftarrow$ lerp$(B_T(n), n)$    interpolate top with neighbor
```

4.3 Halo Rendering

Once the vessel-radius of a pixel from a wavefront reaches zero, the pixel is optionally added as halo for propagation again. The halos will augment the lumen visualization to delineate visibility of different overlapping vessels, as illustrated in Fig. 8 and presented in Fig. 1(c) with halos in red. The user can select the size of the halo in pixels. Although halos are rendered in the same way as the lumen, they are treated slightly differently. Their depth is increased proportionally to the distance to the centerline during spreading (indicated in the third line of the previous algorithm). Furthermore, only pixels of the same type can be interpolated, namely if both are either halos or not halos.

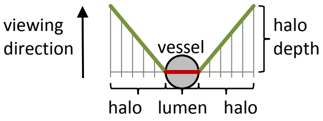


Figure 8: Illustration of halos (green) around the vessel lumen (red).

4.4 Context Rendering

In order to preserve an overview, a context visualization can augment the lumen visualization. If the lumen of every selected vessel has been rendered, possibly with halos, different tasks according to the type of every pixel p_j of the buffer $B_T(p_j)$ are performed. There are three pixel types: vessel, halo and background pixels. If the pixel is a *vessel*, the volume data is sampled at the stored voxel position. For *halo* pixels the user-specified halo color is assigned. If a *background* pixel is encountered in the top-level buffer, either a specific color is assigned or a context, using MIP or Maximum Intensity Difference Accumulation (MIDA) [2], is rendered. Since we use orthogonal projection for the lumen visualization, we apply it for the context rendering as well.

Rendering of halos combined with the alternative rendering in the context area is similar to the focus and context rendering proposed by Straka et al. [27]. They introduce the VesselGlyph which combines DVR (context) together with CPR (focus) within a single image. While Straka et al. specify the VesselGlyph geometry by means of a volumetric model based on vessel centerlines and corresponding 3D distance fields, we distinguish between the vessel lumen, halo and background areas only in the 2D projection space. Like in the VesselGlyph, this offers the possibility to embed the lumen smoothly into the context area.

5 Implementation

We integrated our technique in a frame-work which is deployed in a hospital and is utilized in the daily routine of radiologists. Since this software currently uses CPU only, the proposed algorithm is implemented on the CPU. However, we can visualize the vessel lumen at interactive frame-rates, where the performance is inverse proportional to the number and length of the rendered vessels. Although pixels can be rejected before spreading, the propagation consumes the major part of the rendering time, whereas the projection and creation time of the vessel graph remain almost constant. The time consumption of the update pass scales with the number of overlapping vessels, because more candidates have to be processed.

Since context rendering is also performed on the CPU, we accelerated it by pre-computing and caching ray directions only once for all slices. This is possible as we use orthogonal projection. Nevertheless, a context cannot be rendered in real-time.

6 Results and Discussion

An overview of several data sets processed by the described vessel-detection pipeline is given in Table 1. The value ranges of the artificial helices and the CT data are largely different. Therefore also the respective Hysteresis thresholds (HT) differ considerably. To support clinicians in the daily routine, default configurations for the vessel-detection pipeline are given.

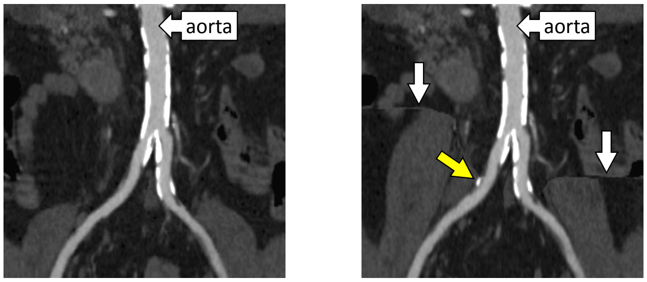
A comparison between mpCPR, using horizontal scanlines, and CR is given in Fig. 9 for the aorta of a human lower extremity data set. Fig. 10 demonstrates the advantage of CR over mpCPR for a data set that includes a bypass connecting both femoral arteries. Our method correctly processes both vertical and horizontal vessels in one image, in contrast to mpCPR, because scanlines are not a proper solution here. In order to visualize the surrounding organs with CR, the vessel-radius is offset accordingly.

Fig. 11 presents another comparison on the artificial helical data set. An issue with mpCPR is the overlapping of helices in 2D pro-

Table 1: Configuration of the vessel-detection pipeline for several processed data sets. Timings are given for a whole run.

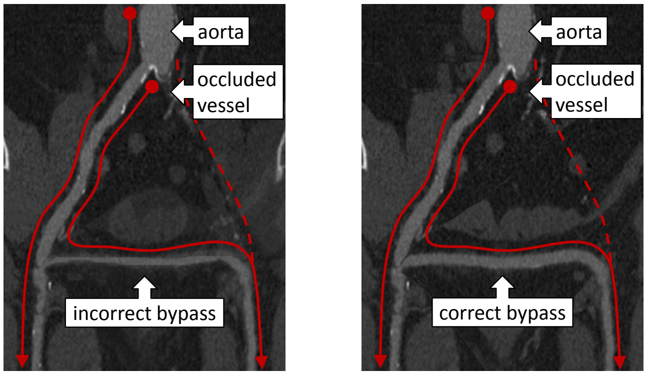
	A	**B**	**C**	**D**
Size	256^3	512^3	$512^2 \times 575$	$512^2 \times 1227$
#Scales	9	6	13	8
Lo/Hi HT	50k/70k	0.6/1.2	0.6/1.2	0.6/1.2
Time PC1[a]	4.8min	7.2min	23.4min[c]	21.6min[c]
Time PC2[b]	1.3min	4.3min	7.4min	7.7min

A... Artificial Helices C... Cervical Vessels
B... Pulmonary data D... Human Lower Extremities
[a] Intel Core2 @ 2.80GHz / 4GB RAM / GF 8800GTS 320MB
[b] Intel Core i7 @ 3.07GHz / 12GB RAM / GF 460GTX 1GB
[c] Individual scales were run sequentially (Fig. 3), since in parallel their demands exceeded the GPU RAM size.



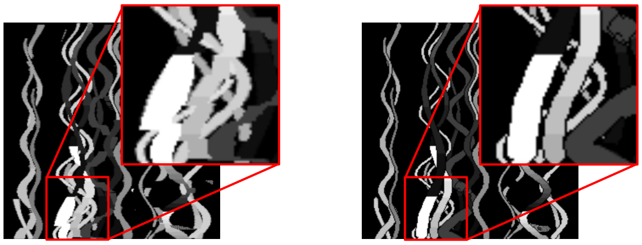
(a) (b)

Figure 9: Comparison between mpCPR (a) and CR (b) of the aorta splitting into femoral arteries left and right. The discontinuities (white arrows) result from the fact that the distance d_{graph} is used to determine whether a pixel is interpolated during spreading. However, they occur only in the surrounding parts and do not affect the lumen itself. The bright calcification (yellow arrow) is correctly shown by CR, because the depth of this region is obtained by wavefronts and not by horizontal lines-of-interest as in mpCPR.



(a) (b)

Figure 10: Data set with a bypass when the aorta splits into left and right (occluded, shown as dashed line) femoral arteries. The red lines indicate the direction of the blood flow. (a) mpCPR cannot visualize the bypass properly. (b) CR clearly delineates it.



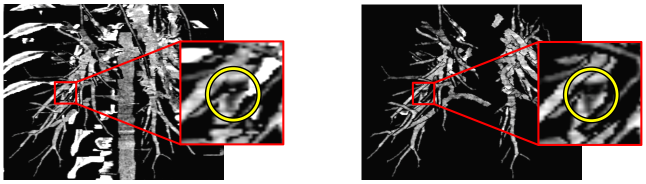
(a) (b)

Figure 11: Comparison between mpCPR (a) and CR (b) presented on several artificial helices.

jection space, which leads to incorrect visibility as shown in the zoom-in of Fig. 11(a).

A pulmonary data set featuring an embolism is presented in Fig. 12(a) using mpCPR and in Fig. 12(b) using CR. With mpCPR the vessels are not distinctly perceivable, whereas CR preserves the visibility and delineates the embolism clearly. Since the peripheral pulmonary vessel-tree is rather complex, selecting specific vessels might be helpful to locate the embolism.

A data set showing the cervical vessels of a human head is presented in Fig. 1. The use of reformation techniques for cervical vessels is required by the domain experts. A 3D visualization using DVR is presented in Fig. 1(a), the lumen of all vessels is visualized with CR using size-based color-coding and MIP as context in Fig. 1(b) and with halos and MIP as context in Fig. 1(c). Fig. 1(d) and Fig. 1(e) show how to refine a selection. In order to navigate to specific vessels of interest, the vasculature can be filtered using length and thickness. The desired vessels are then selected using visual querying by brushing with the mouse.



(a) (b)

Figure 12: Pulmonary data set with an embolism highlighted with yellow circles. The lumen is rendered with mpCPR (a) and CR (b).

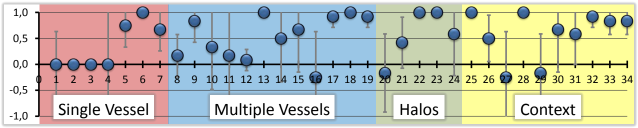


Figure 13: Evaluation of CR with 34 questions. A positive score favours CR, whereas a negative one CPR or mpCPR. The bullets show the average and the error indicators the deviation.

7 Evaluation

The proposed CR technique was evaluated by six domain experts through a questionnaire. They compared CR with CPR and mpCPR in order to address the clinical relevance of visualizing spatially arbitrarily oriented vessels. We specified four key tasks and subsequently summarize them (Fig. 13):

Single vessel visualization. No significant difference between CPR and CR was found when visualizing a single vertical vessel. This part should demonstrate that CR is capable of producing the same results for vertical vessels as the well tested and evaluated CPR [22]. Furthermore, the visualization of the additional organs besides the vessel (surrounding parts) is desired. When using CR, it is helpful to offset the vessel-radius in order to detect calcifications or other suspicious regions.

Multiple vessel visualization. The clinicians have been particularly interested in the visualization of arbitrarily oriented vessels, because they find it helpful to detect pathologies faster or at all. We presented them visualizations of human peripheral pulmonary vessels and cervical vessels. For arbitrarily oriented vessels CR is superior to mpCPR since it preserves the structure of the lumen and accounts for correct visibility (confirmed with the high scores of questions 17, 18 and 19 in Fig. 13).

Halos. Generally, halos are very useful in order to depict spatial relations and enhance depth perception. A small halo size is considered to be sufficient and in the case of visual clutter it is helpful to adjust the halo size.

Context rendering. Comparing MIP and MIDA, the latter one is said to be better, however, both methods have no effect on the perception of the vessels. Halos are also desired when rendering a context and selecting specific vessels is highly favored in order to detect pathologies or reveal them faster.

Our proposed technique got a positive response, especially when investigating vessels of spatially arbitrary orientation. This strengthens the application of CR, because handling such vessels is the main goal of the technique. Enveloping the lumen with halos is also appreciated when investigating a dense vasculature.

8 Conclusion and Future Work

We proposed the novel Centerline Reformation technique for lumen visualization and reformation of spatially arbitrarily oriented and branched vessels. The technique is based on vessel centerlines which are determined by a vessel-detection pipeline and supplemented by an estimation of vessel radii. Subsequently, the centerline and radius information is used in reformation, where vessel, halo and background areas are detected and rendered in different ways. Halos augment the vessel lumen in order to provide better visual cues when visualizing dense vascular structures.

One future aspect is implementing our technique on the GPU to achieve an interactive visualization together with context rendering. Furthermore, geometric methods should be investigated for the visualization of the vessel lumen. This could avoid the early discretization and may lead to advantages when using the GPU.

Acknowledgements

The work presented in this paper is part of the Knowledge Assisted Sparse Interaction for Peripheral CT-Angiography (KASI) project, supported by the Austrian Science Fund (FWF) grant no. TRP 67-N23. Parts of the work were supported by grants no. APVV 20-056105 and VEGA 1/0631/11 (Slovakia). The data sets are courtesy of the Kaiser Franz Josef Hospital and the General Hospital of Vienna.

References

[1] S. R. Aylward and E. Bullitt. Initialization, noise, singularities and scale in height ridge traversal for tubular object centerline extraction. *IEEE Trans. Med. Imaging*, 21(2):61–75, 2002.

[2] S. Bruckner and E. Gröller. Instant volume visualization using maximum intensity difference accumulation. *Computer Graphics Forum*, 28(3):775–782, 2009.

[3] N. D. Cornea, D. Silver, and P. Min. Curve-skeleton applications. In *Visualization, 2005. VIS 05. IEEE*, pages 95–102, 2005.

[4] N. D. Cornea, D. Silver, and P. Min. Curve-skeleton properties, applications and algorithms. *IEEE Trans. Vis. Comput. Graphics*, 13(3):530–548, 2007.

[5] D. Csetverikov. Basic algorithms for digital image analysis. *Course, Institute of Informatics, Eotvos Lorand University*, 2003.

[6] M. H. Everts, H. Bekker, J. B. Roerdink, and T. Isenberg. Depth-dependent halos: Illustrative rendering of dense line data. *IEEE Trans. Vis. Comput. Graphics*, 15(6):1299–1306, 2009.

[7] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. *Lecture Notes in Computer Science*, 1496:130–137, 1998.

[8] H. K. Hahn, B. Preim, D. Selle, and H.-O. Peitgen. Visualization and interaction techniques for the exploration of vascular structures. In *IEEE Visualization*, pages 395–402. IEEE Computer Society, 2001.

[9] V. Interrante and C. Grosch. Strategies for effectively visualizing 3d flow with volume lic. In *Proceedings of the 8th conference on Visualization '97*, VIS '97, pages 421–ff., Los Alamitos, CA, USA, 1997. IEEE Computer Society Press.

[10] R. Jianu, C. Demiralp, and D. H. Laidlaw. Exploring brain connectivity with two-dimensional neural maps. *IEEE Trans. Vis. Comput. Graphics*, PrePrints(99):1–1, 2011.

[11] M. W. Jones, J. A. Bærentzen, and M. Sramek. 3D distance fields: A survey of techniques and applications. *IEEE Trans. Vis. Comput. Graphics*, 12(4):581–599, 2006.

[12] A. Joshi, X. Qian, D. Dione, K. Bulsara, C. Breuer, A. Sinusas, and X. Papademetris. Effective visualization of complex vascular structures using a non-parametric vessel detection method. *IEEE Trans. Vis. Comput. Graphics*, 14(6):1603–1610, 2008.

[13] A. Kanitsar, D. Fleischmann, R. Wegenkittl, P. Felkel, and E. Gröller. CPR - curved planar reformation. In *IEEE Visualization*, pages 37–44. IEEE Computer Society Press, 2002.

[14] A. Kanitsar, D. Fleischmann, R. Wegenkittl, and E. Gröller. Diagnostic relevant visualization of vascular structures. In G.-P. Bonneau, T. Ertl, and G. Nielson, editors, *Scientific Visualization: The Visual Extraction of Knowledge from Data*, Mathematics and Visualization, pages 207–228. Springer Berlin Heidelberg, 2006.

[15] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trousset. Model-based detection of tubular structures in 3D images. *Comput. Vis. Image Underst.*, 80(2):130–171, 2000.

[16] C. C. Law, W. J. Schroeder, K. M. Martin, and J. Temkin. A multi-threaded streaming pipeline architecture for large structured data sets. In *Proceedings of the 10th conference on Visualization '99*, VIS '99, pages 225–232, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

[17] T.-C. Lee, R. L. Kashyap, and C.-N. Chu. Building skeleton models via 3-D medial surface/axis thinning algorithms. *Graphical Models and Image Processing*, 56(6):462–478, 1994.

[18] D. Lesage, E. D. Angelini, I. Bloch, and G. Funka-Lea. A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. *Medical Image Analysis*, 13(6):819–845, 2009.

[19] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Norwell, MA, USA, 1993.

[20] T. Pock, R. Beichel, and H. Bischof. A novel robust tube detection filter for 3D centerline extraction. In *Lecture Notes in Computer Science*, volume 3540, pages 481–490, 2005.

[21] C. Pudney. Distance–ordered homotopic thinning: A skeletonization algorithm for 3D digital images. *Computer Vision and Image Understanding*, 72(3):404–413, 1998.

[22] J. E. Roos, D. Fleischmann, A. Köchl, T. Rakshe, M. Straka, A. Napoli, A. Kanitsar, M. Sramek, and E. Gröller. Multi-path curved planar reformation (mpCPR) of the peripheral arterial tree in CT angiography (CTA). *Radiology*, 244(1):281–290, 2007.

[23] D. Selle, B. Preim, A. Schenk, and H.-O. Peitgen. Analysis of vasculature for liver surgical planning. *IEEE Trans. Med. Imaging*, 21(11):1344–1357, 2002.

[24] J. A. Sethian. Fast marching methods. *SIAM Review*, 41(2):199–235, 1999.

[25] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[26] M. Straka. *Processing and Visualization of Peripheral CT-Angiography Datasets*. PhD thesis, Vienna University of Technology, 2006.

[27] M. Straka, A. Köchl, M. Cervenansky, M. Sramek, D. Fleischmann, A. L. Cruz, and E. Gröller. The VesselGlyph: Focus & Context Visualization in CT-Angiography. In *IEEE Visualization*, pages 385–392, 2004.

[28] S. Svensson, I. Nystrom, and G. Sanniti di Baja. Curve skeletonization of surface-like objects in 3D images guided by voxel classification. *Pattern Recognition Letters*, 23(12):1419–1426, October 2002.

[29] A. Tagliasacchi, H. Zhang, and D. Cohen-Or. Curve skeleton extraction from incomplete point cloud. *ACM Trans. Graph.*, 28:71:1–71:9, July 2009.

[30] A. Telea and J. J. van Wijk. An augmented fast marching method for computing skeletons and centerlines. In *Proceedings of the symposium on Data Visualisation 2002*, VISSYM '02, pages 251–259, Aire-la-Ville, Switzerland, 2002. Eurographics Association.

[31] R. Uflacker. *Atlas of Vascular Anatomy: An Angiographic Approach*. Lippincott Williams and Wilkins, 2006.

[32] H. A. F. G. van Andl, E. H. W. Meijering, A. van der Lugt, H. A. Vrooman, and R. Stokking. Vampire: Improved method for automated center lumen line definition in atherosclerotic carotid arteries in CTA data. In *MICCAI (1)*, pages 525–532, 2004.

[33] A. Varchola, A. Vaško, V. Solčány, L. I. Dimitrov, and M. Šrámek. Processing of volumetric data by slice- and process-based streaming. In H. Slay and S. Bangay, editors, *Afrigraph'07*, pages 101–110, Grahamstown, South Africa, 2007. ACM Siggraph.

[34] A. Wenger, D. F. Keefe, S. Zhang, and D. H. Laidlaw. Interactive volume rendering of thin thread structures within multivalued scientific data sets. *IEEE Trans. Vis. Comput. Graphics*, 10(6):664–672, November 2004.

[35] L. Zhu, S. Haker, and A. Tannenbaum. Flattening maps for the visualization of multibranched vessels. *IEEE Trans Med Imaging*, 24(2):191–198, 02 2005.