



HAL
open science

Automatic localization and quantification of intracranial aneurysms

Sahar Hassan, Franck Hétroy, François Faure, Olivier Palombi

► **To cite this version:**

Sahar Hassan, Franck Hétroy, François Faure, Olivier Palombi. Automatic localization and quantification of intracranial aneurysms. 2011. inria-00598491v1

HAL Id: inria-00598491

<https://inria.hal.science/inria-00598491v1>

Preprint submitted on 6 Jun 2011 (v1), last revised 7 Jun 2011 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Automatic localization and quantification of intracranial aneurysms

Sahar Hassan^{1,2}, Franck Hétroy^{1,2}, François Faure^{1,2}, and Olivier Palombi^{1,2,3}

¹ Université de Grenoble & CNRS, Laboratoire Jean Kuntzmann, Grenoble, France

² INRIA Grenoble - Rhône-Alpes, Grenoble, France

³ Grenoble University Hospital, France

Abstract. We discuss in this paper the problem of localizing and quantifying intracranial aneurysms. Assuming that the segmentation of medical images is done, and that a 3D representation of the vascular tree is available, we present a new automatic algorithm to extract vessels centerlines. Aneurysms are then automatically detected by studying variations of vessels diameters. Once an aneurysm is detected, we give measures that are important to decide its treatment. The name of the aneurysm-carrying vessel is computed using an inexact graph matching technique. The proposed approach is evaluated on segmented real images issued from Magnetic Resonance Angiography (MRA) and CT scan.

1 Introduction

Aneurysms are dilatations in the wall of a blood vessel, leading to little pockets. Aneurysms can be saccular, fusiform or dissecting, see Fig. 1. In this article we are interested in saccular aneurysms which are connected to the vessel by a narrowed zone called the *neck*. If not treated, an aneurysm may burst causing a stroke and in most cases the death of the patient.

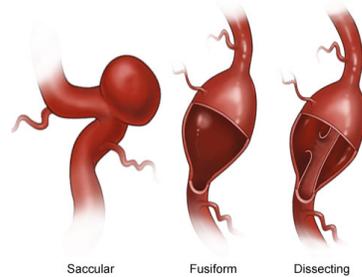


Fig. 1: Aneurysm types⁴.

The decision of treating an aneurysm or just observing it is made according to its risk of rupture. When the treatment is needed, two possible ways exist: either embolization using a platinum coil, or clipping. A lot of studies and statistical surveys have been done in order to know what factors affect the rupture of an aneurysm [1–3], and thus help in making the best decision about the treatment. According to these studies the most important factors are: size, shape, neck, and location of the aneurysm.

A lot of work has been done in the domain of intracranial aneurysms, most of which is about segmenting the vascular tree and giving the user a 3D view of the aneurysm. This segmentation can be statistical [4], or it can be based on the tubular shape of vessels [5–8]. In [9], a morphological characterization of the aneurysm is given in order to predict the rupture rate, and thus decide if there should be a treatment.

In this paper, we suppose that the segmentation is done, and we go further. The set of voxels representing the cerebrovascular tree goes through several processes including: extraction of vessels' centerlines, detection of aneurysms, quantification and

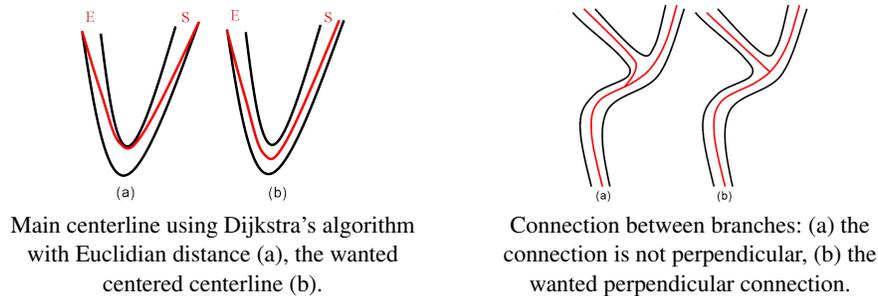
⁴ <http://nyp.org/health/neuro-cerbaneu.html>

localization of the detected aneurysms. An approach based on Dijkstra’s algorithm [10] is proposed to get thin, connected and centered centerlines. These centerlines are then used to study the evolution of the diameters and automatically detect aneurysms. Blood vessels have a cylindrical shape and thus their diameters are almost steady, whereas those of aneurysms change considerably. Relevant measures of found aneurysms and their location are then given using a partial graph matching technique. To our knowledge, this is the first time these steps are performed together to detect, quantify and localize intracranial aneurysms.

2 Methods

2.1 Centerlines extraction

Extraction of blood vessels centerlines can be done either while segmenting these blood vessels [7, 8, 11, 12], or after segmenting blood vessels from medical images as in our case. Various methods for centerline extraction are proposed for different uses. Some categories of these methods are presented in [13] along with the usually desired properties of centerlines. Since we want to use the centerlines to study the evolution of blood vessels diameters, these centerlines should be: **1. connected**: the centerlines we are looking for should be 26-connected, **2. thin**: a centerline is thin if each voxel of the centerline has only two of its neighbors in the centerline, except for the extremities which have one neighbor in the centerline, **3. centered**: the centerlines should be centered within the vascular tree, and **4. connections between branches**: should be as perpendicular as possible, see Fig. 2. Finally, the algorithm should be efficient since it is a step out of four in the processing chain, besides cerebral vascular trees are complex.



Main centerline using Dijkstra’s algorithm with Euclidian distance (a), the wanted centered centerline (b).

Connection between branches: (a) the connection is not perpendicular, (b) the wanted perpendicular connection.

Fig. 2: Important features of the desired centerlines.

In the following, we call *skeleton* the set of centerlines. The longest centerline is called the *main centerline*, while the others are called *branches*. The main centerline and each branch have a diameter which is the mean diameter of the corresponding blood vessels.

To fulfil our requirements, we propose a centerline extraction method that falls in the **distance-based methods** category. The main idea of these methods is to construct a shortest distance tree (SDT) [10]. After the construction of such a tree, we get a graph. Nodes of the graph are the voxels of the object. The voxels will be connected (a connection between two voxels corresponds to an edge in the graph) in a way to minimize the distance to a source voxel S , hereafter called Distance From Source (DFS). The main centerline is then extracted by tracing from E , the voxel with maximum DFS, back to the source S , and thus is connected and thin by construction. The use of a heap for

the priority queue makes the complexity of these methods of $O(N \log N)$ where N is the number of voxels and thus computationally efficient. However, using the Euclidian metric as the distance to minimize leads to a centerline that cuts the corners, see Fig. 2. Several variations of this algorithm were proposed to solve the "cutting corners" problem and get centered centerlines [14–16]. The common idea is to use another distance function while constructing the tree to privilege voxels near the center of the object.

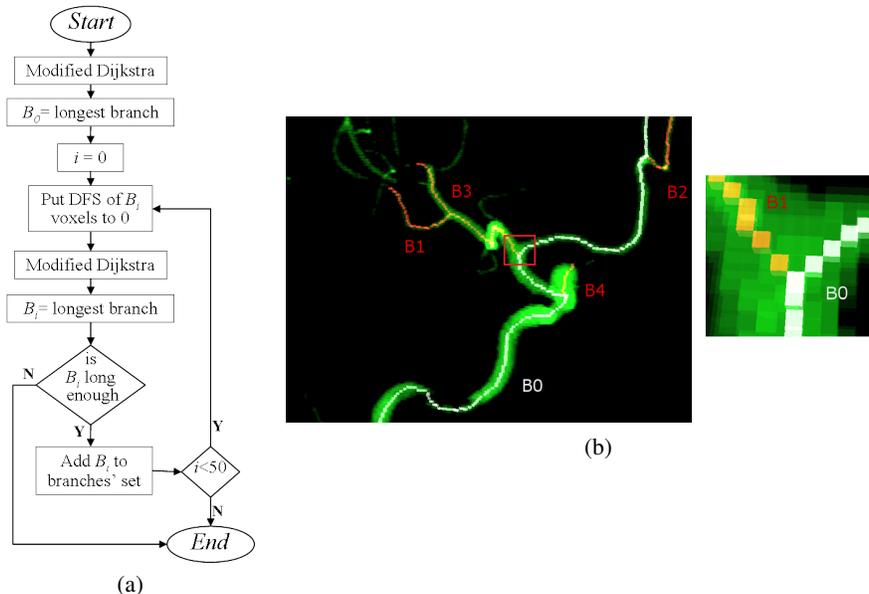


Fig. 3: Our method. (a) Flowchart. 50 is a sufficient number to extract all significant branches in all our experiments. (b) Result of the method on a real dataset with a zoom on the branching.

Our method is illustrated on Fig. 3. **First**, the source voxel is chosen automatically, to be sure that it is an extremity of a vessel. We construct a SDT taking an arbitrary voxel as a source, the end voxel (furthest one of the arbitrary source) is necessarily an extremity and is used as the source voxel for our algorithm.

We use the following distance function instead of using the Euclidian distance:

$$d(v1, v2) = \frac{dist(v1, v2)}{1 + (DFB[v1] + DFB[v2])}$$

with: $dist(v1, v2)$ the Euclidian distance between $v1, v2$, $DFB[v_i]$ v_i 's distance from the boundary, i.e. Euclidian distance between v_i and the closest surface voxel (a surface voxel is a voxel with at least one of its 26-neighbors missing in the voxel set).

The division by the distance from boundary (depth) privileges the voxels that are far from the boundary, and thus enforce the centeredness. At the same time, we keep using the Euclidian distance to find the *end voxel* at each iteration, and thus extract branches in a descending length order. Each branch B_i is connected to a *father* branch that is not necessarily B_{i-1} . Another important advantage of our algorithm is junctions between branches. Putting *DFS* of voxels of extracted branches to zero, makes each branch join its *father* in a perpendicular way (see Fig. 3). We emphasize on this point

because variations of branches' diameters play a major role in aneurysm detection and quantification, see Section 2.2.

The complexity of our algorithm is $O(KN \log N)$ where K is the number of extracted branches, and N is the number of voxels. One drawback of this method is that the set of branches is not homotopic to the object. This method gives by construction a tree-like structure with no loops.

2.2 Automatic detection of the aneurysm

One key characteristic that differentiates a saccular aneurysm from a normal vessel, is that the normal vessel -which has a cylindrical shape- has an almost steady diameter, whereas the aneurysm -which has an irregular shape- has a diameter that changes considerably.

In order to model the appearance of a vessel, we define a set of points (x, y) . Each point corresponds to a voxel v of the branch, where:

- x , represents the distance between the voxel v and the origin of the branch j .
- y , represents the approximate diameter of the branch at v .

To calculate y , we compute the real plane P passing through the center of voxel v and perpendicular to the branch, see Fig. 4. P cuts the vessel or aneurysm surface on voxels $v_i, 1 \leq i \leq k$. Let y_i be the distance between v_i and v , y is defined as the average value of

$$y_i : y = \frac{\sum_{i=1}^k y_i}{k}.$$

Thanks to the centeredness of centerlines, and perpendicular connections between branches, y represents a reliable measure of the diameter changes.

Then, we use the least-squares method to find the quadratic function ($y = a + bx + cx^2$) that best matches our set of points. A more complex function could be used, but this one is sufficient to discriminate between a diameter variation which is linear and a one that is not. Since normal vessels have a cylindrical shape, their diameter is almost steady and thus the value of c is very small. So, by thresholding on c , we decide if the corresponding branch is in an aneurysm. The threshold we use has been found after a ROC analysis, and is 0.2. The threshold is not null because a branch can traverse several blood vessels (see branches in Fig. 3), which makes the associated diameter change. However, this change remains insignificant in comparison with the one caused by an aneurysm.

During the extraction of branches, the above test is made on each branch B_i to decide if it is an aneurysm or not. Branches that are in aneurysms are saved in a list to be treated later for quantification.

2.3 Aneurysm quantification

The construction of a shortest distance tree creates an oriented graph. The nodes of the graph are the voxels. The oriented edges link these voxels together to minimize their distance from the source voxel. Voxels of an aneurysm are the voxels that can be reached from voxels of the *aneurysm branch* by descending the graph. Since the *aneurysm branch* is connected to the *father* branch, which is inside the holding vessel, some of its voxels are inside the holding vessel, see Fig. 6-(a). In order to get rid of

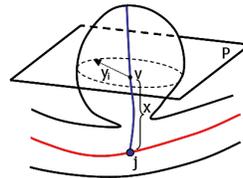
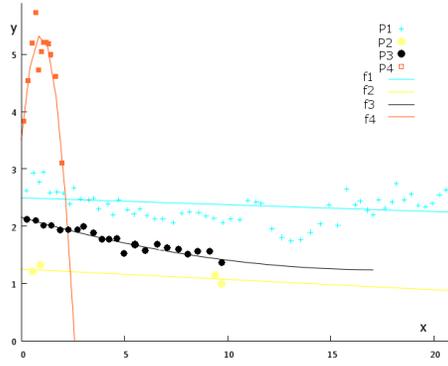


Fig. 4: Calculating y



Branch	a	b	c
B_1	2.494	-0.012	0.000
B_2	1.250	-0.018	0.000
B_3	2.156	-0.105	0.003
B_4	3.509	3.831	-2.006

(a)

(b)

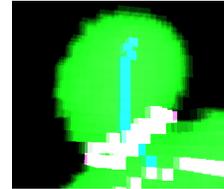
Fig. 5: Diameters variations for branches of the real dataset shown in Fig. 3: (a) The quadratic functions, note that they closely match straight lines for vessels, which is not the case for the one of the aneurysm (B_4). (b) Table 1 shows values of a, b and c for each branch.



(a) In yellow, voxels linked to those of the aneurysmal branch.



(b) The voxels of the aneurysm.



(c) The neck of the aneurysm.

Fig. 6: Compute aneurysm's neck.

these voxels, we only add voxels if their distance from the branch of the holding vessel is greater than its radius, see Fig. 6-(b). The aneurysm's neck is the surface voxels of the aneurysm that have at least one neighbor that is not in the aneurysm, see Fig. 6-(c),(d).

Following a discussion with a surgeon, we found out that the following measures of the aneurysm are relevant to help the treatment decision:

- Size of the aneurysm: number of aneurysmal voxels.
- Maximum vertical diameter of the aneurysm ($Diam1$): to find this diameter, we look for the surface voxel which is the furthest from the origin j of the aneurysmal branch. $Diam1$ is the distance between this voxel and j .
- Maximum horizontal diameter of the aneurysm ($Diam2$): we look for the voxel m of the aneurysmal branch with maximum DFB, then $Diam2 = 2 \times DFB[m]$.

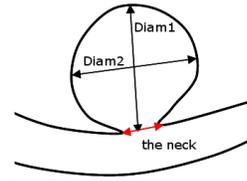


Fig. 7: Measures of an aneurysm.

2.4 Localization of the aneurysm

Regarding the method we use to extract centerlines, the result is a set of branches where each branch B_i (except B_0) has a father branch. On the same time, the branches do not correspond to blood vessels, a branch can be within several blood vessels. To get a graph that represents the resulting tree, we deal with segments. A segment is made of the voxels of a branch between its extremity and a junction, or between two successive junctions. We choose the widest segment (aneurysms excluded) as root, because it corresponds to the carotid (widest blood vessel), and we construct a graph. In Fig. 8-(a), we see the graph corresponding to the dataset of Fig. 5-(a).

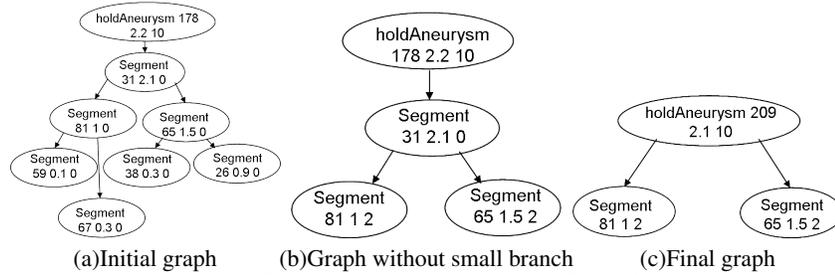


Fig. 8: Graphs for the dataset of Figure5-(a)

Graph matching is a well known problem, and graphs can be with or without attributes for both nodes and edges. If we consider our graph of segments without any attributes, the matching process will be mainly a topological one, meaning that if a node has two child nodes, it may be matched with any node with two children in the reference graph. To get a more accurate matching, we choose to use a graph with attributes.

As can be seen in Fig. 8-(a), we associate to each node of the graph three attributes: length, diameter of the segment, and number of children. The first two attributes are used to give an idea about the importance of the segment. Segments with small diameters or short lengths are considered very patient specific and unimportant. The corresponding nodes are then deleted from the graph (Fig. 8-(b)). We can describe this deletion step as a simplification of the graph. To keep a trace of the deleted nodes, we use the third attribute “number of children”. Each time we decide to delete a node, we increase the number of children of its parent by one. Finally, we give the root of our graph a big number of children (10), to be sure that the root will be matched with the carotid.

Only the third attribute (number of children), is then used in the matching step. It helps to differentiate between vessels that are known to have a lot of bifurcations (vessel M) and those who have less bifurcations (vessel A), and both issued from the same parent (carotid), see Fig. 9.

Since the anatomy of the cerebral vascular tree is known, especially regarding the main vessels, we use a reference graph. In practice, not all vessels are segmented from acquired images, so several reference graphs with different resolutions are needed. Fig. 9 shows the reference graphs we use.

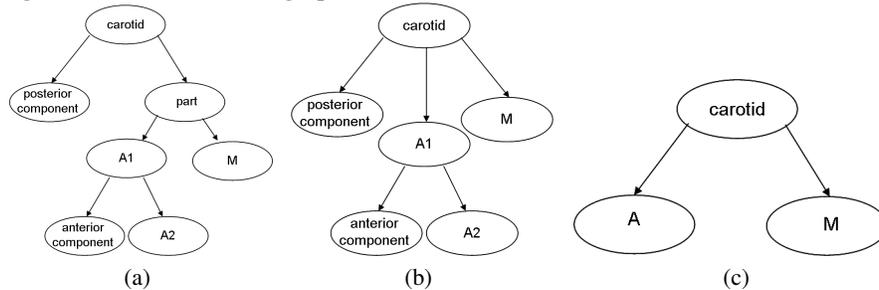


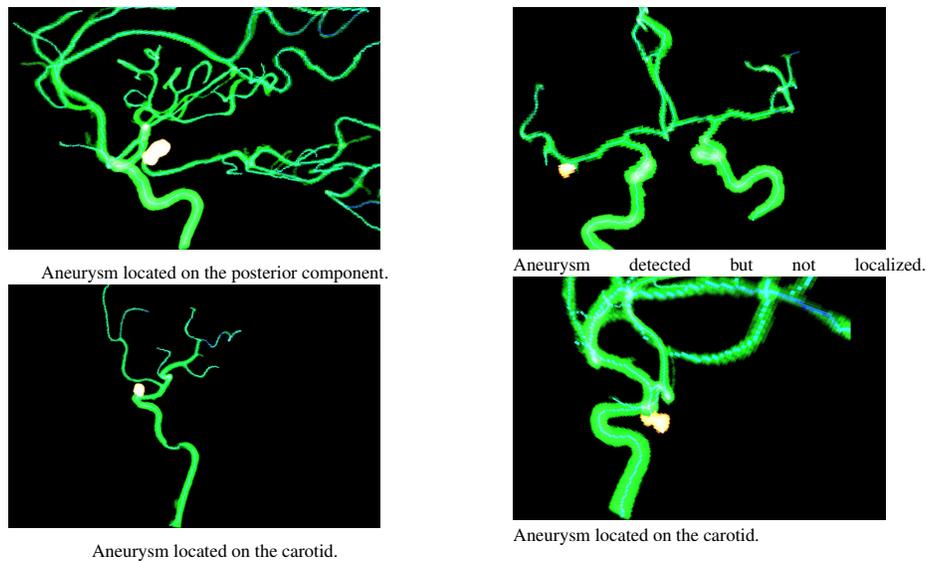
Fig. 9: Reference graphs

The localization of the aneurysm is then reduced to an inexact graph matching problem. We use the VF algorithm [17] to solve this problem. We try first to match our simplified graph with the most detailed reference graph 9-(a), then with 9-(b), and finally with 9-(c). In practice, more reference graphs can be used if needed.

3 Results

We validated our approach on a set of twenty patients, using both MRA and CT imaging techniques for five and fifteen patients respectively. The set contained five males and fifteen females, the patients' ages varied from 33 to 78 years with an average of 51.68.

After segmentation, our method is applied on one connected component (either chosen by the user, or the largest one if no choice is made). The results reported no error of typeI (false negative) and two errors of typeII (false positive). Results of quantifications were compared to those provided by experts (experts provided quantifications for only 10 cases). We use the following formule to calculate the error of a measurement: $E = 100 \times \frac{\|provided - calculated\|}{provided}$. For $Diam1$, the error varied from 0.8 to 48 with an average of 11.7, for $Diam2$, it varied from 1.7 to 17.1 with an average of 8.25.



Aneurysm located on the carotid.

Fig. 10: Some examples of aneurysms detected by our method.

Since our technique of localization does not consider cases where the whole cerebrovascular tree is present, the localisation was possible in ten cases and the localizations were distributed as follows: six aneurysms were localized on the carotid, two on vessel $A1$ and two on the posterior component. Fig. 10 shows some examples of the detected aneurysms. Calculation time on a Pentium(R) 4 CPU 3.00 GHz varied from 4.5 to 145.23 seconds with an average of 29.97. In practice, this time is almost linearly connected to the number of voxels.

4 Conclusion and future work

In this paper, we have presented a complete solution to automatically localize and quantify intracranial saccular aneurysms. First, we use a new distance-based method to find centerlines of the vascular tree. The centerlines are connected, thin (by construction), and centered, due to our modification of Dijkstra's algorithm. Moreover, since the distance map is calculated relative to a source voxel, the presented approach is invariant to rigid transformations. Then, aneurysms are automatically detected and quantified. Finally, the aneurysm is localized by graph-subgraph matching between a graph repre-

senting the centerlines and a reference graph.

When applying our method to 3D medical images, it proved to be fast and robust since the quality of the results is independent of small segmentation artifacts.

References

1. Ujiie, H., Tachibana, H., Hiramatsu, O., Hazel, A., Matsumoto, T., Ogasawara, Y., Nakajima, H., Hori, T., Takakura, K., Kajiya, F.: Effectes of size and shape (aspect ratio) on the heodynamics of saccular aneurysms: a possible index for surgical treatment of intracranial aneurysms. *Neurosurgery* **45** (1999) 119–130
2. Weir, B.: Unruptured intracranial aneurysms: a review. *J Neurosurgery* **96** (2002) 3–42
3. Ecker, R., Hopkins, L.: Natural history of unruptured intracranial aneurysms. *Neurosurg Focus* **17**(5) (2004)
4. Wilson, D.L., Noble, J.A.: Segmentation of cerebral vessels and aneurysms from mr angiography data. In: *IPMI '97: Proceedings of the 15th International Conference on Information Processing in Medical Imaging*, London, UK, Springer-Verlag (1997) 423–428
5. Aylward, S., Pizer, S., Eberly, D., Bullitt, E.: Intensity ridge and widths for tubular object segmentation and description. *Mathematical Methods in Biomedical Image Analysis, IEEE Workshop on* **0** (1996) 0131
6. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale vessel enhancement filtering. *Medical Image Computing and Computer-Assisted Intervention, MICCAI 98* (1998)
7. Wink, O., Niessen, W., Viergever, M.: Multiscale vessel tracking. *Medical Image Analysis* **23**(1) (January 2004) 130–133
8. Descoteaux, M., Collins, D.L., Siddiqi, K.: A geometric flow for segmenting vasculature in proton-density weighted mri. *Medical Image Analysis* **12**(4) (2008) 497 – 513
9. Millán, R.D., Dempere-Marco, L., Pozo, J., Cebal, J., Frangi, A.: Morphological characterization of intracranial aneurysms using 3-d moment invariants. *Medical Imaging, IEEE Transactions on* **26**(9) (2007) 1270–1282
10. Dijkstra, E.W.: A note on two problems in connexion with graphs. *Numerische Mathematik* **1** (1959) 269–271
11. Aylward, S.R., Bullitt, E.: Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. *IEEE Transactions on Medical Imaging* **21**(2) (2002) 61–75
12. Deschamps, T., Cohen, L.: Fast extraction of minimal paths in 3D images and applications to virtual endoscopy. *Medical Image Analysis* **5**(4) (2001)
13. Cornea, N., Silver, D., Min, P.: Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics* **13**(3) (2007) 530–548
14. Bitter, I., Sato, M., Bender, M., McDonnell, K.T., Kaufman, A., Wan, M.: CEASAR: a smooth, accurate and robust centerline extraction algorithm. In: *VIS '00: Proceedings of the conference on Visualization '00*, Los Alamitos, CA, USA, IEEE Computer Society Press (2000) 45–52
15. Bitter, I., Kaufman, A.E., Sato, M.: Penalized-distance volumetric skeleton algorithm. *IEEE Transactions on Visualization and Computer Graphics* **7**(3) (2001) 195–206
16. Wan, M., Liang, Z., Ke, Q., Hong, L., Bitter, I., Kaufman, A.E.: Automatic centerline extraction for virtual colonoscopy. *IEEE Trans. Med. Imaging* **21** (2002) 1450–1460
17. Cordella, L.P., Foggia, P., Sansone, C., Vento, M.: An improved algorithm for matching large graphs. *3rd IAPRTC15 Workshop on Graph-based representations in Pattern Recognition* (2001) 149–159