

Tree matching applied to vascular system

Arnaud Charnoz^{1,3}, Vincent Agnus¹, Grégoire Malandain², Luc Soler¹ and Mohamed Tajine³

¹ Virtual Surg, IRCAD, Strasbourg, France

² Épidaure Research group, INRIA, Sophia Antipolis, France

³ LSIIT, CNRS/ULP, Strasbourg, France

Abstract. In this paper, we propose an original tree matching algorithm for intra-patient hepatic vascular system registration. The vascular systems are segmented from CT-Scan images acquired at different time, and then modeled as trees. The goal of this algorithm is to find common bifurcations (nodes) and vessels (edges) in both trees.

Starting from the tree root, edges and nodes are iteratively matched. The algorithm works on a set of matching hypotheses which is updated to keep best matches. It is robust against topological modification, as the segmentation process can fail to detect some branches.

Finally, this algorithm is validated on the Visible Human with synthetic deformations thanks to the simulator prototype developed at the INRIA which provides realistic deformations for liver and its vascular network.

1. Introduction

1.1. Motivations

Matching and registration are fields in medical imaging with a great impact on visualization, diagnosis and surgery planning. In this paper, we focus on intra-patient follow-up of the hepatic vascular system between two acquisitions. We propose an automatic method which allows to match vessels and bifurcations. This approach is motivated by the fact that the liver is a very high deformable organ. The most reliable landmarks to estimate deformations sustained by the liver are provided by its vascular network.

The principal application of this work is to estimate the deformation of liver between two different times and to make a follow-up of tumors (see previous work [2]).

1.2. Previous works

Related works propose algorithms to match and/or register vascular systems (brain, liver and, in a similar manner, lung airway). Generally, veins are modeled as graphs computed from segmented images and skeletons [8]. Some authors use some tree structure notions in their algorithms to register a tree with an image [1] or two trees [3]. Other approaches really match structures (nodes and vessels),

but use general graph matching methods [9, 4, 5] or too specific methods like subtree isomorphism [7]. To summarize, the vascular tree matching problem is more specific than graph matching because the structure is simpler. On the other hand, it cannot be considered as a subtree isomorphism problem because of the segmentation problems. As a matter of fact, the segmentation process can miss some branches. This implies a (virtual) pruning on both trees, and thus an edge in a tree could be represented by several successive edges on the other tree.

In our previous work [2], vascular systems are modeled as a tree and then graph vertices have been matched together without taking into account possible segmentation errors. The previous algorithm works well on most branches but suffers from a lack of robustness in complex (but real) cases.

1.3. Proposal

The new algorithm proposed in this paper manages a matching hypotheses graph (MHG) where each matching hypothesis is associated with a cost. The MHG is updated as the matched branches set grow. This global approach allows us to find the best match (which minimizes a cost function) and not only a local solution.

The remainder of this paper is organized as follows. The first part presents the tree matching. We describe the generation of hypotheses and their associated cost functions. We explain how we update the MHG by keeping the best potential solutions.

The second part shows results and the algorithm's efficiency. We explain the validation protocol and we discuss tests for virtual and real patient. We finish with a discussion on future possible improvements.

2. Tree matching

The proposed algorithm is a tree matching. Indeed, trees are a representation of skeletons computed from segmented vascular systems. The orientation symbolizes blood circulation flow. Nodes represent bifurcations and edges correspond to vessels between two bifurcations. Vessels has some geometric attributes: 3D positions, radius, vessel path.

Our goal is to find common bifurcations in both trees. Trees represent the same vascular system. However, their topology differ due to segmentation errors as well as 3D positions due to deformations applied on them. Furthermore, we assume that the tree roots are known (detection of vascular system entrance) and that the tree deformations are small (standard case).

In the next sections, we explain this tree matching. After introducing some notations, we see the framework used to generate all matching hypotheses. More specifically, we detail the two steps of the algorithm. Then, we focus on the selection of the best matching hypothesis.

2.1. Notations

We work on a tree noted $T = (V, E, r)$ where V represents the set of vertices, $E \subset V \times V$ the set of edges and r the root. For a node u in a tree T , $T(u)$ denotes the subtree of T induced from u . For a vertex v , $\text{sons}(v)$ denotes the set of their child vertices, and $\text{father}(v)$ its father vertex. For a vertex v , $\text{out}(v)$ denotes the set of out-edges of v , and $\text{in}(v)$ its in-edge. For an oriented edge $e = (v, u)$, we define $\text{src}(e) = v$ and $\text{tgt}(e) = u$. For two vertices $v, w \in V$, $P(v, w)$ is the unique path in T linking v to w . A path is a subtree of T . Let e an edge and its target vertex v , and let $DV_L(e) = \{u, \forall u \in \text{vertices of } T(v), \|P(v, u)\| \leq L\}$ denote the descendant vertex set composed of L -first depth level vertices in subtree induced from e . Let a vertex v , $T_+(v)$ denotes the subtree $T(v)$ where $\text{father}(v)$ is added to vertex set and $\text{in}(v)$ to the edge set.

We introduce also some notations on functions. Let A and B two set with same size. Let $\mathcal{B}_{A,B}$ the bijection class from A to B . Let $\mathcal{C}_{A,B}^k$ the function class which defines a subset of k elements of A and subset of k elements of B . If $h \in \mathcal{C}_{A,B}^k$, then $h(A)$ and $h(B)$ denotes these subsets.

2.2. Framework of the algorithm

Our algorithm searches for the best tree matching between path of $T_1 = (V_1, E_1, r_1)$ and $T_2 = (V_2, E_2, r_2)$ starting from roots (r_1 match with r_2). This algorithm process with a depth first search on T_1 and T_2 . Two successive steps are repeated during the process and different hypotheses are studied (see figure 1): the first step determines the best out-edge matching set from a vertex. The second step determines the next best vertex matching in each out-edge subtree.

As the number of possible solutions is too large, some "bad" hypotheses are eliminated. The difficulty with this approach is the choice of *best match* at each step. This algorithm builds a research tree representing all possible matches where only the most probable configurations are studied.

2.3. Step I: Out-edge matching set hypothesis

Assumption:

Let v_1 and w_1 (respectively v_2 and w_2) be two vertices of V_1 (respectively V_2). $\mathcal{P}(w_1, v_1, w_2, v_2)$ denotes a path matching between $P(w_1, v_1)$ and $P(w_2, v_2)$. Actually, we assume that v_1 and v_2 (respectively w_1 and w_2) represent the same bifurcation in a vascular system. At this step, $T_1(v_1)$ and $T_2(v_2)$ are not yet matched.

Generation:

First, to continue the matching process between $T_1(v_1)$ and $T_2(v_2)$, the best out-edge matches between $\text{out}(v_1)$ and $\text{out}(v_2)$ have to be determined (figure 2). Let $O_1 = \text{out}(v_1)$ and $O_2 = \text{out}(v_2)$. An out-edge matching set hypothesis is noted $\mathcal{H}e(v_1, v_2)$. An hypothesis is represented by an out-edge matching set $\mathcal{E}_f(v_1, v_2)$ which characterizes a match between k elements of O_1 and O_2 . $\mathcal{E}_f(v_1, v_2) =$

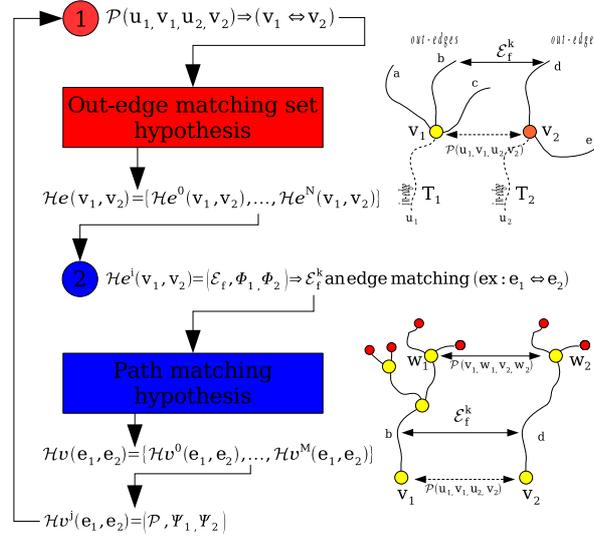


Fig. 1. This figure shows the successive steps of tree matching process and hypotheses generation.

$\{(e, f(e)), \forall e \in h(O_1)\}$ where $f \in \mathcal{B}_{h(O_1), h(O_2)}$ and $h \in \mathcal{C}_{O_1, O_2}^k$. Indirectly, this out-edge matching set assumes that some out-edges of O_1 (respectively O_2) noted $h(O_1)^c$ (respectively $h(O_2)^c$) have no association. Thus, some subtrees have no match in the other graph.

Let $\phi(v, E) = \{T_+(u), \forall (v, u) \in E\}$ the subtree induced by a vertex and a subset of its out-edges. $\phi(v_1, h(O_1)^c)$ represents subtrees starting from v_1 that have no match.

If we assume that $|O_1| \leq |O_2|$, the possible hypotheses are given by:

$$\mathcal{H}e(v_1, v_2) = \{(\mathcal{E}_f(v_1, v_2), \phi(v_1, h(O_1)^c), \phi(v_2, h(O_2)^c))\}, \quad (1)$$

$$\forall k \in [0, |O_1|], \forall h \in \mathcal{C}_{O_1, O_2}^k, \forall f \in \mathcal{B}_{h(O_1), h(O_2)}$$

Combinatory:

When this association rule is respected, all out-edge matching sets can be created. Let $k \in [0, |O_1|]$, the number of possible function h which chooses two subsets with k elements in O_1 and O_2 is $|\mathcal{C}_{O_1, O_2}^k| = C_{|O_1|}^k \times C_{|O_2|}^k$. Moreover, the number of possible bijections between two subsets with k elements is $|\mathcal{B}_{h(O_1), h(O_2)}| = k!$. Thus, the number of out-edge matching set hypotheses is $|\mathcal{H}e(v_1, v_2)| = \sum_{k=0}^{N_{min}} k! C_{|O_1|}^k C_{|O_2|}^k$ where $N_{min} = \min(|O_1|, |O_2|)$.

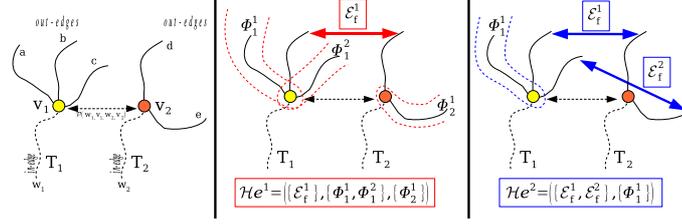


Fig. 2. The figure shows the creation of out-edge matching set hypotheses from a vertex matching. The left illustration resumes previous hypothesis. Other show 2 possible solutions where an out-edge matching set is chosen for each solution. Hypotheses suppose that few out-edges have no their equivalent in other tree and thus that the subtree correspondent is not match.

2.4. Step II: Path matching hypothesis

Supposition:

An out-edge matching, noted $\mathcal{E}_f^i(v_1, v_2) = (e_1, e_2)$, assumes that an edge $e_1 \in O_1$ and an edge $e_2 \in O_2$ match (represent the same start vessel). This step purpose consist in finding the next common bifurcation in subtrees $T_1(tgt(e_1))$ and $T_2(tgt(e_2))$ closest to v_1 and v_2 then we restart at step I. Due to segmentation defects, $tgt(e_1)$ and $tgt(e_2)$ not necessarily represent the same bifurcation. For this fact, we search a vertex matching in subtrees and not only between $tgt(e_1)$ and $tgt(e_2)$ (Fig. 3).

Generation:

The research of next vertex matching is restricted on the L first level of subtrees $T_1(tgt(e_1))$ and $T_2(tgt(e_2))$. Thus, we search the best vertex matching between $DV_L(e_1)$ and $DV_L(e_2)$.

Now, Let (w_1, w_2) a vertex matching with $w_1 \in DV_L(e_1)$ and $w_2 \in DV_L(e_2)$. w_1 are not necessary equal to $tgt(e_1)$ and this vertex matching imply a path matching $\mathcal{P}(v_1, w_1, v_2, w_2) = (P(v_1, w_1), P(v_2, w_2))$. This match also imply that some subtrees starting from $P(v_1, w_1)$ are not matched. We note this forest of no matching subtrees as $\psi(v, w) = \{T_+(u), \forall u \in \text{sons}(k), \forall k \in V_P, T_+(u) \cap P(v, w) = \{k\}\}$ where $V_P = \text{vertices of } (P(v, w))/\{v, w\}$. The set of possible path matching is defined as:

$$\begin{aligned} \mathcal{H}v(e_1, e_2) &= (\mathcal{P}(v_1, w_1, v_2, w_2), \psi(v_1, w_1), \psi(v_2, w_2)), \\ &\forall w_1 \in DV_L(e_1), \forall w_2 \in DV_L(e_2) \\ &\text{with } v_1 = \text{src}(e_1) \text{ and } v_2 = \text{src}(e_2) \end{aligned} \quad (2)$$

Combinatory:

Many path matches can be created. Thus, if we assume that $T_1(tgt(e_1))$ and $T_2(tgt(e_2))$ are complete on the L-first level and if in each bifurcation there

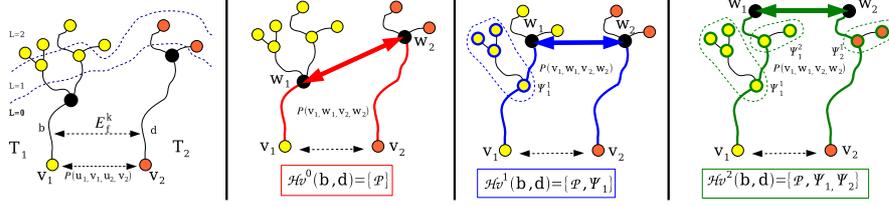


Fig. 3. Figure shows the creation of path matching hypotheses from an out-edge matching. Three solutions are illustrated.

are two out-edges, the number of path matching hypotheses is $|\mathcal{H}v(e_1, e_2)| = \sum_{k=0}^L 2^k \times \sum_{k=0}^L 2^k = (2^{L+1} - 1)^2$

2.5. Hypotheses selection

In the previous sections, we have seen how to generate all matching hypotheses. However, all possible tree matchings can not be explored due to huge combinatorial and only the best hypotheses must be kept. The matching criterion is computed on the current match and only best solutions are kept to explore subgraphs. In fact, we want to minimize a global cost function (sum of local criteria) and discard temporary solutions with high cost. Nevertheless, we cannot accurately compare the same matchings between hypotheses. We have introduced a weight for hypotheses which represents the tree area already processed. It is used to compute a relative cost and thus to compare hypotheses.

In this manner, the n best out-edge matching set hypotheses $\mathcal{H}e^i$ must be selected for the step I and the m best path matching hypotheses $\mathcal{H}v^i$ for step II.

The local cost functions are computed for each hypothesis, and are used to distinguish two hypotheses and keep the best choices.

$$\begin{aligned} \text{cost}(\mathcal{H}e^i(v_1, v_2)) = & \sum_{i=1}^{N_1} \text{cost}(\mathcal{E}_f^i(v_1, v_2)) + \sum_{i=1}^{N_2} \text{cost}(\phi^i(v_1, h(O_1)^c)) + \\ & \sum_{i=1}^{N_3} \text{cost}(\phi^i(v_2, h(O_2)^c)) \end{aligned} \quad (3)$$

$$\begin{aligned} \text{cost}(\mathcal{H}v^i(e_1, e_2)) = & \text{cost}(\mathcal{P}(v_1, w_1, v_2, w_2)) + \sum_{i=1}^{N_1} \text{cost}(\psi^i(v_1, w_1)) + \\ & \sum_{i=1}^{N_2} \text{cost}(\psi^i(v_2, w_2)) \end{aligned}$$

In these equations, we can observe three types of cost: a matching cost between two out-edges, a cost between two paths and a cost for subtrees which are no matched. In next sections, we detail each costs.

Out Edge Matching Cost:

To simplify notation, we note an out-edge matching cost $\text{cost}(\mathcal{E}_f^i(v_1, v_2)) =$

$oemc(e_1, e_2)$. Remember that an edge e represents a vessel between two bifurcations. In the following expression costs, $e(t)$ is the 3D parametric curve representation of the vessel, $r(t)$ represents the vessel's radius along the curve and l is the curve's length. With $oemc$, we compare edge orientation and edge radius.

$$oemc(e_1, e_2) = \int_0^{l_{min}} \|e_1(t) - e_2(t) + e_1(0) - e_2(0)\|^2 dt + \gamma \int_0^{l_{min}} \left\|1 - \frac{r_1(t)}{r_2(t)}\right\|^2 dt \quad (4)$$

Path Matching Cost:

$P(v_1, w_1)$ denotes a path composed of successive edges (vessels). However to simplify notations, we note $P(v_1, w_1) = e_1$ where e_1 represents a virtual edge. Notations become $cost(\mathcal{P}(v_1, v_2, w_1, w_2)) = pmc(e_1, e_2)$. In this cost, weights are added to favor path with same length and short paths, as there are many common nodes in both studied subtrees.

$$pmc(e_1, e_2) = \left(1 + \alpha \frac{l_{max}}{l_{min}} + \beta \frac{l_1 + l_2}{2}\right) \times \left(\gamma \int_0^1 \left\|1 - \frac{r_1(t \times l_1)}{r_2(t \times l_2)}\right\|^2 dt + \int_0^1 \|e_1(t \times l_1) - e_2(t \times l_2) + e_1(0) - e_2(0)\|^2 dt\right) \quad (5)$$

No Matching Tree Cost:

We have previously considered a cost for no inclusion subtree in the matching solution. We have noted these costs $cost(\phi^i(u, E))$ and $cost(\psi^j(u, v))$. These subtrees $T_+(w)$ are defined by a vertex w . To simplify notations, we replace the previous expression cost by $nmtc(w)$. This cost is very important and the choice for weighs is difficult. If this cost is too high then all nodes are matched and conversely, if it is too low, we have no selected match.

$$nmtc(v) = \left(1 + \delta \frac{|T(v)|}{|T|}\right) \times pmc(e, g(e)) + \sum_{k=1}^{|\text{sons}(v)|} nmtc(w_k)$$

with: $g(e) = e$ but with $r(t) = R_{min}$ minimum radius to vessel segmentation (6)

3. Experiments and validation

3.1. Validation protocol on virtual patients

To test and validate our algorithm, we have worked on a liver and its hepatic vascular system. To work on a complex vascular system (280 nodes), the Visible Man (cf. The Visible Human Project of the NLM) has been segmented.

To simulate deformations, we have used the minimally invasive hepatic surgery simulator prototype (Fig. 4) developed at the INRIA [6]. The goal of this simulator is to provide a realistic training framework to learn laparoscopic gestures. For this paper, we used it only to simulate deformations of the liver and its vascular system. This simulator uses complex biomechanical models, based on linear elasticity and finite element theory which include anisotropic deformations.

To simulate segmentation errors on our phantom, we have pruned random tree branches. It's more probable to loose small vessels than to loose large vessels.

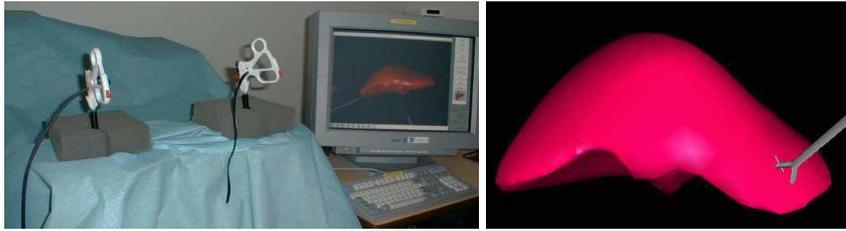


Fig. 4. [Left] Surgery simulator prototype developed by INRIA. [Right] Modeling a contact between a surgical tool and the liver soft tissue model.

3.2. Results on a virtual patient

The results on a virtual patient are good (figure 6) and fast (about 4 minutes to register 380 nodes on 1GHz PC). We have realized 10 different deformations on the Visible Man's liver. For each deformation, 50 random prunings are computed to lost approximately 20% of surface branches in both trees (figure 5). We match

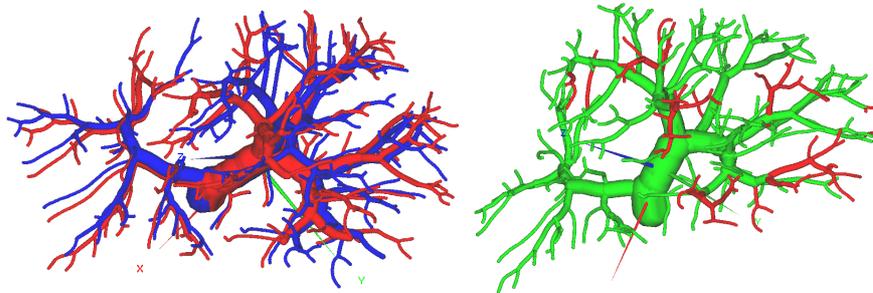


Fig. 5. [Left] Example of small deformations realized with the simulator. [Right] Example of a pruning representing 20% of the surface tree.

90% of all common nodes. The most part of matching errors (incorrect node correspondences and lost branches) is localized on terminal edges. On these

nodes, the algorithm suffers from a lack of information (no subtree, dense node concentrations, small vessels). This make the matching task harder.

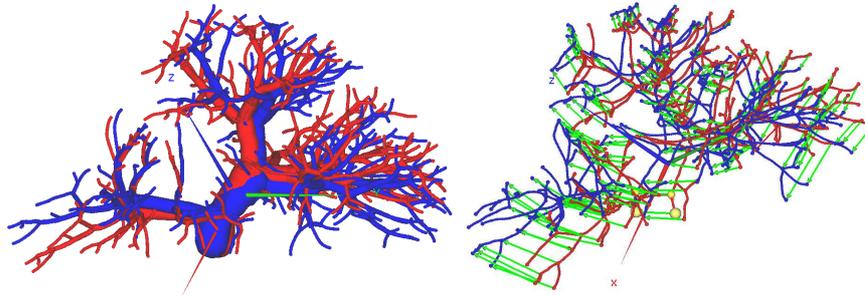


Fig. 6. [Left] Deformation and pruning on the Visible Man computed by the INRIA simulator. [Right] Figure shows result of our oriented tree matching, match are represented by arrows and represent 90% of all nodes.

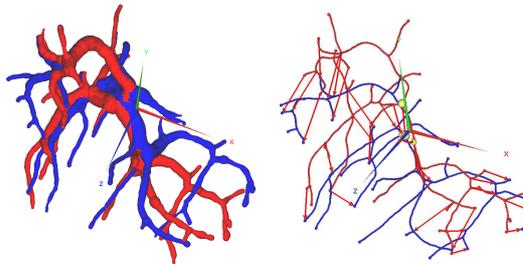


Fig. 7. [Left] Real patient where the vascular system has been segmented between two acquisitions. [Right] Figure shows result of our oriented tree matching, match are represented by arrows and represent 95% of all nodes.

3.3. Results on a real patient

We have tested our algorithm on a real patient between two acquisitions during his therapy. The trees are simpler for our virtual patient and cost weights have been modified in order to get better matches. However, the result is good and promising for the next validation on a database of real patients: we have matched 95% of all common nodes (figure 7).

4. Conclusions and future work

We have presented an original new method to match vascular system between two acquisitions with a tree matching. This method is specific, fast and robust on a complex vascular system. The early stage of validation is very encouraging:

most nodes are matched correctly. Thanks to the virtual database generated by the INRIA simulator we could test several configurations. Nevertheless, a lot of work needs to be done.

Presently, we concentrate our efforts on the design of cost function and its relative cost weights to get an algorithm more robust on large deformations. We will soon propose to apply the estimated deformations on a subtree of matched nodes to superimpose them.

In parallel, we will validate our works on a real patients database with the collaboration of Strasbourg hospital and also propose a new tool for automatic diagnosis of tumors evolution in the liver. The graph matching algorithm could ease the vessel segmentation process by detecting missed branches on the other graph.

Acknowledgments We thank the Strasbourg hospital and their surgeons for providing images as well as their advices on “standard” deformations applied on the liver. This work have benefitted from the segmentation program of vascular system developed by the IRCAD R&D team. The realistic liver deformations are provided by the INRIA simulator from the Epidaure project. Many thanks to Clément Forest for his assistance during the use of the simulator.

References

1. S.R. Aylward, J. Jomier, S. Weeks, and E. Bullitt. Registration and analysis of vascular images. *IJCV*, 55(2-3):123–138, 2003.
2. A. Charnoz, V. Agnus, and L. Soler. Portal vein registration for the follow-up of hepatic tumours. In *MICCAI*, volume 3217 of *LNCS*, pages 878–886, Saint-Malo, France, September 2004. Springer Verlag.
3. T. Lange, S. Eulenstein, M. Hunerbein, H. Lamecker, and P.-M. Schlag. Augmenting intraoperative 3d ultrasound with preoperative models for navigation in liver surgery. In *MICCAI*, volume 3217 of *LNCS*, pages 534–541, Saint-Malo, France, September 2004. Springer Verlag.
4. Y. Park. *Registration of linear structures in 3-D medical images*. PhD thesis, Osaka University, Japan. Departement of informatics and Mathematical Science, 2002.
5. M. Pelillo, K. Siddiqi, and S.W. Zucker. Matching hierarchical structures using association graphs. *PAMI*, 21:1105–1120, November 1999.
6. G. Picinbono, J-C. Lombardo, H. Delingette, and N. Ayache. Improving realism of a surgery simulator: linear anisotropic elasticity, complex interactions and force extrapolation. *JVCA*, 13(3):147–167, july 2002.
7. C. Pisupati, L. Wolff, W. Mitzner, and E. Zerhouni. Tracking 3-d pulmonary tree structures. In *MMBIA*, page 160. IEEE Computer Society, 1996.
8. L. Soler, H. Delingette, G. Malandain, J. Montagnat, N. Ayache, J.-M. Clément, C. Koehl, O. Dourthe, D. Mutter, and J. Marescaux. A fully automatic anatomical, pathological and fonctionnal segmentation from ct-scans for hepatic surgery. In *Medical Imaging*, SPIE proceedings, pages 246–255, San Diego, February 2000.
9. J. Tschirren, K. Palágyi, J.M. Reinhardt, E.A. Hoffman, and M. Sonka. Segmentation, Skeletonization, and Branchpoint Matching - A Fully Automated Quantitative Evaluation of Human Intrathoracic Airway Trees. In *MICCAI*, volume 2489, pages 12–19. Springer-Verlag Heidelberg, 25 September 2002.