Matching of Tree Structures for Registration of Medical Images

Jan Hendrik Metzen[1], Tim Kröger[2], Andrea Schenk[2], Stephan Zidowitz[2], Heinz-Otto Peitgen[2], and Xiaoyi Jiang[3]

[1] University of Bremen, Faculty of Mathematics and Computer Science, Robert Hooke Str. 5, 28359 Bremen, Germany
jhm@informatik.uni-bremen.de
[2] MeVis Research GmbH, Universitaetsallee 29, 28359 Bremen, Germany
[3] University of Münster, Faculty of Mathematics and Computer Science, Einsteinstraße 62, 48149 Münster, Germany

Abstract. Many medical applications require a registration of different images of the same organ. In many cases, such a registration is accomplished by manually placing landmarks in the images. In this paper we propose a method which is able to find reasonable landmarks automatically. To achieve this, nodes of the vessel systems, which have been extracted from the images by a segmentation algorithm, will be assigned by the so-called association graph method and the coordinates of these matched nodes can be used as landmarks for a non-rigid registration algorithm.

1 Introduction

Medical imaging methods like computed tomography (CT) and magnetic resonance imaging (MRI) are able to provide three-dimensional, digital images of organs like liver or lung. In many medical applications, it is desirable to provide the user different images of the same organ. For instance, this might be reasonable if a lung shall be examined both in the inhaled and exhaled state or if there are CT as well as MRI images of the same organ. Another possible application is the monitoring of an organ over a long time period by regularly scanning the organ.

Because of respiration, heartbeat etc. it is possible that the position and shape of an organ might considerably differ between two scannings. That makes it difficult to detect regions in the images which depict the same part of the organ. Such a mapping between different images of an organ is called *registration*. Following Hill [1], registration is the process of transforming different image data sets into one coordinate system.

In order to allow an automatic registration of image data sets it is necessary to use properties of the organs which are invariant against respiration, heartbeat etc. The vessel system of the organs is one possibility for such an invariant feature. The position and extension of these vessel systems might change but their structure remains (nearly) constant. The identification of corresponding areas

in these structures provides structural information, which eases the registration of the image data sets. The vessel systems of liver and lung (e. g. portal vein of the liver and bronchi respectively) are trees and it is therefore possible to apply structural pattern recognition methods to this problem of *matching tree structures*.

2 State of the Art

There are different approaches for matching of tree structures. We briefly sketch four of them:

Pelillo et al. [2, 3] used the so-called association graph for detecting maximal subtree isomorphism of rooted and free trees. Possible assignments of tree nodes are represented as nodes of the association graph. Two nodes of the association graph are connected via an edge if the corresponding assignments are consistent. Two assignments are considered to be consistent if the topological relationship between the two involved nodes in both trees is equal. The definition of this topological relationship differs for matching of rooted and free trees; for free trees it is exactly the topological distance[1] of two nodes while for rooted trees the difference of node levels[2] additionally has to be equal. In the derived association graph a maximal clique is detected by applying pay-off monotonic dynamics from evolutionary game theory on a continuous formulation of the problem obtained by the Motzkin-Straus theorem [4].

Bartoli et al. [5] and Pelillo et al. [6] proposed an extension of the association graph approach to achieve many-to-one and many-to-many matchings of attributed trees. Many-to-many matching means that a group of nodes can be assigned (contracted) to a single node in the other tree while many-to-one matching means that this relationship holds only in one direction. The latter might be adequate when matching a tree to a model. For the purpose of many-to-many matching, each node is rated with a value $r \in \mathbb{R}^+$ which depends only on its attributes. A group of nodes can be contracted if the ratings of all but one node fall below a certain threshold.

Tschirren et al. [7] proposed a method for matching of human airway trees. They first perform a pruning step on the trees in order to improve their comparability and subsequently a rigid registration in order to map the trees into the same coordinate system. Thereafter, a hierarchical approach using an association graph is applied to the data to accomplish a matching. While this approach performs well for some input trees, there are two major drawbacks: The method needs robust ways of detecting major branchpoints in the trees and relies on the invariance of the topological distance. The former may be possible in airway trees but proved to be difficult in liver vessel systems. The latter is susceptible to erroneous segmentation due to noise (see Figure 1).

[1] The topological distance is defined as the number of edges, which have to be traversed on the unique path from one node to another.

[2] The level of a tree node is defined as the number of edges, which have to be traversed on the unique path from a node to the root of the tree.

Charnoz et al. [8] proposed an algorithm, which performs a parallel depth first search on both trees. During this process, a set of matching hypotheses is generated. All matching hypotheses contained in this set are rated and the global optimal matching is chosen. Because of the exponential number of potential matching hypotheses it is crucial to study only the most promising hypotheses. Hence, on each step of the depth first search only a certain number of those node assignments is considered, whose node attributes are similar. This selection based on local properties is risky since the global optimal matching may contain assignments of nodes whose attributes differ significantly. If one correct assignment of nodes close to their root is missed, the whole generated set of matching hypotheses may be significantly flawed. The approach attained promising results when matching one tree segmented from the Visible Mans liver with a perturbed version of itself. Nevertheless, the applicability to trees segmented from real patient data with significant differing topology and node attributes (see Section 3) remains open.

In this work, we will propose an enhanced version of the association graph approach. The association graph approach proposed by Pelillo et al. [2, 3] has been applied successfully to matching of shock trees and shape-axis trees. When applying it to matching of anatomical vessel trees, there are a few additional issues which have to be considered:

- Due to noise and motions of the recorded organs, there will nearly always be errors in the extracted tree structures: Noise, for example, might result in additional branches, which do not exist in the real organ. An additional branch can influence topological distances as well as the level of nodes (see Figure 1) .
- Since the resolutions of CT and MRI have increased continuously in the recent years, the tree structures might be quite big, i. e. have up to 1000 nodes.

A many-to-many matching as described by Bartoli and Pelillo [5, 6] might in principle deal with the first issue; however, it is not clear how to obtain a rating based on node attributes that has small values for exactly those nodes, which were erroneously detected.

Therefore, the described approaches which use an association graph are not well suited for the purpose of matching of anatomical vessel trees. The main contribution of this work is to enhance the association graph approach so that the method can deal with the issues mentioned above. Consequently, it is adequate for the matching of anatomical vessel trees.

3 Methods

3.1 Association Graph

In this section, we will present our enhanced version of the association graph approach. We start with the definition of the enhanced tree association graph, and explain it in more details later on:

Definition 1 (Tree Association Graph). *Let* $T_1 = (V_1, E_1, w_1)$ *and* $T_2 = (V_2, E_2, w_2)$ *be two rooted trees. We define the* tree association graph $G = (V_A, E_A)$ *of* T_1 *and* T_2 *with respect to a set of unary constraints* C_F *and a set of binary constraints* C_G *as:*

1. $V_A = \{v_a \in V_1 \times V_2 | \sum_{f_i \in C_F} \omega_i f_i(v_a) \geq 0.5\},\ \sum_i \omega_i = 1,\ \omega_i \in [0,1]$
2. $E_A = \{(v_a, v_b) \in V_A \times V_A | \sum_{g_j \in C_G} v_j g_j(v_a, v_b) \geq 0.5\},\ \sum_j v_j = 1,\ v_i \in [0,1]$

An interpretation of this definition is as follows: A node $v_a = (v_{a1}, v_{a2}) \in V_A$ represents the potential assignment of the tree nodes v_{a1} and v_{a2}. Consequently, the set V_A is the set of all promising assignments of nodes in V_1 to nodes in V_2. An assignment is considered to be *promising* if it fulfills a set of unary constraints, formalising similarity measures for two nodes, to a certain extent. Each unary constraint measures the similarity of two nodes and rates this similarity with a value between 0 and 1. If the rating is close to 1, the two nodes are nearly indistinguishable for this constraint, while two nodes with rating close to 0 possess only little similarity. Furthermore, each of these constraints has a parameter which determines its selectivity[3]. Since there are different unary constraints and not all of them might be equally decisive, each unary constraints is weighted with a factor $\omega_i \in [0,1]$. The sum of all weights has to be 1. An assignment is promising if the average weighted sum of the rating of all unary constraints for this assignment is greater than or equal to 0.5. Hence, in contrast to the original definition [2, 3, 5, 6] of an association graph $V_A \neq V_1 \times V_2$ but $V_A \subsetneq V_1 \times V_2$. The reduction of the cardinality of V_A enables us to apply the association graph approach to trees with a great number of nodes. A collection of unary constraints is introduced in Subsection 3.2.

Two nodes $v_a = (v_{a1}, v_{a2})$ and $v_b = (v_{b1}, v_{b2})$ of an association graph are connected via an edge e iff the assignments $v_{a1} \leftrightarrow v_{a2}$ and $v_{b1} \leftrightarrow v_{b2}$ are *consistent* to each other. Two assignments are considered to be consistent if they fulfill a set of binary constraints, which formalise consistency measures for two assignments, to a certain extent. Analogue to unary constraints, each binary constraint gives a rating between 0 and 1, is controlled by a selectivity parameter, and is weighted with a factor v_j. Two assignments are consistent if the average weighted sum of the ratings of all binary constraints for the corresponding association graph nodes is greater than or equal to 0.5. Some binary constraints are proposed in Section 3.3.

The aim of matching of tree structures is to determine a set of node assignments of maximum cardinality in which each two assignments are pairwise consistent. Such a set corresponds directly to a maximum clique[4] in the association graph. Since detecting a maximum clique of a graph is known to be

[3] "Selectivity" means how similar two nodes must be to be rated with a value greater than 0.5.

[4] Given an arbitrary undirected graph $G = (V, E)$, a subset of vertices $C \subseteq V$ is called a clique if all its vertices are mutually adjacent; a clique is said to be maximum if there is no other clique with higher node cardinality in the graph.

$\mathcal{NP}$-hard [9], we have to apply approximate methods. As discussed in Section 2, Pelillo gives a promising approach for approximating the maximum clique based on applying pay-off monotonic dynamics from evolutionary game theory on a continuous formulation of the problem obtained by the Motzkin-Straus theorem [4]. This approach has been successfully adopted to this problem.

3.2 Unary Constraints

Unary constraints detect promising assignments of tree nodes in an early stage of the matching process. Obviously, two nodes which might be assigned should be similar. In this context, similarity means that some local properties of the nodes should differ only to a small amount. Examples of such local properties are:

- The level of a tree node.
- The length (or diameter) of the discharging edge of a node. The discharging edge is the unique incident edge of a node, whose other endpoint is a node with minor level. The length of an edge has been computed during the segmentation process and is defined as the length of the anatomical vessel, which corresponds to the edge.
- The size of the induced subtree of a node. The induced subtree of a node is that part of the tree which is rooted in this node. The size of a subtree is defined as the sum of the length of all edges of this subtree.
- The spatial coordinate of a node.

Unfortunately, all of these properties are perturbed by noise, movements of the organs and resultant errors during the segmentation and extraction of the tree structure. For example, the spatial coordinate of a node in the lung is heavily influenced by respiration. The size of a subtree depends among other things on the resolution of the medical imaging method. Methods with higher resolution are able to detect more subtle parts of the tree structure, which increase the size of a subtree. Noise might cause the erroneous detection of a node, which can influence the level of nodes as well as the length of edges (see Figure 1).

Thus, these local properties proposed above are no good choices for unary constraints (as shown by the results presented in Section 4) . A more reliable property, though not local, is the spatial course of the unique path from a node v to a reference node[5] r of the tree. This path traverses a set of edges, whose spatial course is described by a sequence of skeleton points. Hence, the spatial course of each path can be described as a polyline $S = [v, v_1, ..., v_n, r]$ in $\mathbb{R}^3$ consisting of the concatenation of the skeleton points of the edges. We will now define a similarity measure for two of these polylines. This similarity measure compares the curve progression but is independent of the length of the polylines. Therefor, each polyline will be normalised first by transforming it as follows:

[5] A reference node is a tree node, which can be reliably detected in both trees. Possibilities for reference nodes are the root or the first major bifurcation node of a tree.

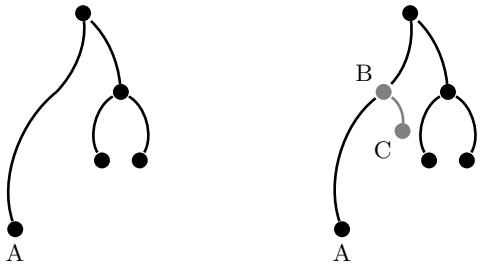

Fig. 1. Erroneously detected nodes influence the level of nodes as well as the length of edges. In this example, the erroneous node C causes an additional bifurcation node B, which splits one edge into two parts. Thus, the level of the leftmost node A has increased by one as well as its topological distance to all other tree nodes. Furthermore the length of its discharging edge has decreased considerably, because this edge has been split into two parts.

1. The polyline will be displaced by $-v$ so that its start point is located in the origin of the coordinate system.
2. Each skeleton point of the polyline will be scaled by $\frac{1}{||r-v||_2}$ so that the start and end point of the polyline will have the euclidean distance 1.
3. The polyline will be rotated in order that its end point will be $x_1 = (1,0,0)^t$. For this purpose, the angle α between the vectors $r - v$ and $x = (1,0,0)^t$ will be computed. Thereafter, the polyline will be rotated by α around the axis, which is orthogonal to $r - v$ and x. This axis is uniquely determined unless $r - v = \pm(1,0,0)^t$. In this case, an axis which is orthogonal to x can be arbitrarily chosen, because α has to be 0 or π.

After this transformation, all polylines $S = [s_0, s_1, ..., s_{n-1}, s_n]$ will begin at $s_0 = (0,0,0)^t$ and end at $s_n = (1,0,0)^t$. We define a partial-polyline as $S_i = [s_0, s_1, ..., s_i]$ and the length of a polyline S_i as $||S_i|| = \sum_{j=1}^{i} ||s_j - s_{j-1}||_2$. Furthermore, we define a parametrisation f of a polyline as follows: $f : [0;1] \rightarrow \mathbb{R}^3$ with $f(\frac{||S_i||}{||S||}) = s_i$, in particular $f(0) = (0,0,0)^t$ and $f(1) = (1,0,0)^t$. The other values of f are linearly interpolated: For $t \in \left]\frac{||S_i||}{||S||}, \frac{||S_{i+1}||}{||S||}\right[$ let $f(t) = (1-\alpha)s_i + \alpha s_{i+1}$ with $\alpha = \frac{t||S|| - ||S_i||}{||S_{i+1}|| - ||S_i||}$.

An obvious similarity measure for two polylines S_1 and S_2 with parametrisation f_1 and f_2 is the integral $\int_0^1 ||f_1(t) - f_2(t)||_2 \, dt$. This integral corresponds to the area between the polylines. However, as shown in Figure 2, this similarity measure is not appropriate since there is still one degree of freedom which affects the value of the integral. Therefore, a better similarity measure is

$$d = \min_{\phi \in [0,2\pi]} d(\phi) \qquad \text{with} \qquad d(\phi) = \int_0^1 ||f_1(t) - A_\phi f_2(t)||_2 \, dt,$$

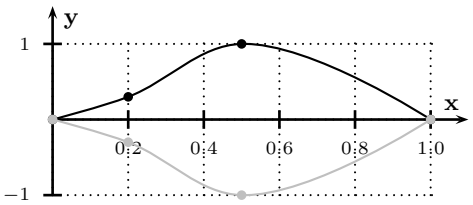

Fig. 2. Two normalized polylines which are to be compared. If the rotational degree of freedom around the x-axis is not considered, even very similar polylines can be rated as very different. In this example, if one of the polylines is rotated by π around the x-axis, the polylines would be identical.

where A_ϕ is a matrix which describes a rotation by ϕ degrees around the x-axis. Unfortunately, this optimisation of the angle ϕ is computationally expensive. This is critical since we have to apply each unary constraints very often[6], which requires the computation of d each time.

Instead of computing the area between two polylines, the directions of the polylines can be compared as basis for a similarity measure, too. Since the polylines are piecewise linear, the derivative $f'(t)$ exists almost everywhere and f' is piecewise constant. For each $t \in [0, 2\pi]$, $g_i(t) = \frac{f_i'(t)}{||f_i'(t)||_2}$ is a unit vector which describes the direction of the polyline's tangent for the parameter value t. We define the similarity of two unit vectors a and b as their dot product $a^t b$. A property of the dot product of two unit vectors is $-1 \leq a^t b \leq 1$ with $a^t b = 1 \Leftrightarrow a = b$. Hence, the similarity of two polylines can be defined as $\frac{1}{2}\int_0^1 (1 + g_1(t)^t g_2(t))dt \in [0, 1]$. The greater the value of the integral, the more similar are the two polylines. However, this similarity measure is influenced by the rotational degree of freedom, too. Fortunately, the optimisation problem

$$\max_{\phi \in [0,2\pi]} \frac{1}{2} \int_0^1 (1 + g_1(t)^t A_\phi g_2(t))dt \tag{1}$$

with a matrix A_ϕ which describes a rotation around the x-axis can be solved analytically [10]. Using $g_i = (g_{i1}, g_{i2}, g_{i3})^t$ the solution is given by

$$\max_{\phi \in [0,2\pi]} \int_0^1 (g_1(t)^t A_\phi g_2(t))dt = C + \sqrt{D^2 + E^2}$$
$$\text{with } C = b_{11},\ D = b_{22} + b_{33},\ E = b_{23} - b_{32} \text{ and } b_{jk} = \int_0^1 g_{1j}(t) g_{2k}(t)dt.$$

[6] If the two trees have n and m nodes respectively, a unary constraint has to be applied $O(nm)$ times.

3.3 Binary Constraints

Binary constraints determine whether two assignments $v_{a1} \leftrightarrow v_{a2}$ and $v_{b1} \leftrightarrow v_{b2}$ are *consistent* to each other, i. e. whether the corresponding nodes of the association graph shall be connected via an edge. First ideas for binary constraints might be:

- The *euclidean distance* d_e of both nodes in the two trees should be similar, i. e. $d_e(v_{a1}, v_{b1}) \approx d_e(v_{a2}, v_{b2})$
- The *topological distance* d_t of both nodes in the two trees should be similar, i. e. $d_t(v_{a1}, v_{b1}) \approx d_t(v_{a2}, v_{b2})$

Unfortunately, the same disturbing factors (noise, respiration, heartbeat) which affect the local properties in Section 3.2 influence the mentioned properties, too. Thus, we have to develop more sophisticated similarity measures. It has turned out that the comparison of the directions of two polylines can be easily transferred onto the binary case and results in a robust constraint: Instead of comparing the two polylines which connect the nodes with the reference nodes of their trees we can simply compare the two polylines which connect v_{a1} and v_{b1}, and v_{a2} and v_{b2}, respectively.

Since binary constraints might be applied very often[7] the comparison of the directions of two polylines can be too expensive for large trees. In this case, the comparison of the *length* $l(v_a, v_b)$ of the connecting path[8] of two nodes v_a and v_b is an option, which can be computed very fast and yields acceptable results. Another similarity measure, which is computationally cheap, is the *curvature* of the connecting path. We define the curvature c of a path connecting two nodes v_a and v_b as $c(v_a, v_b) = l(v_a, v_b)/d_e(v_a, v_b)$.

4 Results

The proposed method has been implemented and tested in the *MeVisLab* research and prototyping platform (`http://www.mevislab.de/`). To provide a basis for the empirical evaluation of the quality of the matchings achieved by the association graph method, we use two matchings as ground truth, which were created manually by human experts. These datasets match a pair of bronchi trees (in inhaleted and exhaleted state) and a pair of portal vein trees (one CT and one MRI image). In each case, both trees have roughly 200 nodes and both ground truths contain 34 assignments of significant nodes distributed all over the trees.

We analyzed the quality of unary and binary constraints. In this section we present mainly the results obtained with the portal vein dataset. Nevertheless,

[7] If the two trees have n and m nodes respectively, in the worst case a binary constraint has to be applied $O(n^2m^2)$ times. The worst case occurs, if most of the possible assignments have been rated as promising by the unary constraints.

[8] The length of a path is defined as the sum of the length of its edges.

Table 1. Comparison of different unary constraints. For each constraint, the selectivity parameter σ has been chosen empirically in order to optimise the resulting matching. Neither the comparison of the diameter of the discharging edge (*EdgeDiameter*) nor the size of the induced subtree (*SubtreeSize*) of a node yield in promising results. Better results are obtained by the non-local constraints. These constraints require the choice of a reference node. For the results presented here, the root has acted as reference node. The runtime was measured on a Pentium4 3.2GHz.

Constraint	σ	Correct	Error	Runtime
EdgeDiameter	1.5	0	3	0.08 sec
SubtreeSize	1.1	4	2	2.36 sec
PathCurvature	1.08	7	3	0.16 sec
PolylineArea	1.04	11	1	14.79 sec
PolylineDirection	1.15	18	4	2.58 sec

the results for the bronchi tree were similar. First, we present a comparison[9] of the different unary constraints proposed in Subsection 3.2. We will use the following names for the different types of constraints (regardless if unary or binary) proposed in Section 3:
EdgeDiameter: Compares the diameter of the discharging edge of a node.
SubtreeSize: Compares the size of the subtree induced by a node.
PathLength: Compares the length of the (unique) path connecting two nodes.
PathCurvature: Compares the curvature of the (unique) path connecting two nodes.
PolylineArea: Compares the area between two (normalized) polylines.
PolylineDirection: Compares the direction of two (normalized) polylines.

As can be seen in Table 1, the best results yield from the *PolylineArea* and the *PolylineDirection* constraints. As expected, the runtime of the *PolylineArea* constraint is (because of the expensive optimization of the angle ϕ) significantly larger than the runtime of the other constraints. Further improvements of the matching can be achieved when combining several unary constraints. It has turned out that it is optimal to combine the *PathCurvature* constraint with the *PolylineDirection* constraint in the proportion $1:4$ (see Table 2).

Similarly, we have analysed the quality of several binary constraints proposed in Subsection 3.3 (in combination with the optimal set of unary constraints, see above). The results are shown in Table 3 and Table 4.

Subsequently, the set of constraints and parameters, which was optimal for the portal vein datasets, has been applied to the bronchi tree datasets to evaluate

[9] Since the proposed algorithm can only be tested with a combination of unary and binary constraints, we show here the results of a particular unary constraint in combination with a specific set of binary constraints. Since we used the same set of binary constraints for all tests, we can compare the quality of the unary constraint. Furthermore, it has turned out that the relative quality of the unary constraints does not depend on the choice of the set of binary constraints.

Table 2. Combination of unary constraints. In parentheses, the weight of the respective constraint is shown. A combination of the *PolylineArea* and the *PolylineDirection* does not improve the quality of the matching. It is likely that this is due to the fact that both constraints assess similar properties and a combination comprises a lot of redundancy. Better results are obtained when combining one of these constraints with the *PathCurvature* constraint. The best result is achieved when combining this constraint with the *PolylineDirection* constraint in the proportion 1 : 4. This proportion is grounded in the fact that the *PolylineDirection* constraint provides better results than the *PathCurvature* constraint when applied solely.

Configuration	Correct	Error
PolylineArea(0.5) : PolylineDirection(0.5)	16	4
PathCurvature(0.5) : PolylineDirection(0.5)	13	2
PathCurvature(0.2) : PolylineDirection(0.8)	20	2
PathCurvature(0.2) : PolylineArea(0.8)	10	4

Table 3. Comparison of binary constraints. For each constraint, the selectivity parameter σ has been chosen empirically in order to optimise the resulting matching. The *PathLength* constraint alone does not yield in good results as it is not distinctive enough. Better results are obtained by the *PathCurvature* and *PolylineDirection* constraints. The *PolylineArea* constraint is omitted here, since its runtime is (due to the large number of applications of a binary constraint) very high and it does not yield in better results than the *PolylineDirection* constraint.

Constraint	σ	Correct	Error	Runtime
PathLength	1.7	2	4	6 sec
PathCurvature	1.3	18	2	8.9 sec
PolylineDirection	1.13	17	2	203.97 sec

Table 4. Combination of binary constraints. The weight of the respective constraint is shown in parentheses. Even though the *PathLength* constraint alone does not seem to be promising, it can improve the results of the *PathCurvature* and *PolylineDirection* constraints. The best results are achieved when combining the *PolylineDirection* with the *PathLength* constraint in the proportion 3:1.

Configuration	Correct	Error
PolylineDirection(0.5) : PathCurvature(0.5)	17	4
PathCurvature(0.75) : PathLength (0.25)	20	2
PolylineDirection(0.75) : PathLength(0.25)	17	0

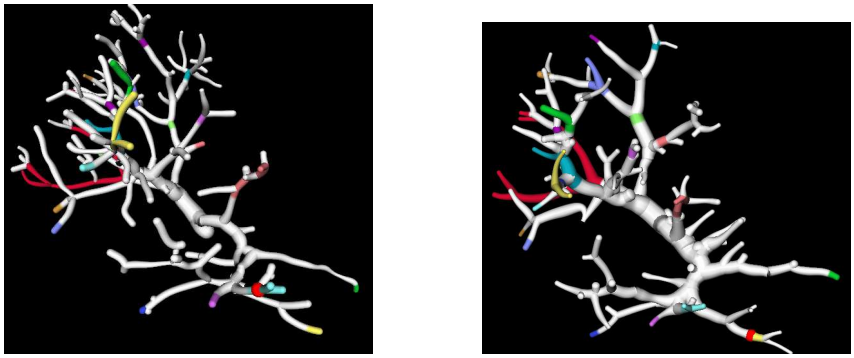

Fig. 3. Matching of two portal veins. Depicted are two portal veins as well as the attained matching. Assigned nodes are dyed in the same colour. If a set of node assignments induces a subtree isomorphism, the whole subtrees are dyed with the same colour.

if the approach performs equally well for datasets originated from other organs. The results are summarised in Table 5, and Figure 3 depicts the matching of the portal vein trees. The quality of both matchings is satisfying, whereas the matching of the portal vein dataset is superior to that of the bronchi trees. Since this discrepancy remains when differing weights and selectivity parameters, the matching of bronchi trees is apperently intrinsically more complex than matching of portal veins. One possible reason is that bronchi trees are dichotomous structures, which means that they have many subtrees which are very similar to each other. An example for such similar subtrees are the right and the left part of a bronchi tree, which split in the first major branchpoint.

Table 5. Results of the matching process: In both cases, the matching algorithms assigned approximately 80 nodes. In case of the portal vein dataset, 17 of these assignments were covered by the ground truth and none of them was incorrect. In case of the bronchi tree, 25 of the assignments were covered by the ground truth and 4 of them were incorrect. Nevertheless, these erroneous assignments match nodes, which are topological neighbours and geometrically at close quarters and therefore, it was even difficult for humans to determine the correct matching of these nodes.

Dataset	Portal Vein	Bronchi Tree
Correct	17	21
Error	0	4
Runtime	207.35 sec	369.23 sec

5 Conclusions

The results indicate that the proposed method is able to achieve good results for typical examples of vessel trees. A significant ratio of tree nodes is assigned in an admissible amount of time. The acquired matching covers most parts of the trees and contains no or only few errors. In our future work, we will analyze if the acquired landmarks are able to improve the registration of the image datasets. Furthermore, it will be examined if the method acquires promising results for harder datasets like images taken during the regeneration of a liver after living liver donation. Also, it will be analyzed if a rigid registration and a hierarchical decomposition of the trees (as described in [7]) can reduce the required computation time.

References

1. Hill, D.L.G., Batchelor, P.G., Holden, M., Hawkes, D.J.: Medical image registration. Physics in Medicine and Biology **46** (2001) R1–R45
2. Pelillo, M., Siddiqi, K., Zucker, S.W.: Matching hierarchical structures using association graphs. IEEE Trans. Pattern Anal. Mach. Intell. **21**(11) (1999) 1105–1120
3. Pelillo, M.: Matching free trees, maximal cliques, and monotone game dynamics. IEEE Trans. Pattern Anal. Mach. Intell. **24**(11) (2002) 1535–1541
4. Pelillo, M.: Replicator equations, maximal cliques, and graph isomorphism. Neural Comput. **11**(9) (1999) 1933–1955
5. Bartoli, M., Pelillo, M., Siddiqi, K., Zucker, S.W.: Attributed tree homomorphism using association graphs. ICPR **02** (2000) 2133–2136
6. Pelillo, M., Siddiqi, K., Zucker, S.W.: Many-to-many matching of attributed trees using association graphs and game dynamics. In: IWVF-4: Proceedings of the 4th International Workshop on Visual Form, London, UK, Springer-Verlag (2001) 583–593
7. Tschirren, J., Mclennan, G., Palagyi, K., Hoffman, E.A., Sonka, M.: Matching and anatomical labeling of human airway tree. IEEE Transactions on Medical Imaging **24**(12) (2005) 1540–1547
8. Charnoz, A., Agnus, V., Malandain, G., Soler, L., Tajine, M.: Tree matching applied to vascular system. In Brun, L., Vento, M., eds.: GbRPR. Volume 3434 of Lecture Notes in Computer Science., Springer (2005) 183–192
9. Karp, R.: Reducibility among combinatorial problems. Complexity of Computer Computations (1972) 85–103
10. Metzen, J.H.: Matching von Baumstrukturen in der medizinischen Bildverarbeitung. Diploma Thesis, University of Münster (2006)