Eurographics Workshop on Visual Computing for Biology and Medicine (2010)
D. Bartz, C. P. Botha, J. Hornegger, R. Machiraju, A. Wiebel, and B. Preim (Editors)

Efficient Globally Optimal Matching of Anatomical Trees of the Liver

Cristina Oyarzun Laura[1] and Klaus Drechsler[1]

[1]Fraunhofer Institute for Computer Graphics Research IGD, Dept. Cognitive Computing & Medical Imaging, Darmstadt, Germany

Abstract

Many inexact automatic tree matching algorithms are nowadays available. However, they provide matches that are not completely error free. Another option is to use manually matched node-pairs, but this enormously slows down the process. Our contribution to the state of the art is to combine the advantages of both solutions. We enhance the automatic tree matching algorithm designed by Graham et al., so that it is possible to interact with it by previously selecting important matches or by subsequently fixing the provided wrong matches. Thanks to this enhancement the speed of the algorithm is greatly increased. It takes 7.45 seconds for trees up to 192 nodes and less than 1 second if three input matches are provided. In addition to this an in-depth evaluation of the robustness of the algorithm is presented. The results are remarkable. The average of wrong matches varies between 1.17 and 1.4 node-pairs in the worst cases. The rate of correct matches is high.

Categories and Subject Descriptors (according to ACM CCS): G.2.2 [Discrete Mathematics]: Graph Algorithms; I.5.3 [Pattern recognition]: Similarity measures; I.4.3 [Image Processing and Computer Vision]: Registration

1. Introduction

The registration of liver volumes gathered from different modalities like Computed Tomography (CT) or Magnetic Resonant Tomography (MRT) is a necessary step for the qualitative and quantitative comparison of pre- with post-operative data to validate the outcome and accuracy of the surgery with a resection plan. One way to carry out the registration is by previously matching corresponding anatomical landmarks between two liver datasets. To find proper landmarks is usually a hard task. Branching points of vessel trees are very often the chosen features as they can be easily identified in datasets from different modalities. Some preprocessing steps must be applied to the data to get the trees that correspond to each vessel tree. First the liver is segmented. Once this is done the vessel trees are also segmented and their skeleton is extracted. The tree is then obtained from that skeleton.

Many automatic-tree matching algorithms have been presented until now. They can be divided into exact and inexact matching algorithms. The former algorithms try to find the maximal subtree isomorphism between both trees. However, in medical imaging we find that both trees are not exactly equal. They present missing branches, extra branches and more than one ramification can be in one of the trees where in the other one there was only one. All these problems that derive from the usage of different segmentation methods, from errors during segmentation and from different resolutions between two modalities, make the use of those algorithms for medical imaging not appropriate. There are authors that decided to adapt them to take all that into account. This is the case of Metzen et al. [MKS*07]. They used the exact matching algorithm designed by Pelillo et al. [PSZ99], which finds the maximal clique of an association graph, introducing a special association graph that deals with all the previously mentioned problems. On the other hand, not all the inexact matching algorithms deal with those problems. This is the case of Gold and Rangarajan [GR96] and Medasani et al. [MKC01]. The proposed algorithms have many similarities. The main difference is that the former tries to match the nodes of both trees and the later the edges.

Finally, there are inexact algorithms specially thought for application in medical imaging. This is the case of Charnoz

DOI: 10.2312/VCBM/VCBM10/075-082

et al. [CAM*05] who search for the best match by defining some hypothesis derived from the attributes of the branches. Tschirren et al. [TMP*05] look for the maximum clique and Kaftan et al. [KKNN06] try to match whole paths instead of nodes or branches. Graham and Higgins [GH06a, GH06b] find the best global match by defining a series of primitives taking into account the aforementioned problems and using a similarity measure based on the attributes of both nodes and branches.

We wanted an algorithm that took into account the problems arose from medical imaging and that presented good results. To this end all the algorithms mentioned in the previous paragraph could be useful. However we did not want to blindly apply an existing algorithm. We wanted to be able to adapt it to our specific problem, namely, liver vessel tree matching with preselection of matches as input. In this sense we found Graham et al.'s algorithm to be a very intuitive and adaptable one, fulfilling everything what we were looking for.

There are currently no automatic tree matching algorithms known that produce no errors. Another option is to select manually corresponding node-pairs, which makes the matching process much slower. However, it is necessary to correct wrongly matched node-pairs. We think that a compromise between fully automatic and manual specification approaches can be a good solution, keeping the speed of the automatic tree matching algorithms, but being able to correct wrong matches. Our contribution to the state of the art is:

- We extended Graham et al.'s automatic tree matching algorithm (Section 2.1) such that it can be applied to trees with ramifications of up to 4 children (Section 2.2). We use the same mathematical notation as Graham et al. to express our extensions.
- We modified Graham et al.'s similarity measure in order to improve the results (Section 2.2).
- We enhanced Graham et al.'s algorithm to take preselected node-pairs into account. As a side effect this greatly reduces the runtime of the algorithm (Section 2.3).
- We present the results of our extensive evaluation of the presented algorithm (Section 3).

We also provide the option to remove or fix matches that are wrong after the automatic matching is applied [DOLCE10].

2. Method

2.1. Graham et al.'s algorithm

Graham et al.'s algorithm is a model based automatic tree matching algorithm that looks for the best possible global match between two trees. The model contains the branches and nodes that are common to both trees. This can be better understood by thinking about the common tree (model) as an initial structure that after undergoing a series of deformations leads to the two trees. The common tree could represent the liver of a patient and both trees the liver of the same patient in different respiratory cycles. Every node in the model corresponds therefore to a match between both trees.

We define the level of a node u, as the number of nodes that form the subtree that has u as root. For example the level of the yellow node in Figure 8(a) is 3. In principle the algorithm visits the node-pairs starting from those that have lower level (leaves), and continues until it reaches the root (node with higher level). For every one of them, it calculates a similarity measure. If all the possible node-pairs were checked, the algorithm would be very inefficient and would not deal with some of the medical imaging deformations that were mentioned above. To avoid that, Graham et al. define a series of primitives to limit the number of matches that can be considered as valid and to take into account the deformations that the common tree can suffer. A pair of nodes will only be candidate to become a match when its corresponding node in the common tree belongs to at least one of those primitives, namely:

1. The planted root.
2. A leaf.
3. A bifurcation point.
4. A trifurcation point.
5. A trifurcation derived from two close bifurcations in both trees where the ordering of the branches is reversed.
6. A trifurcation derived from two close bifurcations in one of the trees and a trifurcation in the other tree.

The goal of the algorithm is to obtain the matches that lead to the highest global similarity measure, since they form the best match between both trees. The similarity measure is calculated using the length attribute of the branches as well as their directions, the coordinates of the nodes and a measure that favors those matches that keep the topology of the tree unchanged. In this sense, the lowest similarity measure for each attribute is obtained when the difference of the attributes of both nodes is above/below a given threshold (e.g. the difference between the lengths of the branches is too big). Graham et al. give a different weight to each attribute depending on the importance that they have in order to obtain the final similarity measure. In addition to this they also take into account the similarity measure of the node-pairs that were calculated in previous steps (their level is lower). They compare then the similarity measure of each node-pair and each primitive and keep the node-pairs that lead to the higher global similarity measure. Detailed information as well as the mathematical expressions for the primitives and the similarity measures can be found in [GH06a, GH06b].

2.2. Adaptation to the current problem

The first change that we apply to Graham et al.'s algorithm is to further limit the number of matches that are considered

as valid. We do this using the length of the path from the root to the node under consideration. If the difference of the path lengths is too big, we consider it as a non valid match.

We also changed the similarity measure. As it was explained before, Graham et al. combine the similarity measures obtained from each attribute giving to them a specific weight to obtain the final similarity measure. We consider that even if the weight of an attribute for the calculation of the final similarity measure is very low, the node-pair is directly considered as wrong match if the difference between those attributes is below/above the threshold, e.g. if the direction attribute of both branches is very similar but the length is very different this match is wrong even if the direction attribute has a higher weight in the similarity measure. With this change the number of wrong matches is reduced.

After inspecting vessel trees obtained from liver CTs, we found out that trifurcations were not enough to cover all the sizes of ramifications that were contained in the tree. That is why we adapted the algorithm and defined seven new primitives, to deal with the new deformations that we found. We will give their mathematical expressions in the same way as Graham et al. did it. Notice that the numbering of our primitives start from 6, as the primitives from 0 to 5 were defined by Graham et al. and can be found in [GH06a]. For our application we need to consider ramifications with up to four children. As Graham et al. were doing it we also assume that there can be cases where two ramifications that are very close together correspond to only one in the common tree. This means that those ramifications and their corresponding branches should be merged.

We will first give a few definitions. $\Psi(T_1^{u_1}, T_2^{\phi(u_1)})$ denotes a primitive that fulfills certain conditions. In this case the subtree T_1 with root u_1 is matched to the subtree T_2 with root $\phi(u_1)$. $\phi(u_1)$ represents the node that is going to be matched to u_1. $\delta(u_1)$ is the degree of u_1, in other words, the number of children of the node u_1 and $lca(u,v)$ denotes the least common ancestor of u and v. $E(T_1)$ is the set of branches of the tree T_1.

The first new primitive corresponds to a deformation where a bifurcation is very close to a trifurcation in one tree and there is a ramification of four nodes in the second tree. In this situation, bifurcation and trifurcation could be merged to form a ramification with four children (Figure 1). Whether or not they are merged in the common tree depends uniquely on the similarity measure. This primitive is defined in the following way:

$\psi_\phi^{u_1} \in \Psi^{(6)}(T_1^{u_1}, T_2^{\phi(u_1)})$ if

(1) $\psi_\phi^{u_1} = \{(u_1,\phi(u_1)),$
$(v_1,\phi(v_1)),(w_1,\phi(w_1)),(x_1,\phi(x_1)),(z_1,\phi(z_1))\}$
(2) $\delta(u_1) = 2$ *in* T_1, $\delta(\phi(u_1)) = 4$ *in* T_2
(3) $lca(v_1,w_1) = lca(w_1,z_1) = y_1$ *in* T_1, *where* $(u_1,y_1) \in E(T_1)$
(4) $lca(v_1,x_1) = lca(w_1,x_1) = lca(z_1,x_1) = u_1$ *in* T_1
(5) $lca(\phi(v_1),\phi(w_1)) = lca(\phi(v_1),\phi(z_1)) =$
$lca(\phi(w_1),\phi(z_1)) = lca(\phi(v_1),\phi(x_1)) =$
$lca(\phi(w_1),\phi(x_1)) = lca(\phi(z_1),\phi(x_1)) = \phi(u_1)$ *in* T_2

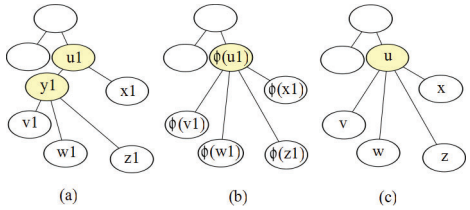

Figure 1: *Primitive 6. (a) and (b) trees to be matched, (c) common tree.*

The second new deformation that our adaptation takes into account is when a trifurcation is very close to a bifurcation in one tree and there is a ramification with four children in the other tree. In this case, trifurcation and bifurcation could be merged resulting on a ramification of four nodes, both in the trees and in the model (Figure 2). We show again its mathematical description:

$\psi_\phi^{u_1} \in \Psi^{(7)}(T_1^{u_1}, T_2^{\phi(u_1)})$ if

(1) $\psi_\phi^{u_1} = \{(u_1,\phi(u_1)),$
$(v_1,\phi(v_1)),(w_1,\phi(w_1)),(x_1,\phi(x_1)),(z_1,\phi(z_1))\}$
(2) $\delta(u_1) = 3$ *in* T_1, $\delta(\phi(u_1)) = 4$ *in* T_2
(3) $lca(v_1,w_1) = y_1$ *in* T_1, *where* $(u_1,y_1) \in E(T_1)$
(4) $lca(v_1,x_1) = lca(w_1,x_1) = lca(z_1,x_1) = lca(v_1,z_1) =$
$lca(w_1,z_1) = u_1$ *in* T_1
(5) $lca(\phi(v_1),\phi(w_1)) = lca(\phi(v_1),\phi(z_1)) =$
$lca(\phi(w_1),\phi(z_1)) = lca(\phi(v_1),\phi(x_1)) =$
$lca(\phi(w_1),\phi(x_1)) = lca(\phi(z_1),\phi(x_1)) = \phi(u_1)$ *in* T_2

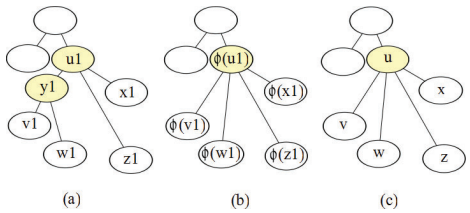

Figure 2: *Primitive 7. (a) and (b) trees to be matched, (c) common tree.*

The next deformation is where a trifurcation is very close to a bifurcation in both trees but their branches are reversed. Both trees could be merged resulting in a common tree with a ramification of four nodes (Figure 3).

$\psi_\phi^{u_1} \in \Psi^{(8)}(T_1^{u_1}, T_2^{\phi(u_1)})$ if

(1) $\psi_\phi^{u_1} = \{(u_1,\phi(u_1)),$
$(v_1,\phi(v_1)),(w_1,\phi(w_1)),(x_1,\phi(x_1)),(z_1,\phi(z_1))\}$
(2) $\delta(u_1) = 3$ *in* T_1, $\delta(\phi(u_1)) = 3$ *in* T_2

(3) $lca(v_1,w_1) = y_1$ *in* T_1, *where* $(u_1,y_1) \in E(T_1)$
(4) $lca(\phi(w_1),\phi(z_1)) = y_2$ *in* T_2, *where* $(\phi(u_1),y_2) \in E(T_2)$
(5) $lca(v_1,x_1) = lca(w_1,x_1) = lca(z_1,x_1) = lca(v_1,z_1) = lca(w_1,z_1) = u_1$ *in* T_1
(6) $lca(\phi(v_1),\phi(w_1)) = lca(\phi(v_1),\phi(x_1)) = lca(\phi(v_1),\phi(z_1)) = lca(\phi(z_1),\phi(x_1)) = lca(\phi(w_1),\phi(x_1)) = \phi(u_1)$ *in* T_2

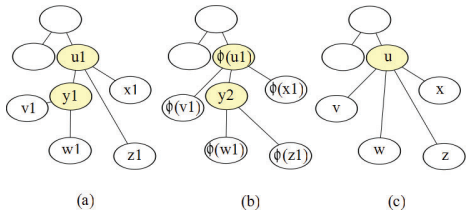

Figure 3: *Primitive 8. (a) and (b) trees to be matched, (c) common tree.*

The next deformation considers when one of the trees has a trifurcation very close to a bifurcation and the other tree has a bifurcation very close to a trifurcation (Figure 4). This primitive is defined as follows:

$\Psi_\phi^{u_1} \in \Psi^{(9)}(T_1^{u_1}, T_2^{\phi(u_1)})$ if

(1) $\Psi_\phi^{u_1} = \{(u_1,\phi(u_1)), (v_1,\phi(v_1)),(w_1,\phi(w_1)),(x_1,\phi(x_1)),(z_1,\phi(z_1))\}$
(2) $\delta(u_1) = 3$ *in* T_1, $\delta(\phi(u_1)) = 2$ *in* T_2
(3) $lca(v_1,w_1) = y_1$ *in* T_1, *where* $(u_1,y_1) \in E(T_1)$
(4) $lca(\phi(v_1),\phi(w_1)) = lca(\phi(v_1),\phi(z_1)) = lca(\phi(w_1),\phi(z_1)) = y_2$ *in* T_2, *where* $(\phi(u_1),y_2) \in E(T_2)$
(5) $lca(v_1,x_1) = lca(w_1,x_1) = lca(z_1,x_1) = lca(v_1,z_1) = lca(w_1,z_1) = u_1$ *in* T_1
(6) $lca(\phi(v_1),\phi(x_1)) = lca(\phi(w_1),\phi(x_1)) = lca(\phi(z_1),\phi(x_1)) = \phi(u_1)$ *in* T_2

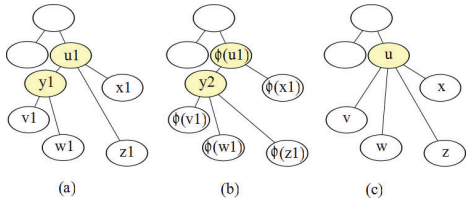

Figure 4: *Primitive 9. (a) and (b) trees to be matched, (c) common tree.*

The next new deformation is where a trifurcation is very close to a bifurcation in one of the trees while the other tree has a trifurcation. Both branches of the bifurcation could be matched to two of the branches of the trifurcation (Figure 5). This means that one of the branches will be a spurious branch. The branch leading to the lowest similarity measure will be considered to be the spurious branch.

$\Psi_\phi^{u_1} \in \Psi^{(10)}(T_1^{u_1}, T_2^{\phi(u_1)})$ if

(1) $\Psi_\phi^{u_1} = \{(u_1,\phi(u_1)), (v_1,\phi(v_1)),(w_1,\phi(w_1)),(z_1,\phi(z_1))\}$
(2) $\delta(u_1) = 3$ *in* T_1, $\delta(\phi(u_1)) = 3$ *in* T_2
(3) $lca(v_1,w_1) = y_1$ *in* T_1, *where* $(u_1,y_1) \in E(T_1)$
(4) $lca(v_1,x_1) = lca(w_1,x_1) = lca(z_1,x_1) = lca(v_1,z_1) = lca(w_1,z_1) = u_1$ *in* T_1
(5) $lca(\phi(v_1),\phi(w_1)) = lca(\phi(v_1),\phi(z_1)) = lca(\phi(w_1),\phi(z_1)) = \phi(u_1)$ *in* T_2

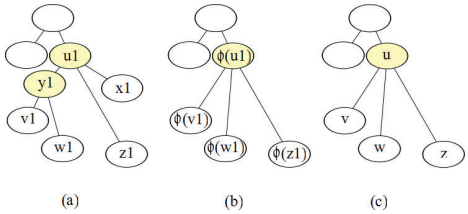

Figure 5: *Primitive 10. (a) and (b) trees to be matched, (c) common tree.*

It is also possible to find the situation where two close bifurcations are in one of the trees and a bifurcation is in the other tree (Figure 6). As in the case before, one of the branches resulting from the merging will be considered as a spurious branch. In that case:

$\Psi_\phi^{u_1} \in \Psi^{(11)}(T_1^{u_1}, T_2^{\phi(u_1)})$ if

(1) $\Psi_\phi^{u_1} = \{(u_1,\phi(u_1)),(v_1,\phi(v_1)),(w_1,\phi(w_1))\}$
(2) $\delta(u_1) = 2$ *in* T_1, $\delta(\phi(u_1)) = 2$ *in* T_2
(3) $lca(v_1,w_1) = y_1$ *in* T_1, *where* $(u_1,y_1) \in E(T_1)$
(4) $lca(v_1,x_1) = lca(w_1,x_1) = u_1$ *in* T_1
(5) $lca(\phi(v_1),\phi(w_1)) = \phi(u_1)$ *in* T_2

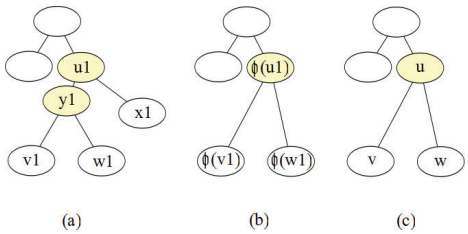

Figure 6: *Primitive 11. (a) and (b) trees to be matched, (c) common tree.*

Finally, a ramification with 4 children can be found in both trees (Figure 7), leading to the following mathematical definition.

$\Psi_\phi^{u_1} \in \Psi^{(12)}(T_1^{u_1}, T_2^{\phi(u_1)})$ if

(1) $\Psi_\phi^{u_1} = \{(u_1,\phi(u_1)), (v_1,\phi(v_1)),(w_1,\phi(w_1)),(x_1,\phi(x_1)),(z_1,\phi(z_1))\}$
(2) $lca(v_1,w_1) = lca(v_1,z_1) = lca(v_1,x_1) = lca(w_1,z_1) = lca(w_1,x_1) = lca(z_1,x_1) = u_1$ *in* T_1

(3) $lca(\phi(v_1),\phi(w_1)) = lca(\phi(v_1),\phi(z_1)) = lca(\phi(v_1),\phi(x_1)) = lca(\phi(w_1),\phi(z_1)) = lca(\phi(w_1),\phi(x_1)) = lca(\phi(z_1),\phi(x_1)) = \phi(u_1)$ *in* T_2

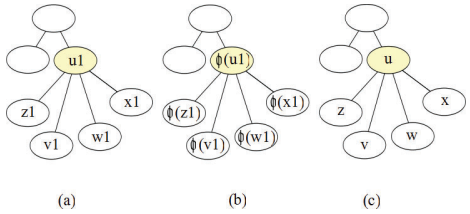

Figure 7: *Primitive 12. (a) and (b) trees to be matched, (c) common tree.*

There are some primitives that have been omitted. The reason for that is that they are indirectly included in some of the primitives introduced by Graham et al.. This is the case for example when a bifurcation is very close to a trifurcation in both trees. This is defined by Graham et al.'s primitive number three.

2.3. Setting input matches

One of the goals of our current work is to let the doctor interact with the automatic tree matching algorithm. We have enhanced Graham et al.'s algorithm so that it is possible for the doctor to set as input some manually selected matches if he considers it necessary. These matches will be always respected by the algorithm which will find the best global tree matching according to those input matches.

We use the example shown in Figure 8 to clarify the process. Figure 8(a) contains the complete trees to be matched. The colored nodes represent the node-pairs that the doctor has selected as input matches. This way, the doctor has decided to match the yellow node of the first tree to the yellow node of the second. As the algorithm starts, we search for those input matches. Being the order to visit the nodes determined by their level (start from the leaves, level 1) the first input node to be found is the yellow (b) one, followed by the red (c), green (d), and blue (e) nodes. The way to proceed when an input match is found is as follows:

1. Generate two subtrees with the found input nodes as roots.
   1.1. Nodes that were contained in previous subtrees are not further considered (e.g. Figure 8(d), the nodes that were checked in (b) are not checked again).
   1.2. Input matches contained in previous subtrees are considered if they do not have an ancestor that is an input match (except the root)(Figure 8(e), the red node is still considered but the yellow node is not because it has an input match (green) as ancestor).
2. Apply adapted Graham et al.'s algorithm with some restrictions (explained below) to get the best match between both subtrees.
3. Store the best similarity measure obtained (it is stored as the similarity measure to match both roots).
4. Store the node-pairs that lead to the mentioned similarity measure.

In subtrees which contain more than one input match it is not possible to blindly apply Graham et al.'s algorithm. For example in Figure 8(d) it is topologically not possible to match node v to node u and at the same time both yellow matches. It must be taken into account that the goal of this subtree-matching is to calculate the node-pairs that lead to the best similarity measure when trying to match both green nodes (current roots). Both green nodes are bifurcations and if the yellow nodes are matched (as it was desired by the doctor), node v can only be matched to z and not to u. There is a yellow line in the figure that joins yellow and green nodes. To avoid that the yellow node-pair is not matched due to other matches that are topologically incompatible with it we define the following rule: "Nodes that do not descend from a line (e.g. v) can not be matched to nodes that descend from a line (e.g. u)". There will be as many independent lines as input matches in the tree. In Figure 8(e) there are red and blue lines. Nodes that descend from the green line in one tree will only be matched to nodes that descend from the green line in the other tree. The nodes in the orange circle can not be matched to any node of the other tree and will not be considered at all.

This enhancement makes the algorithm much faster than the original one (subsection 3.1). On the one hand dividing the tree into different subtrees limits the number of node-pairs that are checked. To continue with the example, the nodes in the first subtree in (b) form node-pairs only with those in the second subtree of (b). This is a big difference with respect to Graham et al.'s algorithm, where node-pairs would be formed with all the nodes of the complete second tree (a). On the other hand the lines between input matches further limit the number of valid matches, as it was explained before.

3. Evaluation

In this section we present an extensive evaluation of the presented algorithm. In subsection 3.1 the efficiency of the algorithm for different sizes of trees is presented. In subsection 3.2 the process of choosing the more appropriate attributes is explained. After that the robustness of the algorithm is evaluated. We evaluate it towards topological deformations (subsection 3.3) and noise addition (subsection 3.4). Finally, two realistic experiments were done with trees where both attributes and topology were slightly different (subsection 3.5). All experiments were executed on a PC equipped with an AMD Athlon 2.41 GHz 64 X2 Dual Core Processor 4600+, 3.25 GB RAM and Windows Vista x64 as operating system. The dataset was acquired with a GE Medical Systems LightSpeed 16 CT Scanner (dimension:

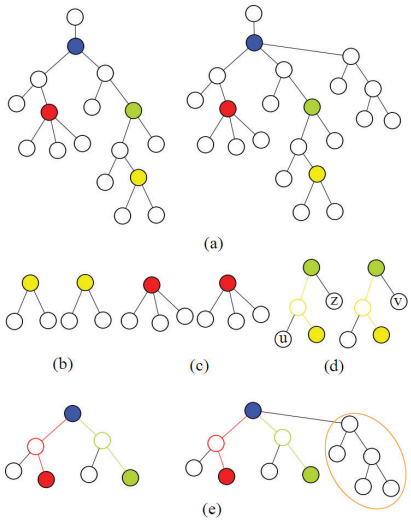

Figure 8: *(a) The complete trees. (b) Subtrees with the yellow node as root. (c) Subtrees with the red node as root. (d) Subtrees with the green node as root. The yellow node is kept but all the nodes that were derived from the yellow in (b) are removed. The yellow imaginary line is respected in the matching. (e) The blue node is the root, red and green nodes are kept but yellow is removed. There are two lines (red and green) to be respected in the matching. The nodes in the orange circle will not be matched.*

512x512x291, spacing: 0.64x0.64x1.0 mm, slice thickness: 1.25 mm).

3.1. Efficiency

The runtime of the algorithm for trees with different number of nodes was measured. Four experiments were performed for every tree pair: one where no input matches were set as input, and three with one, two and three input matches respectively. For the latter experiments, input node-pairs were selected that belong to the main branch, as we think that it is more realistic to believe that the doctor will select those nodes.

3.1.1. Results

Table 1 shows the efficiency of the algorithm. As it can be seen, the algorithm is slower for bigger trees with no input matches (7.45 seconds when both trees have 192 nodes). It is important to notice the big time difference by just selecting one node-pair as input to the algorithm. In this case the runtime improves from 7.45 to 2.56 seconds.

3.2. Evaluation of different attributes

In a first approach to the algorithm we evaluated it with the length, mean radius and angle attributes of the branches, as well as the radius attribute of the nodes. We found out that the only robust attribute was the length. Both mean radius of the branches and radius of the nodes were providing many wrong matches as result, since there where many nodes and branches with similar values. As for the angle attribute, even if it was not very robust, we got less wrong matches than with the other two attributes. Due to the bad results that we got from using radius and mean radius, we decided to use a combination between length and angle to deal with situations where two branches have identical length. After testing the algorithm more deeply we realized that this assumption was not good enough and started to use attributes that Graham et al. were advising in their work. We evaluated them and found that for our application the coordinate attribute was not good, and we decided to use a combination between length and direction. The best results were obtained with threshold values for length and direction of 1.5 and 0.6 respectively and giving to both of them the same weight (0.5).

Table 1: *Efficiency. Time in seconds that the algorithm requires for trees with different number of nodes and input matches. The No. nodes column shows the number of nodes of the first and the second trees.*

No. nodes	0 Input	1 Input	2 Input	3 Input
192-192	7.45	2.56	1.41	0.74
192-101	2.84	1	0.48	0.32
192-52	0.95	0.46	0.22	0.21
192-11	0.15	0.12	0.12	0.13
101-101	0.96	0.45	0.24	0.19
101-52	0.35	0.18	0.11	0.1
101-11	0.1	0.07	0.08	0.07
52-52	0.14	0.13	0.09	0.1
52-11	0.03	0.04	0.04	0.05

3.3. Topological experiments

We use two different methods to introduce topological deformations to the trees. For each of the method 60 experiments were realized (30 with 25% of the nodes removed and 30 with 50% of the nodes removed). The first method consists on the removal of random nodes. When a node is removed all the nodes and branches that derive from it are also removed. The second method removes also nodes randomly with the difference that the nodes that derive from it are not removed anymore. The nodes that are direct children of the removed node are connected to the parent of the mentioned node. This introduces big topological deformations. The more nodes are removed the bigger are the deformations and the more difficult is to match every node-pair correctly.

3.3.1. Results

The results are summarized in table 2 and table 3. The wrong matches are given as an average of wrong matches obtained

in the experiments. In addition to this the tables contain information about the number of obtained matches, the number of experiments that had wrong matches, and the maximum and minimum number of matches obtained during the 30 experiments. The number in parenthesis represents the maximum number of wrong matches that were obtained.

For the first method the results are very good. As can be seen there are no errors and all the possible node-pairs are matched (the size of the trees is 192 and 144 when 25% of the nodes are removed and 192 and 96 when 50% are removed)(Table 2). The second method shows that the algorithm is very robust towards topological deformations (Table 3). Even when 50% of the nodes are removed (the topological deformations are too big, unrealistic for medical applications) the average of wrong matches is 1.4 errors for the 30 performed experiments, and only a maximum of 3 wrong nodes are obtained. It can be seen that the number of matches is not as high as with the first method. As we wrote before, the deformations derived from this method are very big. It would be possible to get more matches by simply introducing more primitives to deal with ramifications of more than 4 children. However, until now we find this is not necessary in realistic applications that deal with liver vessel trees.

Table 2: *Topological experiments. Method 1*

Rem	Matches	Wrong	Max	Min	Wrong Exp
25%	144	0	144	144	0
50%	96	0	96	96	0

Table 3: *Topological experiments. Method 2*

Rem	Matches	Wrong	Max	Min	Wrong Exp
25%	103.82	1.17(2)	131	62	6
50%	46.59	1.4(3)	68	22	7

3.4. Attribute experiments

To evaluate the effect of having trees with slightly different attributes uniformly distributed noise is added to one of the trees. In our experiment the standard deviation σ of the noise is changed from 0.2 to 0.8. For each standard deviation 30 experiments were performed and the average of the results is calculated.

3.4.1. Results

Table 4 show the results of the experiments. The algorithm is very robust against noise addition to the attributes. In the worst case the average of wrongly matched nodes is 1, and not less than 184 nodes are matched from all the nodes (192). It is worth to mention that of the 30 experiments realized for $\sigma = 0.8$, there were only 2 experiments with wrongly matched nodes.

Table 4: *Addition of noise*

σ	Matches	Wrong	Max.	Min.	Wrong Exp.
0.2	192	0	192	192	0
0.4	191.47	0	192	189	0
0.6	189.83	1	192	184	1
0.8	183.13	1	192	174	2

Figure 9 is a summary of the presented results for the topological and the attribute experiments. As can be seen the worst results are obtained for the second topological method, where fewer nodes are matched. In all the cases the algorithm is very robust and presents almost no wrong matches.

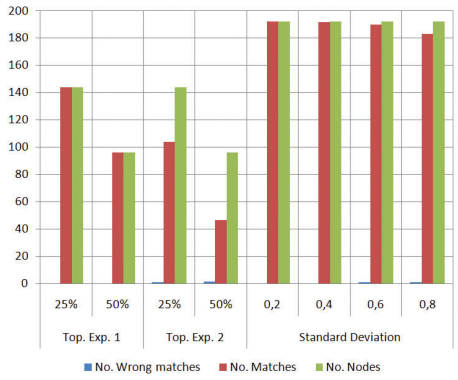

Figure 9: *Summary of the results of the experiments. Average number of wrong matches (blue), average number of matched nodes (red) and total number of nodes (green)*

3.5. Realistic experiments

It is difficult to determine the number of correct or wrong matches in the case of more realistic experiments. We checked every node-pair one by one to determine which of them were matched as expected and which not. It must be noticed that it is not always objective to determine the nodes that are wrongly matched as there are branches in one tree that are shorter than branches on the other one, but that belong together. In this case matches could occur that seem to be correct but are not. On the contrary, matches could be correct but appear as if they were not. Figure 10 shows the result of an experiment.

We applied the algorithm to two different datasets. We obtained both of them by cropping the complete CT volume, and getting therefore, a dataset that it is smaller and slightly different than the previous one. In the second case the algorithm employed to generate the graph was also different getting two datasets containing many different branches. After evaluating of the results, 0 to 3 matches were considered as wrong. In the first experiment 63 of all the nodes (78) where

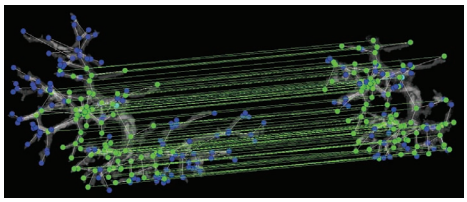

Figure 10: *Visualization of results of a realistic experiment.*

matched, and in the second 83 of all the nodes (129). As it can be seen in the realistic experiments the number of obtained matches is high. We come therefore to the conclusion, that the relatively small number of matches obtained with the topological deformations is not decisive in real applications. On the other hand the algorithm is still robust toward realistic experiments.

4. Discussion

The algorithm as it was presented is not pose independent. The similarity measure is formed by the length (pose independent) and the direction (pose dependent) attributes. Two possible solutions to this problem have been considered. The first one is to replace the direction attribute with another more appropriate one. In this direction, some preliminary experiments have been performed using the angle between branches. Even if the obtained results are good, they are not as good as the ones obtained with the direction attribute. They present wrong matches in the order of 2 or 3 more than in the presented algorithm. The combination with more pose independent attributes could improve those results. Another option would be to continue using the same attributes that make the algorithm very robust by previously applying a rigid registration. Far to be a disadvantage this would make the task of the doctor to interact with the trees easier, at the same time allowing the algorithm to use the direction attribute. From this discussion we expect that the presented algorithm can be made robust against pose changes.

5. Conclusion

The presented work provides an enhancement to Graham et al.'s algorithm. We proposed seven new primitives as an extension to Graham et al.'s algorithm when trifurcations are not enough to cover all the ramifications of the tree. Furthermore, it enables the preselection of important matches, which makes it more efficient. With 3 preselected matches our algorithm is always below 1 second for trees of up to 192 nodes. In addition to this the preselection of matches makes the algorithm more suitable for use in medical environment, as the doctor is able to guide the process. We provided also a detailed evaluation of the presented algorithm. Topological deformations as well as noise are introduced to the trees, proving it to be a very robust tree matching algorithm. It would be interesting as future work to let the doctor choose the root of the trees, as there will be intraoperative situations where the whole tree will not be available. This can be the case with ultrasound data where only a small part of the tree will be visible. More evaluation should be done in this direction.

Acknowledges

This work was partially supported by the Fraunhofer Internal Programs under Grant No. MAVO 817 775.

References

[CAM*05] CHARNOZ A., AGNUS V., MALANDAIN G., SOLER L., TAJINE M.: Tree matching applied to vascular system. *Graph-Based Representations in Pattern Recognition 3434* (March 2005), 183–192. 2

[DOLCE10] DRECHSLER K., OYARZUN LAURA C., CHEN Y., ERDT M.: Semi-automatic anatomical tree matching for landmark-based elastic registration of liver volumes. *Journal of Healthcare Engineering 1*, 1 (2010), 101–123. 2

[GH06a] GRAHAM M. W., HIGGINS W. E.: Globally optimal model-based matching of anatomical trees. In *Medical Imaging: Image Processing* (2006), vol. 6144. 2, 3

[GH06b] GRAHAM M. W., HIGGINS W. E.: Optimal graph-theoretic approach to 3d anatomical tree matching. *IEEE ISBI* (2006), 109–112. 2

[GR96] GOLD R., RANGARAJAN A.: A graduated assignment algorithm for graph matching. *IEEE TPAMI 18*, 4 (April 1996), 377–388. 1

[KKNN06] KAFTAN J. N., KIRALY A. P., NAIDICH D. P., NOVAK C. L.: A novel multi-purpose tree and path matching algorithm with application to airway trees. *Medical Imaging 2006: Physiology, Function, and Structure from Medical Images 6143* (March 2006), 215–224. 2

[MKC01] MEDASANI S., KRISHNAPURAM R., CHOI Y.: Graph matching by relaxation of fuzzy assignments. *IEEE Trans. on Fuzzy Syst. 9*, 1 (February 2001), 173–182. 1

[MKS*07] METZEN J. H., KRÖGER T., SCHENK A., ZIDOWITZ S., PEITGEN H.-O., JIANG X.: Matching of tree structures for registration of medical images. *Graph-Based Representations in Pattern Recognition 4538* (August 2007), 13–24. 1

[PSZ99] PELILLO M., SIDDIQI K., ZUCKER S. W.: Matching hierarchical structures using association graphs. *IEEE TPAMI 21*, 11 (November 1999), 1105–1119. 1

[TMP*05] TSCHIRREN J., MCLENNAN G., PALÁGYI K., HOFFMAN E. A., SONKA M.: Matching and anatomical labeling of human airway tree. *IEEE Trans. Med. Imaging 24*, 12 (December 2005), 1540–1547. 2