

Correspondences Search for Surface-Based Intra-Operative Registration

Thiago R. dos Santos*, Alexander Seitel,
Hans-Peter Meinzer, and Lena Maier-Hein

German Cancer Research Center, Div. Medical and Biological Informatics,
Heidelberg, Germany
`t.santos@dkfz.de`

Abstract. Intra-operative registration is one of the main challenges related to computer-assisted interventions. One common approach involves matching intra-operatively acquired surfaces (e.g. from a laser range scanner) to pre-operatively acquired planning data. In this paper, we propose a new method for correspondences search between surfaces, which can be used for the computation of an initial alignment. It generates graph representations and establishes correspondences by maximizing a global similarity measure. The method does not rely on landmarks or prominent surface characteristics and is independent on the initial pose of the surfaces relative to each other. According to an evaluation on a set of liver meshes, the method is able to correctly match small submeshes even in this presence of noise.

1 Introduction

Computer-assisted interventions generally require registration of pre-operative planning data with the patient’s anatomy. Usually this registration is performed through the use of landmarks. Another approach to achieve this is to acquire intra-operative surfaces (e.g. with a laser range scanner) and match them to surfaces extracted from the planning data. Some commercial systems integrate such surface acquisition technologies for marker-less registration (e.g. BrainLab VectorVision[®]). However, a complete view of the organ of interest is usually not available in intra-operative situations, making only the acquisition of a partial surface possible, which must be matched to a reference surface [1]. Furthermore, many physical factors, such as lightning and reflexivity, may influence the acquisition devices and generate noise. In general, the process of surface matching comprises three steps: *feature extraction*, *correspondence search* and *transformation computation* [2]. Regardless of the choice of a transformation class (rigid, affine, free-form deformation) and the method for computing it, the matching result depends crucially on the correspondences established in the second step.

Despite the various surface matching methods that have been presented so far (e.g. [3,4]), we found that most of them rely on the identification of points in

* T.R dos Santos is financed by the CAPES/DAAD (Brazil/Germany) scholarship program. The authors would like to thank Tobias Heimann for the helpful insights.

prominent regions and disregard the spatial relation between descriptors, thus neglecting contextual information that can be crucial for the correct and unambiguous match of small submeshes. In this paper, we present a graph matching-based method for finding correspondences between region features which does not rely on prominent characteristics or on an initial alignment of the meshes. This method is based on the determination of a maximal global similarity, instead of searching for common subgraphs, making it robust to noise.

2 Methods

Our method for correspondences search is performed according to the following procedure (Fig. 1): First, the input meshes are segmented (sec. 2.1). Next, a graph representation is generated from both meshes (sec. 2.2). A graph matching method based on the maximization of a global similarity measure is used for establishing correspondences (sec. 2.3). A method for the elimination of false assignments is employed in the last stage of the graph matching procedure. The established correspondences can then be used to rigidly align the two surfaces, using e.g. the centroids of the assigned region pairs as corresponding points.

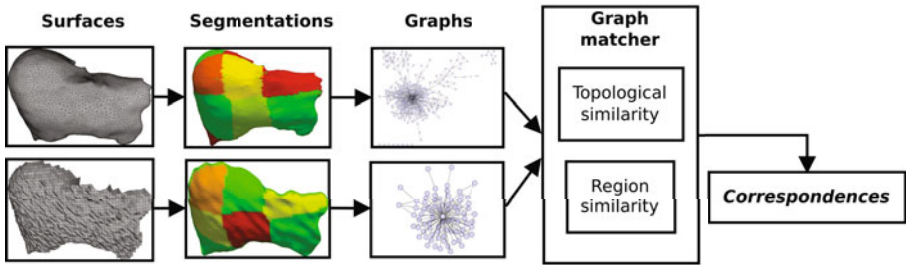


Fig. 1. The correspondence search pipeline. In order to match two surfaces, they are smoothed, segmented and graph representations are created. Those graphs are then matched in order to establish correspondences between the surfaces, by the maximization of the global node assignment similarity.

2.1 Mesh Segmentation

In this step, the input meshes are subdivided into regions that share similar descriptors. In theory, any method for mesh segmentation can be used here. In this paper, we adopted two curvature-based measures that describe the local shape of a particular mesh: the *shape index* and *curvedness*, which are both position and orientation independent [5]. Given two adjacent vertices, v_1 and v_2 , the distances in shape index (Δs) and curvedness (Δc) between v_1 and v_2 are computed according to the following equations:

$$\Delta s(v_1, v_2) = |s(v_1) - s(v_2)| \quad (1)$$

$$\Delta c(v_1, v_2) = \begin{cases} \frac{\max(c(v_1), c(v_2))}{\min(c(v_1), c(v_2))} - 1 & \min(c(v_1), c(v_2)) > 0 \\ 0 & c(v_1) = 0 \wedge c(v_2) = 0 \\ \infty & \textit{otherwise} \end{cases} \quad (2)$$

Because of the fact that the curvedness scales inversely with size [5] and is thus dependent on the actual size of the object, computing its distance as a quotient leads to a scaling invariant measure. In the case where one of the vertices is associated with a planar surface region ($c(v) = 0$), there is either no variation, when both vertices are planar, or it is set to ∞ (maximal distance).

Starting at any vertex v , we use local distances of shape index and curvedness between it and its neighbors to segment the mesh. Vertices with descriptor distances smaller than given threshold values (t_s for shape index and t_c for curvedness) are assigned to the same region. Each region R is then represented by a 2-tuple $\bar{R} = (\bar{s}_R, \bar{c}_R)$, which is composed of the means of the shape indices and the curvedness values of all points contained in it.

2.2 Graph Construction

Once a surface S has been partitioned into regions, an attributed graph $G_S = (V_S, E_S)$ can be constructed, where V_S is a set of nodes and E_S is a set of arcs. Every $n \in V_S$ represents a surface region R and is attributed with the respective representative \bar{R} . Every arc $e \in E_S$ represents a neighborhood relation between two regions R_i and R_j on S . Each arc e is oriented from the region with lower curvedness to the region with higher curvedness.

2.3 Graph Matching

Given two graphs $G_A = (V_A, E_A)$ and $G_B = (V_B, E_B)$, a similarity matrix between their nodes is computed. This matrix is then used to compute an assignment between the nodes, such that the global similarity (the sum of the similarity of all assigned node pairs) is maximal.

Let us define the matrix $S_V = [p_{n_B n_A}]_{|V_B| \times |V_A|}$, which holds the similarity scores between the nodes of G_A and G_B . S_V is composed of two parts

$$S_V \leftarrow S_T + S_R \quad (3)$$

where, $S_T = [q_{n_B n_A}]_{|V_B| \times |V_A|}$ represents the topological similarity scores, and $S_R = [r_{n_B n_A}]_{|V_B| \times |V_A|}$ represents the region similarity scores. Once, the similarities have been computed, S_V is used to determine an optimal global assignment.

Topological similarity scoring. The topological similarity scores are computed locally for each pair of nodes in $(V_B \times V_A)$ through a neighborhood scoring method [6]. According to this method, two nodes are considered similar if their topological neighborhoods are similar. Starting with initial node scores, the similarity between two particular nodes is computed as the sum of the similarity

between their neighbors. This procedure is repeated until a convergence limit is reached. In this way, scores between nodes propagate along to neighboring nodes at each iteration step.

Let us assume two nodes $n_A \in V_A$ and $n_B \in V_B$, and the functions $s(e)$ that returns the source node of a particular arc e , and $t(e)$ that returns the target node. The scores are computed iteratively as follows:

$$q_{n_B n_A}^k \leftarrow \sum_{s(e_B)=n_B, s(e_A)=n_A} (q_{s(e_B)s(e_A)}^{k-1} + q_{t(e_B)t(e_A)}^{k-1}) + \sum_{t(e_B)=n_B, t(e_A)=n_A} (q_{s(e_B)s(e_A)}^{k-1} + q_{t(e_B)t(e_A)}^{k-1}) \quad (4)$$

until $|q_{n_B n_A}^k - q_{n_B n_A}^{k-1}| \leq \varepsilon$ for every n_A and n_B , where ε denotes the convergence limit.

Instead of setting $q_{ij}^0 = 1$, for every i and j , as proposed by Zager and Verghese [6] in the case where no previous information about the nodes is available, we use the region similarity scores to initialize it, with cut off thresholds of ∞ . Zager and Verghese [6] have proven equation 4 converges independent of the initial values chosen.

Region similarity scoring. Topological similarity by itself would be insufficient for finding corresponding node pairs because many nodes are topologically identical. To address this issue, we also compute the similarity between the regions using their representatives \bar{R} (sec. 2.1). The similarity scores between them are evaluated through a Gaussian kernel, which assigns higher scores to smaller descriptor distances, while giving smaller scores to higher distances. Assuming two region representatives \bar{R}_1 and \bar{R}_2 , we denote their shape index and curvedness distances as $\bar{\Delta}s(\bar{R}_1, \bar{R}_2)$ and $\bar{\Delta}c(\bar{R}_1, \bar{R}_2)$ respectively (sec. 2.1). The Gaussian kernel is defined as follows:

$$g(\Delta, \sigma, \tau) = \begin{cases} -\infty & d > \tau \\ e^{-\frac{\Delta^2}{\sigma^2}} & d \leq \tau \end{cases} \quad (5)$$

where Δ is the distance, σ is the kernel width and τ is the threshold. Having $n_A \in V_A$ and $n_B \in V_B$, the region similarity matrix is computed as follow:

$$r_{n_B n_A} \leftarrow g(\bar{\Delta}s(\bar{R}_A, \bar{R}_B), \sigma_s, \tau_s) + g(\bar{\Delta}c(\bar{R}_A, \bar{R}_B), \sigma_c, \tau_c) \quad (6)$$

for every n_A and n_B . In the equation σ_s and τ_s denote the shape index kernel width and threshold parameters, and σ_c and τ_c denote the curvedness ones.

Assignment computation. In this stage, an assignment between the nodes of both graphs is computed, such that the global similarity is maximal. This problem is known as assignment problem and there are several methods to solve it [7]. Those methods take a cost matrix and create assignments between each row element to a column element, such that the sum of the costs of the assigned elements is maximized. We use the node similarity scores matrix S_V as a cost matrix and apply the popular *Munkres'* algorithm [8].

Elimination of false assignments. In order to reduce false matches among the assigned nodes, we adopt the premise that, if a particular node $n_A \in V_A$ was assigned to a node $n_B \in V_B$, there must be some other nodes in the neighborhood of n_A that were assigned to nodes in the neighborhood of n_B . If this premise is not confirmed, then the assignment between these nodes is probably incorrect and is removed.

Let N_n^h denote the topological neighborhood with radius h of a particular node n , and $C(N_{n_A}^{h_A}, N_{n_B}^{h_B})$ be a function that returns the number of nodes in the set $N_{n_A}^{h_A}$ that are assigned to nodes in $N_{n_B}^{h_B}$. If n_A was assigned to n_B in the assignment computation stage, this assignment is eliminated if the following equation does not hold:

$$C(N_{n_A}^{h_A}, N_{n_B}^{h_B}) \geq \beta \quad (7)$$

where β denotes the minimal number of node assignments that must exist between $N_{n_A}^{h_A}$ and $N_{n_B}^{h_B}$ in order to maintain the assignment between n_A and n_B .

3 Evaluation

We performed three evaluation studies on five liver meshes, to assess the robustness of the proposed method with respect to the submesh size (sec. 3.1), the influence of the shape descriptor (sec. 3.2), and the influence of noise (sec. 3.3). In all studies, an initial alignment between the shapes was obtained through a transformation that matched the centroids of the assigned regions in a least squares sense.

3.1 Evaluation of Robustness with Respect to the Submesh Size

The following experiment was performed for each reference surface: For each integer i in $[1, 50]$, 500 random submeshes (samples) with an area making up $i\%$ of the area of the reference mesh were generated with a region growing method. Each of the samples was translated and rotated with a random rigid transformation, and the proposed correspondence search algorithm was applied to initially realign the submesh with the reference mesh. Subsequently, ICP was performed to adjust the positioning.

To assess the accuracy of the alignment, the percentage of correctly classified regions and the percentage of incorrectly classified regions was determined. Furthermore, the quality of the final match (after ICP) was evaluated by calculating all distances between the transformed submesh vertices and their corresponding vertices in the reference mesh. The match was considered correct, if all distances were smaller than 10^{-6} (note that all meshes were scaled to fit into the unit box). The parameters were chosen empirically on a set of different meshes as follows: $t_s = 0.3$; $\sigma_s = \sigma_t = 0.1$; $\tau_s = \tau_c = 0.5$; $h_A = h_B = 2$; $\beta = 8$. The curvedness threshold (t_c) was set according to the submesh size to be matched: The smaller the submesh size, the smaller the threshold. This way, a reduction of the size of the

graphs and thus to a reduction of processing time could be achieved for relatively big submeshes. The curvedness threshold was linearly increased from 0.1 to 2.

3.2 Curvature Classes vs. Continuous Curvature Measures

To evaluate the influence of the the shape descriptor on the presented correspondence search method, we repeated the first experiment with an alternative mesh segmentation method. As originally proposed by Besl *et al.* [9], the surfaces were partitioned based on curvature *classes*, which are determined through the sign of the Gaussian and mean curvatures.

3.3 Evaluation on Noisy Data

In the case of noisy data, the parameters of the segmentation algorithm must be chosen carefully and in an application-specific manner. Two livers were used to empirically optimize the mesh segmentation parameters for each submesh size $i \in \{10\%, 20\%, 30\%, 40\%, 50\%\}$. The smaller the submesh gets, the smaller the tolerance in the mesh segmentation, yielding bigger graphs. Bigger graphs increase the number of nodes that are common to both meshes and help identifying correct correspondences.

The parameters were chosen empirically on a set of different meshes as follows: $\sigma_s = 0.05$; $\sigma_t = 0.15$; $t_s = (0.15, 0.15, 0.12, 0.12, 0.1)$ and $t_c = (0.15, 0.13, 0.12, 0.1, 0.03)$ for submeshes of size (50%, 40%, 30%, 20%, 10%) respectively; $h_A = h_B = 2$; $\beta = 2$. For each submesh size, a set of 50 random submeshes were extracted, and a random vector with direction and magnitude drawn randomly from the interval $[0, 8]$ mm was added to each vertex to simulate noise in the data. The submeshes were then smoothed through the method proposed in [10]. Unlike in the previous experiments, the segmentation of the submesh was potentially different from the segmentation of the corresponding area in the reference mesh due to the noise in the data. Hence, there was generally no isomorphism in the graphs, making an evaluation of the correctly assigned nodes impossible. However, vertex correspondences were known. We assessed the quality of the alignment by computing the percentage of submesh vertices that were assigned to their corresponding ones in the reference mesh after the final iteration of ICP.

4 Results

The results of our evaluation are shown in Figures 2 and 3. When there was partial isomorphisms between the graphs (Fig. 2a), i.e. there was no noise added to the data, the percentage of correct and incorrect node assignments averaged over all five livers ranged from $83.5\% \pm 6.1\%$ and $0.3\% \pm 0.3\%$ respectively (submesh size: 1%) to $97.9\% \pm 3.1\%$ and $0.1\% \pm 0.1\%$ respectively (submesh size: 50%). The correct transformation was found in almost all cases ($94.9\% \pm 11.9\%$ for submesh size: 1%, $100.0\% \pm 0.0\%$ for submesh size: 50%). Depending on the submesh size, processing times for the correspondence search including mesh

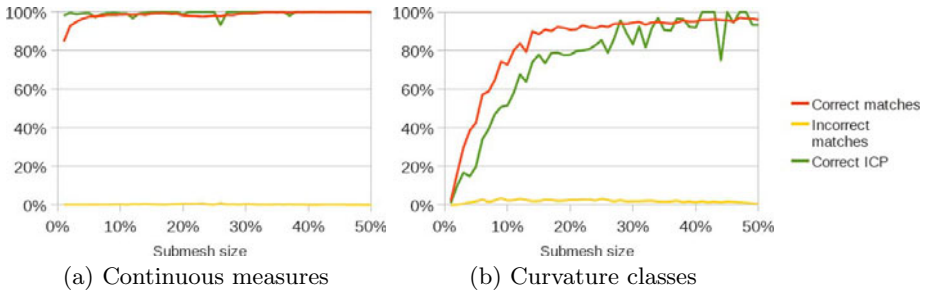


Fig. 2. Results of the surface matching experiment described in sections 3.1 (a) and 3.2 (b). For each reference mesh, the mean percentage of correct matches and incorrect matches after the initial alignment averaged over 500 samples is shown as a function of the submesh size. *Correct ICP* represents the quality of the final match (after ICP) as described in section 3.1. The values are averaged over the five reference surfaces.

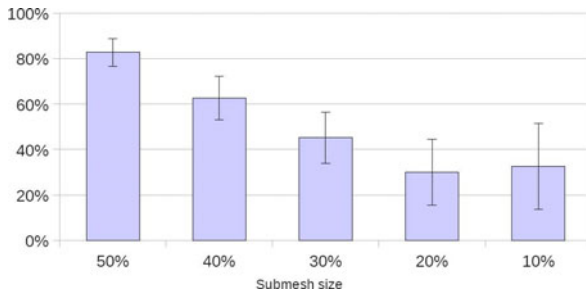


Fig. 3. Results of the experiment described in section 3.3, where the proposed correspondence search method was evaluated with noisy data. The mean percentage (averaged over 50 samples) of submesh vertices that were assigned to their corresponding vertices in the reference mesh, after the final iteration of the ICP, is shown as a function of the size of the submesh. The values are averaged over the five reference surfaces.

segmentation and graph generation ranged from 1 ms to 1.64 s. All processing times were measured using a non-threaded 2.4 GHz Intel machine.

The segmentation based on curvature classes yielded significantly worse results than the segmentation based on shape index and curvedness (Fig. 2b). In the former case, we obtained a percentage of correct and incorrect node assignments ranging from $1.0\% \pm 3.7\%$ and $0.1\% \pm 0.2\%$ respectively (submesh size: 1%) to $94.9\% \pm 1.9\%$ and $0.3\% \pm 0.3\%$ respectively (submesh size: 50%). For small submeshes ($\leq 10\%$) the ICP did not converge into the global optimum. Processing times for correspondence search ranged from 1 to 10 ms.

For noisy data (Fig. 3), the mean percentage of submesh vertices that were assigned to their corresponding vertices in the reference mesh, after the final iteration of the ICP, ranged from $32.5\% \pm 19.3\%$ (submesh size: 10%) to $82.9\% \pm 6.1\%$ (submesh size: 50%). Processing times ranged from 29 to 100 seconds.

5 Discussion

Although surface matching has been subject to considerable research efforts in the past decades, *partial* surface matching with small submeshes remains challenging. This holds especially when no prominent mesh features are available and in the presence of noise. According to our evaluation, the method presented in this paper is highly accurate, yielding a perfect match in almost all cases for submeshes without noise and that made up at least 5% of the size of the reference meshes used in the study. In the presence of noise, good matches were still obtained for submeshes larger than 30% of the mesh size.

We are aware of the fact that we used a relatively simple method for mesh segmentation by combining a classical region growing approach with a surface descriptor that is rather unstable in the presence of noise. However, the good performance of our algorithm despite these conditions demonstrates the potential of the graph-based registration approach. On the other hand, the results obtained with curvature classes indicate that the performance depends considerably on the descriptor and segmentation technique chosen.

Although our approach leaves room for future improvements, we conclude, that (1) our correspondence search algorithm proved to be accurate and robust even for very small submeshes given that these meshes represent exact instances of some reference mesh part and (2) that the results on noisy data are promising and suggest further development of the approach for intra-operative registration purposes.

References

1. Cash, D.M., Miga, M.I.: Compensating for intraoperative soft-tissue deformations using incomplete surface data and finite elements. *IEEE Trans. Med. Imag.* 24(11), 1479–1491 (2005)
2. Audette, M.A., Ferrie, F.P., Peters, T.M.: An algorithmic overview of surface registration techniques for medical imaging. *Med. Img. Anal.* 4, 201–217 (1999)
3. Gelfand, N., Mitra, N.J., Guibas, L.J., Pottmann, H.: Robust global registration. In: *Proc. EG Symp. Geo. Proc.*, pp. 197–206 (2005)
4. Gal, R., Cohen-Or, D.: Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.* 25(1), 130–150 (2006)
5. Koenderink, J.J., van Doorn, A.J.: Surface shape and curvature scales. *Img. & Vis. Comp.* 10(8), 557–565 (1992)
6. Zager, L.A., Verghese, G.C.: Graph similarity scoring and matching. *App. Math. Lett.* 21, 86–94 (2008)
7. Dell’Amico, M., Paolo, T.: Algorithms and codes for dense assignment problems: the state of the art. *Disc. App. Math.* 100, 14–48 (2000)
8. Bourgeois, F., Lassalle, J.C.: An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Comm. ACM* 14(12), 802–804 (1971)
9. Besl, P.J., Jain, R.C.: Invariant surface characteristics for 3d object recognition in range images. *Comp. Vis. Graph. & Img. Proc.* 33(1), 33–80 (1986)
10. Jones, T.R., Durand, F., Desbrun, M.: Non-iterative, feature-preserving mesh smoothing. In: *SIGGRAPH*, pp. 943–949. ACM Press, New York (2003)