Interactive and Simultaneous 3D Segmentation of Pleural Thickenings at Different Points in Time Using Graph Cuts

Peter Faltin1, Phan-Anh Nguyen1, Kraisorn Chaisawong1,2, Thomas Kraus3, Dorit Merhof1

1Institute of Imaging & Computer Vision, RWTH Aachen University
2King Mongkut’s University of Technology North Bangkok, Thailand
3Institute and Out-Patient Clinic of Occupational Medicine, Uniklinik RWTH Aachen
peter.faltin@lfb.rwth-aachen.de

Abstract. A precise correction of segmentations within an acceptable amount of interaction time is a demanding task. In this paper we present a method offering an intuitive way to correct the segmentation surface by user interaction, while including image information at the same time. An efficient segmentation of follow-up images is achieved by an automatic transfer of the segmentation results to a second point in time. Additionally, this leads to a more consistent follow-up assessment, since independent segmentations for both images might otherwise independently suffer from small deformations and image noise. We apply this method on pleural thickenings in follow-up CT scans. The thickenings are located at the pleura and typically have a relatively low resolution. The objective is a consistent segmentation and growth rate estimation of thickenings for the early diagnosis of malignant pleural mesothelioma.

1 Introduction

Pleural mesothelioma is a malignant form of cancer and is typically related to the previous exposure to asbestos fibers. Inhaled fibers can cause pleural thickenings. The thickenings are observable in thoracic CT data and act as an indicator for pleural mesothelioma. Their morphology can be complex. For a precise quantitative assessment automatic segmentation procedures [1] might need manual corrections by medical experts. With standard 2D segmentation methods this is a time consuming task and subject to strong inter- and intra-reader variability. To reduce the workload and to introduce more consistency for follow-up assessment, methods for comfortable and simultaneous segmentation of images at two points in time are desirable.

Existing interaction methods, to modify 3D surfaces, can be adapted for the segmentation correction of 3D volume data e.g. Proksch et al. [2]. Their methods allow a visual feedback from the image data during the correction process. The Live Wire tool [3] directly includes image information and allows the user to drag incorrectly positioned parts of the segmentation surface. The surface automatically snaps to edges, according to the displayed image slice. The range of
Fig. 1. Comparison between different interaction approaches to manipulate segmentation. Images show healthy lung contour (dotted, white) and initial segmentation (dotted, orange). (a) Typical approach: using a brush-like tool (gray). (b) Suggested approach: drag surface, which snaps in ROI (gray circle) to image data (dotted, green).

influence for the interaction is limited by a spherical region of interest (ROI), placed around the mouse cursor. We suggest a new method, which adopts the user interaction from the Live Wire tool, but operates in the voxel space. It is based on Graph Cuts and does not need the construction and maintenance of a triangular mesh. Furthermore, image information from the whole image, i.e. not only the information from the displayed slice, is considered. The proposed tool can also be used for the simultaneous segmentation at two points in time.

Fig. 1 visually compares our interaction approach to the well-known Graph Cuts based approach from Boykov et al. [4]. While dragging the surface, our approach offers a live preview, and allows a precise positioning before dropping the surface.

2 Materials and Methods

The presented method uses Graph Cuts to separate thickenings from the surrounding. For this purpose, a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with a set of nodes $\mathcal{V}$ and a set of connecting edges $\mathcal{E}$ is constructed. Each node $v \in \mathcal{V}_v \subset \mathcal{V}$ corresponds to an image voxel at position $p(v)$, with intensity value $I(p(v))$. Two special nodes, $\{s, t\} = \mathcal{V}_t = \mathcal{V} \setminus \mathcal{V}_v$, are introduced to represent the object and the background labels respectively. All nodes $v \in \mathcal{V}_v$ have incoming edges $(s, v)^T$ from the node $s$ and outgoing edges $(v, t)^T$ to the node $t$, which form the set $\mathcal{E}_t \subset \mathcal{E}$ of so called t-links. Additionally, the set $\mathcal{E}_n \subset \mathcal{E}$ contains all edges $(q, r)^T$ and $(r, q)^T$ connecting a 26-neighborhood $r \in \mathcal{N}(q)$ of each voxel $q$. These links are called n-links. All edges $e = (q, r) \in \mathcal{E}$ have a weight $w(e)$. The graph $\mathcal{G}$ is partitioned into two disjoint subsets $\mathcal{S}$ and $\mathcal{T}$ by a cut $\mathcal{C} \subset \mathcal{E}$, where $s \in \mathcal{S}$ and $t \in \mathcal{T}$. Each edge $e \in \mathcal{C}$ contributes its weight to the total cut cost. The cut with minimal costs, corresponding to the final segmentation, is found by applying the Min-Cut/Max-Flow algorithm suggested by Boykov et al. [4].

In this publication, only the front part of the thickening, facing the lung tissue, is segmented. For the backside, which could not be easily identified, a healthy lung model [1] is applied.
2.1 User interaction

The user interaction of dragging the surface is transferred into edge weights, by adding so-called hard constraints. These constraints are stored in an image $H$ of same size as $I$. Each voxel $v \in V$ has a corresponding hard constraint $H(p(v)) \in \{S, 0, T\}$. If the user drags the surface, in the first step, all hard constraints $H(p(v))$ of voxels $v \in V_{ROI}$ inside the ROI are set to 0. In the next step, all hard constraints $H(p(v))$ of voxels $v \in V_{v, line}$ on a line between the dragging start position $p_{start}$ and the current mouse position $p_{current}$ are set to $S$ or $T$. This choice depends on if the users extends or reduces the segmentation and is determined by the surface normal $n$ of the segmentation and the dragging direction, therefore

$$H(p(v)) = \begin{cases} S & \text{if } n^T \cdot (p_{current} - p_{start}) > 0, \forall v \in V_{v, line} \\ T & \text{else} \end{cases}$$

2.2 Edge weights

One of the most important criterions to separate the different image parts is their intensity. Intensity distributions for the two classes are used to calculate regional penalties based on the negative log-likelihoods [4]

$$R_s(v) = -\ln P(I(p(v))|v \in S),$$

$$R_t(v) = -\ln P(I(p(v))|v \in T),$$

which are added to the edge weights. The conditional probabilities are extracted from training data. The t-link weights are given by

$$w_k((s, v)^T) = \begin{cases} 0 & H(p(v)) \equiv T \\ K & H(p(v)) \equiv S \\ \lambda R_s(v) & \text{else} \end{cases},$$

$$w_k((v, t)^T) = \begin{cases} 0 & H(p(v)) \equiv T \\ K & H(p(v)) \equiv S \\ \lambda R_t(v) & \text{else} \end{cases},$$

where $\lambda$ is a parameter to weight the influence of the intensity model and $K$ is the maximum neighborhood weight, as defined in [4].

Another important criterion is the continuity of the segmenting surface, which is considered by penalizing the surface area of the segmentation boundary. A neighborhood system, as shown in Fig. 2(a), is utilized to calculate the weight terms $w_k(v), v \in V$ for all families $k$ of neighborhood relations. The boundary terms are calculated in Riemannian metric $D$ as suggested by Boykov et al. [4]

$$w_k^R(v) = w_k^S \frac{\det D(v)}{(u_k^T \cdot D(v) \cdot u_k)^2},$$

where $u_k$ is a vector of unit length, pointing in the direction of the neighbor voxel, and $w_k^S(p)$ is the corresponding boundary term in Euclidean metric. The anisotropic Riemannian metric depends on the image $I$,

$$D(v) = g(|\nabla I(p(v))|) \cdot J_v + (1 - g(|\nabla I(p(v))|)) \cdot u \cdot u^T,$$
(a) 26-neighborhood of the gray voxel. Each neighbor is labeled with number \( l \) of edge family.

(b) Corresponding closest distance within edge family.

Fig. 2. Relation between edge family and corresponding closest distance of edges.

where \( I_3 \) is the identity matrix and \( g(x) = \exp(-\frac{x^2}{2\sigma^2}) \). The Euclidean metric is

\[
\frac{w_{\kappa}}{\pi} = \frac{\Delta \rho_k \cdot \Delta \Phi_k}{\pi},
\]

(7)

where \( \Delta \rho_k \) is the closest distance between the lines within the family \( k \), and \( \Delta \Phi_k \) is the angular differences between the nearest families of edge lines. Using a dense neighborhood this can be approximated using a spherical Voronoi diagram [5] and we chose \( \Delta \Phi_k = \frac{4\pi}{26} \). Typically, for CT data the voxel spacing \( \delta_{x,y} \) in \( x \)- and \( y \)-direction is identical and the spacing in \( z \)-direction is \( \delta_z > \delta_{x,y} \). For this special case the distances \( \Delta \rho_k \) are reduced to 5 cases and given in Fig. 2(b). The n-link weights \( w(e) \) are determined by the edge family \( k \), with \( \kappa : e \mapsto k \),

\[
w(e) = w_{\kappa(e)}, \forall e \in \mathcal{E}_n.
\]

(8)

2.3 Simultaneous segmentation

For the transfer of the segmentation between different points in time, we assume that a non-rigid and volume preserving registration is available e.g. by applying the method suggested by Rohlfing et al. [6]. The labeling is transferred to the other point in time, and the signed distance function \( \Phi(p(v)) \), which is the closest distance for each point \( p(v) \) to the segmentation boundary, is extracted. The distance has a negative sign in the interior of the segmentation and a positive sign outside. Thickenings do not shrink over time but may grow. The volume-preservation is required during registration to effectively translate this knowledge into constraints: Shrinking is strongly penalized, while growing is only penalized if it is in an unreasonable range. In combination with the signed distance function
\( \Phi(p(v)) \), we chose the t-link weights at the second point in time

\[
\begin{align*}
w'(s, v) &= \lambda \cdot R_s(v) + \begin{cases} 
\Phi(p(v)) & \text{if } \Phi(p(v)) < 0 \\
0 & \text{else}
\end{cases}, \\
\lambda &= 0.006.
\end{align*}
\]

(9)

where \( \alpha, \beta, \gamma \) are parameters to influence the penalization. \( R_s \) and \( R_t \) are the intensity based penalties for the second point in time. The chosen penalties for the transfer in Eq. 9 and Eq. 10 are shown in Fig. 3(a).

In the graphical user interface both points in time are shown side-by-side and the user can see the live results of the interaction in both images.

3 Results

Our newly developed segmentation tool was tested in an initial evaluation by a group of 11 users and was compared to the Live Wire tool \([3]\), which provides a similar user interface. Both tools were implemented using MITK \([7]\) and tested in an identical scenario, where the users had 120 seconds to segment a given thickening. The tool presented in this paper was utilized for a simultaneous segmentation. Those results for the second point in time, where user interaction was only indirectly applied, were compared to results from the Live Wire tool, directly applied on the second point in time. The weight influence for the intensity model was chosen as \( \lambda = 0.006 \). The resulting segmentations were compared with a reference segmentation of the thickening. This reference was created by a voxel-wise majority decision of segmentations carefully carried out voxel-by-voxel by 3 experienced users. For performance analysis, the Tanimoto coefficient

\[
(10)
\]

was calculated for each user's result.

Fig. 3. Mean segmentation quality and applied penalties for segmentation transfer.

(a) Penalties to keep segmentation consistent, with \( \alpha = 1, \beta = 5 \) and \( \gamma = 3 \).

(b) Mean results off all users over time. Error bars show variance with factor of 10.
coefficient \( \frac{TP}{TP + FP + FN} \) was chosen as a quality criterion. User segmentation and reference segmentation were compared using the healthy lung model [1] as the backside. Resulting quality and variance for different users are plotted over time in Fig. 3(b). Both curves show similar improvement of quality over time, which converges to a Tanimoto coefficient of approximately 0.85. Different starting points were caused by the initial segmentation results of the Graph Cuts method without any user interaction. Both tools resulted in similar inter-user variances, visualized by error bars. Our proposed tool performed slightly better in case of variance and quality compared to the Live Wire tool; especially a faster convergence is achieved. Additionally, the users rated both tools according to their intuition, reaction time, precision, efficiency and usability. Our new tool was, averaged for all users, superior in all criteria except for the reaction time.

4 Discussion

The presented segmentation tool offers a convenient way to correct 3D segmentations, by combining image information and user interaction. Corrections from one point in time are successfully transferred to the other point in time. Initial tests indicate that it is superior to the similar Live Wire tool [3] in most criteria. The results achieved as a consequence of indirect user interaction at another point in time are even better than the results from the Live Wire tool directly applied on this image. A current limitation is the reaction time for a simultaneous segmentation, which is not comparable to the Live Wire tool applied on a single independent image. This will be addressed in future work by interleaving computation and interaction as well as GPU computing. One of the main tasks for the future will be to create a more comprehensive evaluation scenario to judge the tool’s suitability for general segmentation correction.

References