Eurographics Workshop on Visual Computing for Biology and Medicine (2012)
T. Ropinski, A. Ynnerman, C. Botha, and J. B. T. M. Roerdink (Editors)

Sketch-based Image-independent Editing of 3D Tumor Segmentations using Variational Interpolation

F. Heckel[1], S. Braunewell[1], G. Soza[2], C. Tietjen[2] and H. K. Hahn[1]

[1]Fraunhofer MEVIS, Germany
[2]Siemens AG, Healthcare Sector, Imaging & Therapy Division, Computed Tomography, Germany

Abstract

In the past years sophisticated automatic segmentation algorithms for various medical image segmentation problems have been developed. However, there are always cases where automatic algorithms fail to provide an acceptable segmentation. In these cases the user needs efficient segmentation correction tools, a problem which has not received much attention in research. Cases to be manually corrected are often particularly difficult and the image does often not provide enough information for segmentation, so we present an image-independent method for intuitive sketch-based editing of 3D tumor segmentations. It is based on an object reconstruction using variational interpolation and can be used in any 3D modality, such as CT or MRI. We also discuss sketch-based editing in 2D as well as a hole-correction approach for variational interpolation. Our manual correction algorithm has been evaluated on 89 segmentations of tumors in CT by 2 technical experts with 6+ years of experience in tumor segmentation and assessment. The experts rated the quality of our correction tool as acceptable or better in 92.1% of the cases. They needed a median number of 4 correction steps with one step taking 0.4s on average.

Categories and Subject Descriptors (according to ACM CCS): I.4.6 [Image Processing and Computer Vision]: Segmentation—I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

1. Introduction

Segmentation is one of the essential tasks in medical image analysis. It typically is the basis for all successive steps, such as classification or quantification, or even has direct implications for the patient, like in radiotherapy planning. Therefore, a proper segmentation of objects in medical images is crucial in many cases. Since a fully manual segmentation takes too much time in clinical routine and lacks reproducibility, automatic algorithms have been developed as an alternative for specific purposes. However, even sophisticated algorithms fail to provide an acceptable segmentation in some cases. For example, due to low contrast, noise and biological variability, the segmentation might leak into adjacent structures or parts of the objects are missing in the segmentation result. Nevertheless, most parts of the segmentation mask are usually correct and only small regions are erroneous. This makes segmentation editing an indispensable step in the segmentation process, where the user has the possibility to manually modify the segmentation until it satisfies his or her needs.

Manual correction in 3D is a challenging task, where as many segmentation problems as possible should be solved with as few 2D interactions as possible. Hence, the tool needs to estimate the user's intention in 3D while it should modify the segmentation only locally and the algorithm needs to be fast enough to react on inputs in real time. The segmentation problems that should be solved by such a tool are typically difficult, because the preceding algorithm already failed to segment the object appropriately. However, research typically focuses on (semi-)automatic segmentation methods, while manual correction does not receive much attention.

In this paper, we discuss sketch-based segmentation editing in 2D and we present a novel method for intuitive editing of 3D segmentations of compact objects in the context of tumor segmentation in CT. The proposed method is based on

DOI: 10.2312/VCBM/VCBM12/073-080

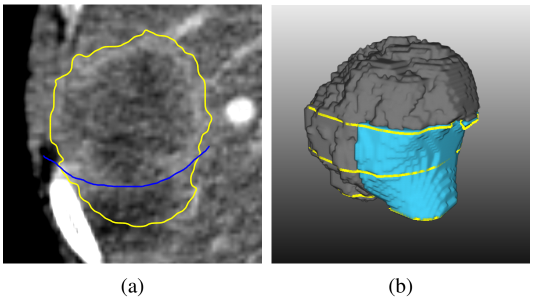


Figure 1: Sketch-based segmentation editing: (a) initial segmentation (yellow), manual correction (blue) and (b) 3D result after editing with our variational-interpolation-based approach.

an object reconstruction using variational interpolation (see Fig. 1). It is image-independent so it does not rely on image information. If the contrast or the signal-to-noise ratio are very low, the image often does not provide enough information for segmentation, so image-independent editing is better suited in such cases. Our goal was developing a tool that allows correcting a segmentation with a few steps in a few seconds for the majority of problem cases. It should only use a segmentation mask and the user input for its computations, so it can be combined with any (semi-)automatic segmentation algorithm and it can be used in any 3D modality, such as CT or MRI.

User interaction is performed by sketching, i.e., in a contour-based manner, which has previously proven to be an intuitive interaction and has been widely accepted by various clinicians in our evaluations [HMB*09]. We have designed our algorithm as a general segmentation tool that also enables the user to specify segmentations of objects from scratch, in cases where a dedicated (semi-)automatic segmentation algorithm has failed completely or is not available at all. The proposed method works for objects given by parallel slices and it also allows a slice-wise manual correction for cases where the extrapolation in 3D does not conform to the user's intention. Our tool can be used in any view and it even allows to user to arbitrarily switch between views during the editing process.

2. Related Work

Up to now only a few authors have focused on fast and easy to use tools for manual correction of 3D medical image segmentations. However, several authors have mentioned the necessity of such tools and propose basic solutions for their frameworks where manual correction is often referred to as manual refinement [HAB95, PFJ*03, YPH*06, BDK*07, HvGS*09, HM09]. As stated by Olabarriaga and Smeulders, manual correction needs high-level tools for being efficient [OS01].

Some basic low-level tools for 3D manual correction are suggested by Kang et al.: hole-filling, point-bridging, surface-dragging [KEK04]. Maleike et al. also implemented a set of general low-level 2D correction tools in their interactive segmentation framework (add, subtract, correction (meaning automatic add/subtract), interactive region growing, fill, erase) [MNMW09]. Their contour-based 2D correction tool is based on work by Heimann et al. [HTKM04]. A manual leak prevention approach for region-growing-based segmentation algorithms has been mentioned by Kunert et al. [KSH*04]

An image-based high-level approach using the random walker algorithm has been suggested by Grady and Funka-Lea [GFL06]. Maleike et al. have developed a manual correction tool using their automatic surface-based segmentation algorithm for lymph-nodes, which uses a statistical shape model that is adapted to gradients within the image [MFT*08]. For their dedicated graph-cut-based segmentation algorithm, Egger et al. propose a manual refinement approach where the user has the possibility to manually delete and add seed points [ECFN11]. Based on the image foresting transform a general manual correction algorithm is proposed by Miranda et al. applied to brain segmentations [MaRC11]. In a previous work we have proposed a contour-based manual correction algorithm for tumor segmentations based on edges within the image, which uses a live-wire-like approach [HMB*09]. However, image-based methods can fail for low-contrast or noisy images.

Image-independent manual correction is often done on the basis of deformable 3D meshes [JSG03, TPvB*03, BBS06, BBB*07,SHT*08,SSMS10]. Interaction techniques for such surface-based tools have been discussed by Proksch et al. and Rahner et al. [PDP10, RDDP10]. An approach by Silva et al. also allows adding and removing voxels in 3D by a spherical brush with a user defined radius [SSMS10]. However, surface models and drawing tools are not intuitive in 3D. Moreover, surface based correction methods can result in unexpected modifications, as shown by Silva et al. [SSMS10].

Manual correction for 3D medical image segmentation could also be inspired by 3D modeling approaches developed in the field of computer aided design (CAD), where 3D objects are generated from user-drawn 2D strokes. Such tools have been proposed by Igarashi et al., Karpenko et al. and Nealan et al. for example [IMT99, KHR02, NSACO05]. CAD-based approaches, however, have not yet been used for editing of medical image segmentations. In contrast to general CAD modeling approaches, the segmentation of a medical image is given in parallel slices from which we can derive contours. This makes the reconstruction of a 3D object easier because we do not have to infer the 3D geometry of the user-drawn contour and its relationship to the object. A well-known method for smooth surface reconstruction from unorganized point clouds is called variational in-

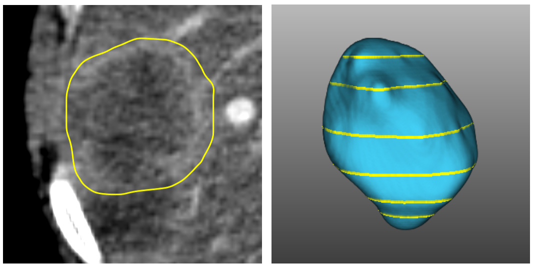

(a) (b)

Figure 2: Variational-interpolation-based segmentation of a liver metastasis in CT using 6 parallel contours.

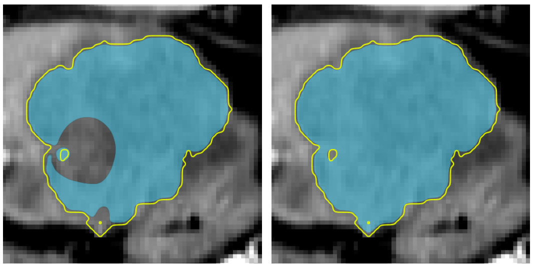

(a) (b)

Figure 3: Hole-correction example for a set of contours generated from an initial segmentation: (a) result without hole-correction and (b) result after correcting the sign of the normal constraints based on the level of embedding.

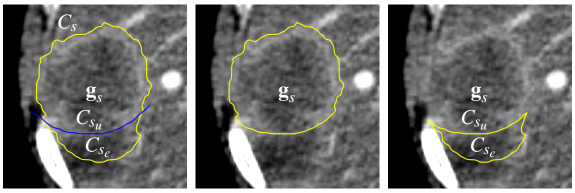
terpolation. It has first been described by Turk and O'Brien [TO99]. We have discussed the application of variational interpolation to contour-based interactive segmentation previously [HKHP11].

3. Variational Interpolation with Hole-correction

In variational interpolation methods an object is represented by an implicit function $f(\mathbf{x})$ that evaluates to zero for each point on the surface of the object. $f(\mathbf{x})$ is an approximation of the real object based a set of constraints given by a point cloud, which contains points that are known to be on the surface (surface constraints $\mathbf{c}^S$), as well as additional points that are known to be inside or outside of the object. As discussed in [HKHP11], these additional points can be derived from the local normal in each contour point for example (normal constraints $\mathbf{c}^N$). Using an appropriate radial basis function (RBF) $\phi(\mathbf{x})$, $f(\mathbf{x})$ can be written as

$$f(\mathbf{x}) = P(\mathbf{x}) + \sum_{j=1}^{k} w_j \phi(\mathbf{x} - \mathbf{c}_j), \qquad (1)$$

where $\mathbf{c}_j$ denote the constraints, w_j are the weights of each RBF and $P(\mathbf{x})$ is a degree-one polynomial that accounts for the linear and constant portions of $f(\mathbf{x})$. With $\phi(\mathbf{x}) = \|\mathbf{x}\|^3$ Eq. 1 generates a smooth C^2-continuous surface (see Fig. 2). $P(\mathbf{x})$ as well as the weights w_j are computed by solving a system of linear equations that is given by inserting all constraints in Eq. 1. A quality-preserving constraint reduction can be used to speed up computations so interactive rates are achieved. This reduction is controlled by a quality factor q, which defines the portion of contour points that are used for the reconstruction (see [HKHP11] for details).



(a) (b) (c)

Figure 4: Sketch-based editing in 2D: (a) initial segmentation C_s (yellow), user input C_{s_u} (blue) and the edited contour C_{s_e}, (b) intended correction result containing the center of gravity $\mathbf{g}_s$ and (c) result not containing $\mathbf{g}_s$, defining the edited region.

The contours from which the surface is generated can be arbitrarily oriented. The mask in a slice can contain holes, however, which is not handled by the algorithm proposed in [HKHP11]. This results in an incorrect reconstruction of the surface in such cases as shown in Fig. 3a. Holes can be handled by checking for each contour, whether it is completely contained within another contour by applying a point-in-polygon test for each contour point. As a contour can be included in a contour that itself is surrounded by another contour, this needs to be done recursively defining a *level of embedding*. A hole is then given by a contour whose level of embedding is odd. For a contour defining a hole, the sign of the function value $f(\mathbf{x})$ needs to be inverted for each normal constraint, resulting in a robust reconstruction of the object (see Fig. 3b).

4. Sketch-based Segmentation Editing

4.1. Contour-based Correction in 2D

In 2D, sketch-based interaction provides a rather simple yet precise and intuitive manual correction in the contour domain. An insufficiently segmented slice s is corrected by drawing a contour along the correct object border (see Fig. 4a). Based on this *user contour* C_{s_u} a specific part of the contour of the initial segmentation C_s is replaced (see Fig. 4b). This can be interpreted as adding some part to the segmentation, cutting away some part, combinations of both or replacing the segmentation as shown in Fig. 5. Note that this does not have to be differentiated in the contour domain.

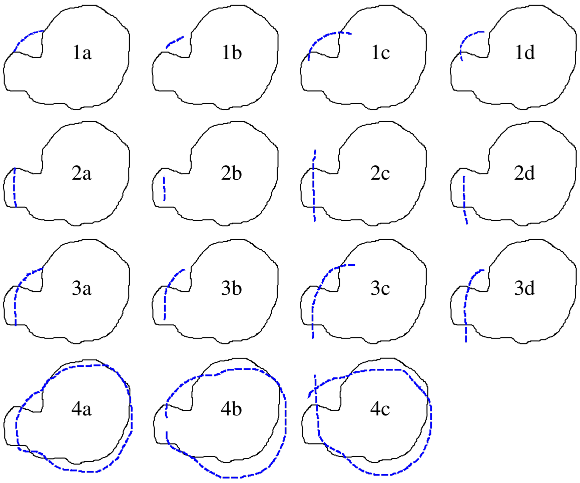


Figure 5: Common user inputs (blue dashed lines) for 2D sketch-based corrections in the contour domain that we have observed during evaluations: (1a-d) add, (2a-d) remove, (3a-d) add + remove and (4a-c) replace.

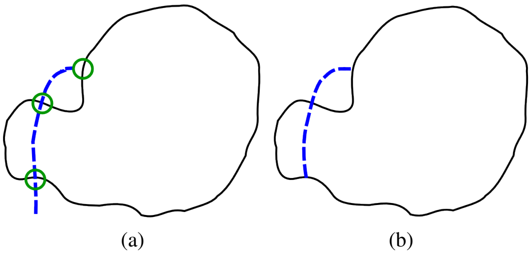


Figure 6: Handling of imperfections in sketch-based editing: (a) user-input (blue dashed line), intersection points with initial contour as well as projected start/end point (green circles) and (b) estimation of the user's intention.

The contours C_s can be efficiently derived from the segmentation mask using a marching squares algorithm [Map03].

The user input will be subject to imperfection, such that C_{s_u} might not be connected to the initial segmentation or it might intersect it in an arbitrary number of points (see Fig. 5). For robustly handling those cases we clamp C_{s_u} to the first and the last intersection point. If C_{s_u} does not intersect the initial segmentation near the first and the last point of C_{s_u}, we project those points on C_s, i.e., the points of C_s that are closest to the start and end points are used to extend C_{s_u} (see Fig. 6). If more than one initial contour exists in slice s, we use the one that is closest to the first point of C_{s_u}. In addition, the user has the possibility to replace the initial segmentation by drawing a self-intersecting or closed contour.

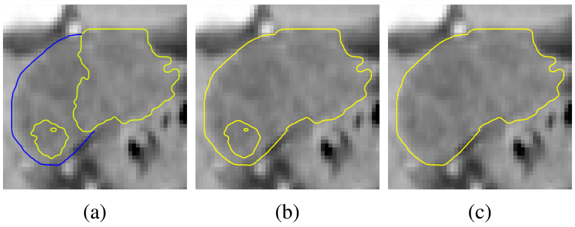


Figure 7: Hole-generation example: (a) initial segmentation (yellow) and user input (blue), (b) result without deleting newly embedded contours (c) intended result after deleting contours whose level of embedding has changed.

The user input given by an open contour is ambiguous, so there are always two possible results for one initial contour C_s and the user contour C_{s_u} as shown in Fig. 4. We use the following heuristics that decides what contour is most likely to be the user's intended result and which assumes that the object is compact:

1. Keep the contour that contains the center of gravity $\mathbf{g}_s$ of C_s.
2. If both or none of the candidates contain $\mathbf{g}_s$ keep the contour with the largest area.

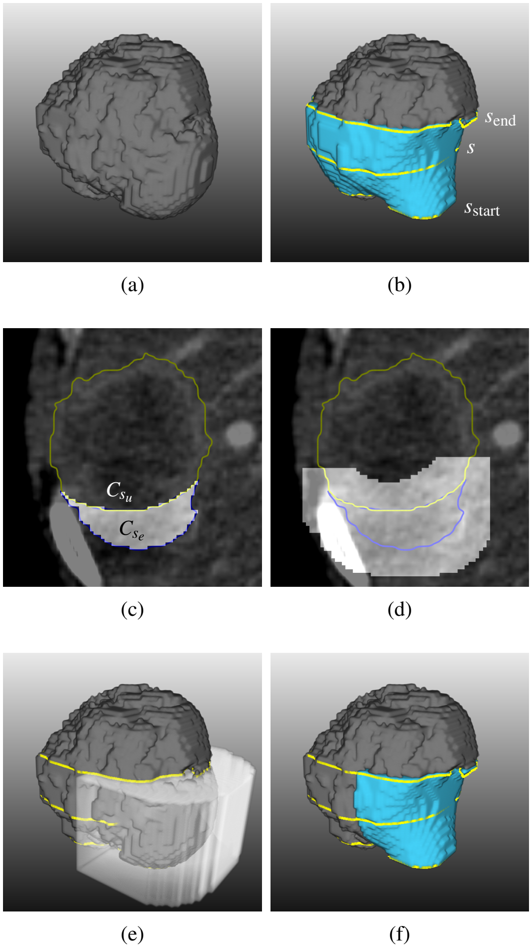
The result that is not kept by this heuristic encloses the region where the segmentation has been edited, i.e., the added or removed part in slice s (see Fig. 4c). We call the edited part of the initial segmentation C_{s_e}, while the whole edited region is given by $C_{s_u} \cup C_{s_e}$.

The user might generate new holes in the edited slice by drawing a contour around an existing part of the segmentation that is initially not connected to the edited part of the segmentation as shown in Fig. 7. The user's intention in this case is, however, to include the separated part into the segmentation. Thus, we remove all contours for which the level of embedding has changed.

4.2. 3D Extrapolation using Variational Interpolation

To transform the 2D correction into 3D, we use contours from the initial segmentation in two additional slices to reconstruct a corrected segmentation that fits into the initial segmentation. These slices are defined based on the *edited slice* s and a *correction depth* $d(C_{s_u})$, which depends on the geometrical properties of the edited region. It represents a trade-off between the necessary number of correction steps and the risk of replacing too much of the initial segmentation. Our evaluations have shown that, for tumor segmentations, the extent in z-direction of the edited region often corresponds to its thickness in the current slice. Therefore, we compute $d(C_{s_u})$ by

$$d(C_{s_u}) = \left\lceil \frac{\max_i \{\min_j \{|C_{s_e}[i] - C_{s_u}[j]|\}\}}{d_s} \right\rceil, \qquad (2)$$



(a) (b)

(c) (d)

(e) (f)

Figure 8: 3D extrapolation of the correction using variational interpolation: (a) the initial segmentation (gray), (b) result after replacing the segmentation between s_{start} and s_{end} with the variational interpolation result (blue), (c) voxelization of the edited region $C_{s_e} \cup C_{s_u}$, (d) dilation of the edited region with a 15×15 kernel (cp. Eq. 4), (e) duplication to all slices between s_{start} and s_{end} and (f) final result after masking.

where $C_{s_e}[i]$ and $C_{s_u}[j]$ refer to the i-th point of C_{s_e} and the j-th point of C_{s_u} respectively. d_s is the distance between adjacent slices in millimeters. In order not to interpolate beyond the initial segmentation, s_{start} and s_{end} are clamped to the minimum and maximum slice $s_{\min}$ and $s_{\max}$ of the initial segmentation, so the additional slices are given by

$$\begin{aligned} s_{\text{start}} &= \max\left(s - d(C_{s_u}), s_{\min}\right) \\ s_{\text{end}} &= \min\left(s + d(C_{s_u}), s_{\max}\right). \end{aligned} \tag{3}$$

Typically, the user needs to perform several correction steps and we cannot assume that slices which have already been edited are segmented correctly. In addition, each user-drawn contour should be part of the final segmentation result after each manual correction step. Therefore, we store all user contours and include them in the reconstruction process as well. This allows the user to arbitrarily switch between different views during the correction process without replacing already performed corrections. The user contours are only considered as surface constraints, because they are not closed and we cannot infer information about what should be inside or outside from them, which would be necessary for defining the normal constraints.

Based on the closed contours in the slices s, s_{start} and s_{end} as well as all user contours C_{s_u} a new segmentation is generated by variational interpolation, which replaces the initial segmentation between s_{start} and s_{end}. Because s_{start} and s_{end} represent the initial segmentation and because of the properties of variational interpolation, the new segmentation continuously fits into the initial segmentation (see Fig. 8b).

A manual correction might separate an initially coherent segmentation into several parts. Our algorithm aims at segmentation of individual compact objects, though. Therefore, we perform a connected component analysis and only keep the component that contains the center of gravity of C_s (see also Sec. 4.1). If the center of gravity is outside of the resulting segmentation we keep the component that is closest to it.

4.3. Making the Correction Local

Replacing the whole segmentation between s_{start} and s_{end} also replaces parts of a segmentation that are actually correct. We need to restrict the correction to a region, where the user intends a correction. Based on the edited region $C_{s_u} \cup C_{s_e}$, we define a region $\mathbb{I}_e \subset \mathbb{I}$ where the interpolated result is used, with $\mathbb{I}$ being all voxels of the image. For all voxels in $\mathbb{I} \setminus \mathbb{I}_e$ the original segmentation is kept (see Fig. 8c-f). We call this step *masking* in the following. $\mathbb{I}_e$ is derived from the edited region as follows:

1. Voxelization of $C_{s_u} \cup C_{s_e}$
2. Dilation of resulting mask using an appropriate kernel
3. Duplication of dilated mask to slices in $[s_{\text{start}}, s_{\text{end}}]$

The size k of the kernel for the dilation operation needs to be adapted to the size of the object, because the masking must cover a larger region for large objects and a smaller region for small objects. We made k proportional to the radius of a sphere, whose volume equals the volume of the initial segmentation given by the number of voxels n:

$$k = 2\left\lceil \frac{1}{4}\sqrt[3]{\frac{3n}{4\pi}} \right\rceil + 1. \tag{4}$$

On our training data, this has provided a good coverage of the edited region while it efficiently prevented the correction from changing too much in regions where no correction was intended.

4.4. Appending, Replacing and Segmentation from Scratch

Sometimes parts on top or below the object are missing in the segmentation so the user needs to add something by drawing a closed contour in a slice that has not yet been segmented. In this case we do not extrapolate beyond the edited slice, so we set

$$s_{\text{start}} = \begin{cases} s_{\max} & \text{if } s > s_{\max} \\ s & \text{if } s < s_{\min} \end{cases}$$
$$s_{\text{end}} = \begin{cases} s & \text{if } s > s_{\max} \\ s_{\min} & \text{if } s < s_{\min} \end{cases} . \tag{5}$$

The masking step is skipped and the interpolated result is used for all slices between s_{start} and s_{end}. Finally, $s_{\min}$ and $s_{\max}$ need to be updated.

If a segmentation in a slice is completely erroneous the user typically wants to replace it by drawing a closed contour. C_{s_u} is used directly in this case and the initial segmentation $C_s = C_{s_e}$ in slice s is discarded. Again, the masking is skipped.

This replacement approach can also be used for segmentation from scratch, i.e., if no segmentation is available, by only using C_{s_u} for computing the segmentation. In this case, we have no initial segmentation and Eq. 2 cannot be used for computation of $d(C_{s_u})$. Instead, we assume the shape of the object to be roughly spherical in this case and $d(C_{s_u})$ is computed as the radius of a circle that has the same area $A(C_{s_u})$ as C_{s_u}, i.e.,

$$d(C_{s_u}) = \left\lceil \sqrt{\frac{A(C_{s_u})}{\pi}} \right\rceil . \tag{6}$$

5. Results and Discussion

Our manual correction algorithm has been evaluated by 2 technical experts with 6+ years of experience in tumor segmentation and assessment. The evaluation was performed on 89 tumors with an initially insufficient segmentation calculated by a dedicated automatic tool [BDK*07]. The data contained a representative set of lung nodules, liver metastases and lymph nodes of different shape, size and image quality. It was collected from several clinics and CT scanners. Correction was possible in axial, sagittal and coronal view and the experts were told to generate an acceptable segmentation within a time that would be acceptable from their point of view. Both participants got a short introduction to the tool and processed one training example. The quality of the manual correction tool in terms of its suitability for the specific segmentation task was rated on a 5-point scale for each case. The results are shown in Tab. 1.

Rating	Meaning	Number of cases	Percentage
++	Perfect	14	15.7%
+	Good	43	48.3%
0	Acceptable	25	28.1%
−	Bad	6	6.7%
−−	Unacceptable	1	1.1%

Table 1: Quality ratings of the evaluation of our image-independent segmentation editing algorithm (n=89).

The experts needed a median number of 4 steps and a median time of 53s, including the time for assessment and rating. Note that an undo operation was counted as one step as well. Overall, our algorithm was well suited for 92.1% of the segmentation tasks (i.e., cases with a rating of acceptable or better), while about 7.9% could not be corrected adequately within an acceptable time. As our algorithm is image-independent, we expect those results to apply to other compact objects and other modalities as well.

Problem cases that cannot be corrected adequately with our method are most often caused by contradictory user-drawn contours as shown in Fig. 9. In such cases the variational interpolation algorithm does not always reconstruct an appropriate surface. This can in particular happen if corrections are performed in different views. The solution proposed in [HKHP11] that uses an approximation instead of an interpolation can only compensate this for minor contradictions in our scenario. Another problem class are segmentations where the larger part of the contour does not belong to the object. In such cases, the wrong contour is kept (cp. Sec. 4.1). This can be handled by drawing a closed contour around the part that should be kept, though. Moreover, filling holes is also not directly supported by sketching in the contour domain and must be done by replacing the segmentation in the affected slice or by editing in another view. Finally, it is currently not possible to delete the segmentation in a slice completely. Possible solutions could be to add some gesture recognition feature and define a specific “delete” gesture or to provide a certain button or key for this purpose on the ap-

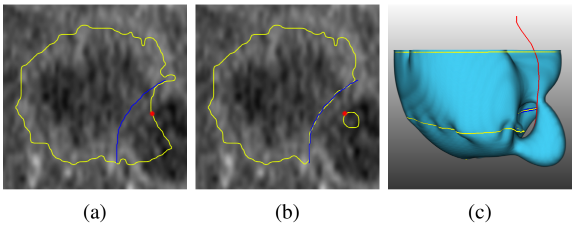


Figure 9: Example for contradictory manual corrections: (a) result (yellow) after the 1st correction step (red, drawn in another view), 2nd user input (blue), (b) result after 2nd correction step and (c) variational interpolation result in 3D.

plication level. In addition, this editing problem can almost always be solved by editing in another view.

Having an efficient and intuitive tool that produces sufficient results with only a few steps in 92% of the cases is very helpful in practice. Current state-of-the-art automatic tumor segmentation algorithms provide a sufficient segmentation in about 87% of the cases as reported by Moltz et al. [MBK*09]. Combining this with our tool means that most lesions can be segmented with little work (about 4 additional correction steps). For the remaining cases a slice-wise correction can still be used if the segmentation is mandatory. This smoothly integrates into the contour-based workflow of our tool and can be achieved by switching into a 2D mode using a modifier key for example, which prevents our algorithm from performing the variational interpolation step. The 2D mode was not supported during our evaluations, though.

To ensure a real-time response of the variational-interpolation-based segmentation algorithm, we only use 20% of the contour points for surface reconstruction. We have used 4 threads for the computations, because most PCs have 4 physical CPU cores today. With these settings, one correction step takes 0.5s on average on an Intel Xeon X5550 for a tumor consisting of about 70,000 voxels in a region of interest (ROI) of 100^3 voxels (see Fig. 7). This is fast enough to allow an editing in real time for tumor segmentations. In our evaluation, the average computation time of a single correction step was 0.4s. The computation time is proportional to the size of the object. In a liver segmentation correction scenario (about 2,000,000 voxels in a $460 \times 365 \times 110$ ROI), the average computation time was about 4.8s, which no longer allows a real-time feedback.

6. Conclusion and Outlook

We have discussed sketch-based segmentation editing in the contour domain and we have shown how variational interpolation can be used for efficient editing of tumor segmentations in 3D medical images. For making variational interpolation more robust in this context, we have extended it by a hole-correction. Because it does not rely on image information, our algorithm can be used for any 3D modality and is applicable for a wide range of other objects as long as they are compact and not too large. It can even be used in a slice-wise manner and if no initial segmentation is available. Because it only needs the segmentation mask and the user input for computation, it can be combined with any automatic segmentation algorithm. In conclusion, the presented algorithm provides a universal and intuitive editing tool for segmentation of compact objects in 3D medical images. An evaluation on segmentations of solid tumors has proven that our algorithm allows an efficient editing of such segmentations.

Future work will focus on further improving our algorithm in terms of calculation time and segmentation quality so it can be applied to larger and more complex objects, like organs. We will also investigate solutions for resolving contradictory user inputs more robustly.

The comparison of different manual correction algorithms and tools is particularly hard, because the results strongly depend on the user and on the specific segmentation task. To our knowledge, no measure exists in the literature that allows an objective comparison of these special kinds of interactive segmentation tools. To overcome this, we propose a segmentation editing challenge where different tools are applied to the same problems by the same users.

Acknowledgment

This work was funded by Siemens AG, Healthcare Sector, Imaging & Therapy Division, Computed Tomography, Forchheim, Germany.

References

[BBB*07] Beichel R., Bauer C., Bornik A., Sorantin E., Bischof H.: Liver segmentation in CT data: A segmentation refinement approach. In *3D Segmentation in The Clinic: A Grand Challenge, MICCAI Workshop* (2007), Springer, pp. 235–245. doi:10.1007/s10462-011-9220-3. 2

[BBS06] Bornik A., Beichel R., Schmalstieg D.: Interactive editing of segmented volumetric datasets in a hybrid 2D/3D virtual environment. In *ACM symposium on Virtual reality software and technology* (2006), pp. 197–206. doi:10.1145/1180495.1180536. 2

[BDK*07] Bornemann L., Dicken V., Kuhnigk J.-M., Wormanns D., Shin H.-O., Bauknecht H.-C., Diehl V., Fabel M., Meier S., Kress O., Krass S., Peitgen H.-O.: OncoTREAT: a software assistant for cancer therapy monitoring. *International Journal of Computer Assisted Radiology and Surgery 1*, 5 (2007), 231–242. doi:10.1007/s11548-006-0059-z. 2, 6

[ECFN11] Egger J., Colen R. R., Freisleben B., Nimsky C.: Manual refinement system for graph-based segmentation results in the medical domain. *Journal of Medical Systems* (08 2011), 1–11. Online First. doi:10.1007/s10916-011-9761-7. 2

[GFL06] Grady L., Funka-Lea G.: An energy minimization approach to the data driven editing of presegmented images/volumes. In *International Conference on Medical Image Computing and Computer Assisted Intervention* (2006), vol. 2, Springer, pp. 888–895. doi:10.1007/11866763_109. 2

[HAB95] Hinshaw K. P., Altman R. B., Brinkley J. F.: Shape-based models for interactive segmentation of medical images. In *SPIE Medical Imaging: Image Processing* (1995), vol. 2434, SPIE, pp. 771–780. doi:10.1117/12.208750. 2

[HKHP11] Heckel F., Konrad O., Hahn H. K., Peitgen H.-O.: Interactive 3D medical image segmentation with energy-minimizing implicit functions. *Computers & Graphics: Special issue on Visual Computing for Biology and Medicine 35*, 2 (2011), 275–287. doi:10.1016/j.cag.2010.12.006. 3, 6

[HM09] Heimann T., Meinzer H.-P.: Statistical shape models for 3d medical image segmentation: A review. *Medical Image Analysis 13*, 4 (2009), 543–563. doi:10.1016/j.media.2009.05.004. 2

[HMB*09] Heckel F., Moltz J. H., Bornemann L., Dicken V., Bauknecht H.-C., Fabel M., Hittinger M., Kiessling A., Meier S., Püsken M., Peitgen H.-O.: 3D contour based local manual correction of tumor segmentations in CT scans. In *SPIE Medical Imaging: Image Processing* (2009), vol. 7259, SPIE, p. 72593L. doi:10.1117/12.812248. 2

[HTKM04] Heimann T., Thorn M., Kunert T., Meinzer H.-P.: New methods for leak detection and contour correction in seeded region growing segmentation. In *International Archives of Photogrammetry and Remote Sensing* (2004), vol. 35, pp. 317–322. 2

[HvGS*09] Heimann T., van Ginneken B., Styner M. A., Arzhaeva Y., Aurich V., Bauer C., Beck A., Becker C., Beichel R., Bekes G., Bello F., Binnig G., Bischof H., Bornik A., Cashman P. M. M., Chi Y., Córdova A., Dawant B. M., Fidrich M., Furst J. D., Furukawa D., Grenacher L., Hornegger J., Kainmuller D., Kitney R. I., Kobatake H., Lamecker H., Lange T., Lee J., Lennon B., Li R., Li S., Meinzer H.-P., Németh G., Raicu D. S., Rau A.-M., van Rikxoort E. M., Rousson M., Ruskó L., Saddi K. A., Schmidt G., Seghers D., Shimizu A., Slagmolen P., Sorantin E., Soza G., Susomboon R., Waite J. M., Wimmer A., Wolf I.: Comparison and evaluation of methods for liver segmentation from ct datasets. *IEEE Transactions on Medical Imaging 28*, 8 (aug. 2009), 1251–265. doi:10.1109/TMI.2009.2013851. 2

[IMT99] Igarashi T., Matsuoka S., Tanaka H.: Teddy: A sketching interface for 3D freeform design. In *ACM SIGGRAPH* (1999), ACM, pp. 409–416. doi:10.1145/311535.311602. 2

[JSG03] Jackowski M., Satter M., Goshtasby A.: Approximating digital 3D shapes by rational gaussian surfaces. *IEEE Transactions on Visualization and Computer Graphics 9*, 1 (2003), 56–69. doi:10.1109/TVCG.2003.1175097. 2

[KEK04] Kang Y., Engelke K., Kalender W. A.: Interactive 3D editing tools for image segmentation. *Medical Image Analysis 8*, 1 (2004), 35–46. doi:10.1016/j.media.2003.07.002. 2

[KHR02] Karpenko O., Hughes J. F., Raskar R.: Free-form sketching with variational implicit surfaces. *Computer Graphics Forum 21*, 3 (2002), 585–594. doi:10.1111/1467-8659.t01-1-00709. 2

[KSH*04] Kunert T., Schroter A., Heimann T., Schobinger M., Bottger T., Thorn M., Wolf I., Engelmann U., Meinzer H.-P.: An interactive system for volume segmentation in computer-assisted surgery. In *SPIE Medical Imaging: Visualization, Image-Guided Procedures, and Display* (2004), vol. 5367, p. 799. doi:10.1117/12.535096. 2

[Map03] Maple C.: Geometric design and space planning using the marching squares and marching cube algorithms. In *International Conference on Geometric Modeling and Graphics* (2003), pp. 90–95. doi:10.1109/GMAG.2003.1219671. 4

[MaRC11] Miranda P. A. V., ao A. X. F., Ruppert G. C. S., Cappabianco F. A. M.: How to fix 3D segmentation interactively via image foresting transform and its use in MRI brain segmentation. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (2011), pp. 2031–2035. doi:10.1109/ISBI.2011.5872811. 2

[MBK*09] Moltz J. H., Bornemann L., Kuhnigk J.-M., Dicken V., Peitgen E., Meier S., Bolte H., Fabel M., Bauknecht H.-C., Hittinger M., Kiessling A., Püsken M., Peitgen H.-O.: Advanced segmentation techniques for lung nodules, liver metastases, and enlarged lymph nodes in ct scans. *IEEE Journal of Selected Topics in Signal Processing 3*, 1 (2009), 122–134. doi:10.1109/JSTSP.2008.2011107. 7

[MFT*08] Maleike D., Fabel M., Tetzlaff R., von Tengg-Kobligk H., Heimann T., Meinzer H.-P., Wolf I.: Lymph node segmentation on CT images by a shape model guided deformable surface method. In *SPIE Medical Imaging: Image Processing* (2008), vol. 6914, p. 69141S. doi:10.1117/12.770352. 2

[MNMW09] Maleike D., Nolden M., Meinzer H.-P., Wolf I.: Interactive segmentation framework of the medical imaging interaction toolkit. *Computer Methods and Programs in Biomedicine 96*, 1 (2009), 72–83. doi:10.1016/j.cmpb.2009.04.004. 2

[NSACO05] Nealen A., Sorkine O., Alexa M., Cohen-Or D.: A sketch-based interface for detail-preserving mesh editing. In *ACM SIGGRAPH Papers* (2005), ACM, pp. 1142–1147. doi:10.1145/1281500.1281553. 2

[OS01] Olabarriaga S. D., Smeulders A. W. M.: Interaction in the segmentation of medical images: A survey. *Medical Image Analysis 5* (2001), 127–142. doi:10.1016/S1361-8415(00)00041-4. 2

[PDP10] Proksch D., Dornheim J., Preim B.: Interaktionstechniken zur Korrektur medizinischer 3D-Segmentierungen. In *Bildverarbeitung für die Medizin* (2010), vol. 574, Springer, pp. 420–424. 2

[PFJ*03] Pizer S. M., Fletcher P. T., Joshi S., Thall A., Chen J. Z., Fridman Y., Fritsch D. S., Gash A. G., Glotzer J. M., Jiroutek M. R., Lu C., Muller K. E., Tracton G., Yushkevich P., Chaney E. L.: Deformable M-Reps for 3D medical image segmentation. *International Journal of Computer Vision 55* (2003), 85–106. doi:10.1023/A:1026313132218. 2

[RDDP10] Rahner S., Dornheim J., Dornheim L., Preim B.: Interaktive techniken zur korrektur medizinischer segmentierungen auf basis stabiler feder-masse-modelle. In *Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie* (Düsseldorf, November 2010), pp. 159–164. 2

[SHT*08] Schwarz T., Heimann T., Tetzlaff R., Rau A.-M., Wolf I., Meinzer H.-P.: Interactive surface correction for 3D shape based segmentation. In *SPIE Medical Imaging: Image Processing* (2008), vol. 6914, SPIE, p. 69143O. doi:10.1117/12.770350. 2

[SSMS10] Silva S., Santos B., Madeira J., Silva A.: A 3D tool for left ventricle segmentation editing. In *International Conference on Image Analysis and Recognition Image Analysis and Recognition* (2010), vol. 6112, Springer, pp. 79–88. doi:10.1007/978-3-642-13775-4_9. 2

[TO99] Turk G., O'Brien J. F.: Shape transformation using variational implicit functions. In *ACM SIGGRAPH* (1999), ACM, pp. 335–342. doi:10.1145/311535.311580. 3

[TPvB*03] Timinger H., Pekar V., von Berg J., Dietmayer K., Kaus M.: Integration of interactive corrections to model-based segmentation algorithms. In *Bildverarbeitung für die Medizin* (2003), vol. 80, Springer, pp. 171–175. 2

[YPH*06] Yushkevich P. A., Piven J., Hazlett H. C., Smith R. G., Ho S., Gee J. C., Gerig G.: User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *NeuroImage 31*, 3 (2006), 1116–1128. doi:10.1016/j.neuroimage.2006.01.015. 2