

An interactive 3D segmentation for the Medical Imaging Interaction Toolkit (MITK)

Andreas Fetzer, Sascha Zelzer, Tobias Schroeder, Hans-Peter Meinzer, and
Marco Nolden

German Cancer Research Center, Heidelberg, Germany
`a.fetzer@dkfz.de`

Abstract. An accurate segmentation of anatomical structures in 3D medical images is an essential step for many clinical and scientific tasks. Since the manual delineation in single slices can be very time-consuming and cumbersome, a lot of research on semi- or fully automated segmentation methods is performed. However, many of these 3D techniques can have drawbacks in practical use, e.g. requiring a complex initialization or failing on images of lower quality, which can occur frequently in routine use. In this work we introduce a fully interactive 3D segmentation for the Medical Imaging Interaction Toolkit (MITK). It is based on a 3D surface interpolation using radial basis functions and is seamlessly integrated into the manual segmentation toolset of MITK. We additionally extended the segmentation tools to allow the delineation of contours in arbitrary orientations, which introduces the possibility to capture the shape of the region of interest with a minimum number of contours. Use cases then demonstrate the improved segmentation workflow based on the proposed methods.

1 Introduction

The segmentation of anatomical structures is an important requirement for various tasks like assessment and quantification of target structures or visualization and model creation e.g. for image-guided interventions or medical simulations. While fully or semi-automated segmentation techniques deliver good results in a short amount of time they are often restricted to certain imaging modalities or anatomical structures or they require arguably complex initialization. Although they are a prevalent topic of research one must not disregard the pure manual segmentation. For many use cases the manual delineation of target structures like it is done in the field of radiotherapy is still the means of choice. Furthermore manual expert segmentations are needed for the validation of (semi-) automated algorithms. However the manual segmentation can be time-consuming and cumbersome depending on the region of interest and the segmentation tools at one's disposal.

Maleike et al. introduced in [1] a comprehensive manual segmentation framework for the Medical Imaging Interaction Toolkit (MITK) an open source cross-platform application framework and software library for medical imaging [2]. A

shape-based 2D interpolation for parallel slices was applied in order to reduce the time needed for a manual segmentation.

In this work we introduce an interactive 3D segmentation for MITK which is based on a 3D surface interpolation using radial basis functions (RBFs). To fully leverage the potential of this approach, the existing segmentation framework was extended to allow for 2D segmentation in arbitrary orientations, which makes it easier to capture the shape of the region of interest using a small number of contours.

The design of the 3D segmentation was influenced by an in-house usability survey of the segmentation framework of MITK, which was conducted by us. The survey comprised five users consisting of biologists, physicists and physicians being differently adept at using the segmentation in MITK. During the survey beside user interviews also questionnaires and user observations were done. The results of the survey identified the following key requirements and expectations for an image segmentation application:

1. Save the current state, continue after application restart
2. Undo/redo the latest actions
3. Easy navigation through the dataset

2 Related work

Medical image segmentation is addressed by various applications. 3D Slicer [3], Seg3D¹ and ITK Snap [4] for example are open source, cross platform applications, which come with some manual or semi-automated segmentation tools whereas none of them provides a fully interactive 3D segmentation.

RBFs are used by Wimmer et al. [5] for interpolating the surface based on control points that can be placed by the user. In a second step the interpolation result is then used as an initialization of a level set algorithm.

TurtleSeg², a closed source application which is freely available for Windows, uses a 3D live wire algorithm which was introduced by Hamarneh et al. [6]. The user can delineate contours using a 2D live wire tool and at any time trigger the 3D live wire algorithm, which computes a 3D reconstruction based on the provided contours. Additionally TurtleSeg has a feature called *Spotlight*, which was introduced by Top et al. [7]. *Spotlight* constitutes a user guidance for placing contours by highlighting regions where the current 3D segmentation has a potential high deviation from the underlying region of interest.

Another approach is the one proposed by Heckel et al. [8] which was developed using MeVisLab.³ They use highly optimized variational interpolation for surface reconstruction based on user drawn contours. The segmentation is pure contour-based which is why it is not possible to cut out inner parts e.g. to interpolate hollow structures.

¹ Seg3D: www.seg3d.org

² TurtleSeg: www.turtleseg.org

³ MeVisLab: <http://mevislab.de>

3 System description

In this section we give a brief overview about the MITK Workbench, an application that is used for performing the interactive 3D segmentation. Furthermore we describe the segmentation tool framework which was introduced by Maleike in [1] and how we extended it in order to provide a fully interactive 3D segmentation.

3.1 The MITK Workbench application

MITK includes a ready-made application - the *MITK Workbench* - a highly extendable and customizable end user application which provides a number of plugins for medical imaging [2]. It offers extensive functionality for visualizing and processing medical image data from various modalities as well as related data like surfaces or landmarks. For data navigation the MITK Workbench has among other means the so called *multiwidget*, a multiplanar reconstruction view where the data is displayed. It consists of four renderwindows, three of them display the data in 2D and one displaying the scene in 3D. In each 2D window the other sectional planes are rendered as a crosshair which can be moved or rotated by the user in order to easily navigate through the current dataset.

3.2 Enhancement of MITK's interactive segmentation framework

Maleike introduced in [1] a sophisticated class framework for manual segmentation tools in MITK. By using one of the 2D segmentation tools like *live wire*, *region growing* or simply a *contouring tool*, the user can create new contours for a selected image slice. This contour is converted to a binary mask and written back into the image. However the tools could only be applied to the reference anatomical planes, i.e. no manual segmentation in oblique planes was possible. In order to allow the user to capture the shape of the segmented structure using just a few contours we enhanced the tool framework so that manual contouring in arbitrary orientations is possible. Therefore we unified the way of how the images are rendered and the way of how the slices are extracted from the image volume during a segmentation. For resampling an arbitrary slice from a volume the *vtkImageReslice*⁴ is used which allows the extraction of slices from any orientation. For writing the segmented slice back into the image volume we derived from *vtkImageReslice* and implemented analogously a version that writes back a slice instead of extracting it, maintaining the correct voxel mapping.

Another limitation was that the contour information itself was discarded after the slice was written back into the image. Hence we extended the framework such that the delineated contours are kept and can be persisted across application restarts. In case of segmentation tools that are not contour based, e.g. a simple thresholding tool, we extract the contour points from the corresponding 2D slice.

⁴ VTK: www.vtk.org

In that way every slice based segmentation tool can be used for the proposed 3D segmentation.

These steps constitute essential preparatory work for realizing an interactive and usable 3D segmentation, which is in line with the requirements derived from the user survey. Fig. 1 shows the segmentation in rotated planes and the displayed contour information in 3D.

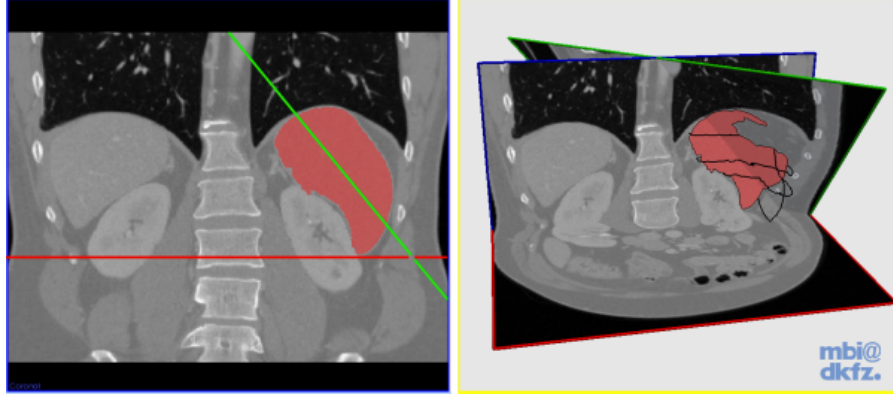


Fig. 1. Segmentation of the spleen in rotated orientation. In the coronal window on the left the rotated intersection plane is displayed as green line. The right image shows the sectional planes of the image and the contours that were drawn as black lines in a 3D scene.

3.3 The 3D segmentation

For the 3D segmentation the existing contour points are used for interpolating a 3D surface. Therefore we represent the surface in an implicit way using a signed distance function. By definition of the distance function the distance value zero is assigned to the provided contour points. In order to guarantee a valid interpolation result we have to compute *off-surface points* that have a certain distance to the surface. For this we approximate the normals for the given contour points and add, respectively subtract them from the according contour point. Points inside the surface will get distance values less than zero and vice versa for points outside the surface. The interpolation itself is then done using radial basis functions similar to the approach of Carr et al. [9] which has the advantage that there is no restriction to the position of the contours which allows arbitrary orientations. The distance function in our case is defined as follows:

$$d(x) = \sum_{i=1}^n \lambda_i \cdot \Phi(\|x - x_i\|_2) \quad \begin{cases} < 0, & \text{if } x \text{ lies inside the surface} \\ = 0, & \text{if } x \text{ lies on the surface} \\ > 0, & \text{if } x \text{ lies outside the surface} \end{cases} \quad (1)$$

Hereby x is the image coordinate for which the distance value has to be calculated, n is the number of existing contour and off-surface points, λ_i are the interpolation weights and Φ is the biharmonic RBF. The interpolation weights are determined by solving the following equation system where c_i stands for the distance value of given the contour or off-surface points x_i :

$$\begin{pmatrix} \Phi(\|x_1 - x_1\|_2) & \Phi(\|x_1 - x_2\|_2) & \dots & \Phi(\|x_1 - x_n\|_2) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi(\|x_n - x_1\|_2) & \Phi(\|x_n - x_2\|_2) & \dots & \Phi(\|x_n - x_n\|_2) \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \quad (2)$$

The zeros of the interpolated distance function describe the pathway of the interpolated 3D segmentation. Using the distance function we calculate a distance image that encloses the segmented area.

In order to reduce the computational time for the interpolation several measures for optimization were taken. First of all we reduce the number of points for each contour. Therefore we use the algorithm of Douglas et al. [10] which reduces the points according to a defined error tolerance. However the method of Douglas et al. delivers far to little sampling points, which has a negative effect on the interpolation result. To address that we modified the method of Douglas et al. so that in addition to the reduction still a regular distribution of the contour points is guaranteed.

Second we do not calculate the distance values for the whole image but start at a given contour point and move through the image along a narrow band similar to a region growing. Pixels with a distance value above a defined threshold will immediately be discarded. For both the contour point reduction and the distance value calculation the tolerance thresholds are set according to the minimum pixel spacing of the underlying image.

Since simultaneously to the contour points also the according binary segmentation exists we can easily determine whether a given point is inside or outside the desired segmentation and hence basically any shape can be interpolated, even hollow structures. Fig. 2 shows the interpolation of a pipe which was delineated free-handed.

3.4 Measures for usability

In this section we describe the measures, which were taken in order to meet the requirements which were identified by the usability survey mentioned above. Since the user at any time has the possibility to edit an existing contour we have to keep track of both the contour and its position so that the stored contours

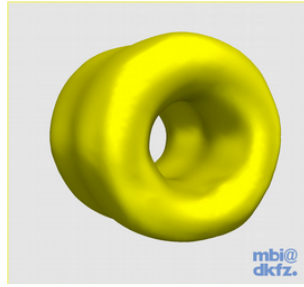


Fig. 2. The interpolation of a pipe, which demonstrates that our approach even works for hollow structures. The contours for this purpose were drawn free-handed.

always correspond to the binary mask. The contour position is used to provide an easy navigation between the segmented slices, which is especially helpful if the contours are located in rotated orientations and which allows a convenient amendment of existing contours.

Since the segmentation tools already support undo and redo of the user interaction we implemented the same mechanism for the 3D interpolation. As soon as a tool interaction is undone or redone the 3D interpolation will be updated accordingly.

The latest interpolation result is displayed both in a 3D render window as a 3D surface mesh and in 2D as the mesh’s intersecting contour. The user can easily determine areas where the interpolation deviates significantly from the region of interest by navigating through the dataset, e.g. via dragging or rotating the crosshair in the multiwidget and observing the intersection contour in the different planar reconstructions. Fig. 3 shows the yellow 2D interpolation feedback during the segmentation of a kidney. The reason for the deviation in the left image is that at this stage of the segmentation the number of the delineated contours is not sufficient for a proper shape description of the kidney and hence for a good interpolation result. By providing additional contours in such areas using the manual segmentation tools it is possible to interactively refine the resulting 3D segmentation. As soon as the user is satisfied with the interpolation result the current surface can be written into the binary mask resulting in a complete 3D segmentation of the considered region. Another requirement was to have the possibility to save the current interpolation state and continue the 3D segmentation e.g. after application restart. Therefore the segmentation session can be saved as MITK scene file, including the grey value image, the segmentation binary mask and the corresponding contours. After loading the scene again the contour information is then used to re-initialize the interpolation.

3.5 Unit testing and validation

To assure the quality of the implementation of the interpolation method within the framework, unit tests were implemented which validate the interpolation

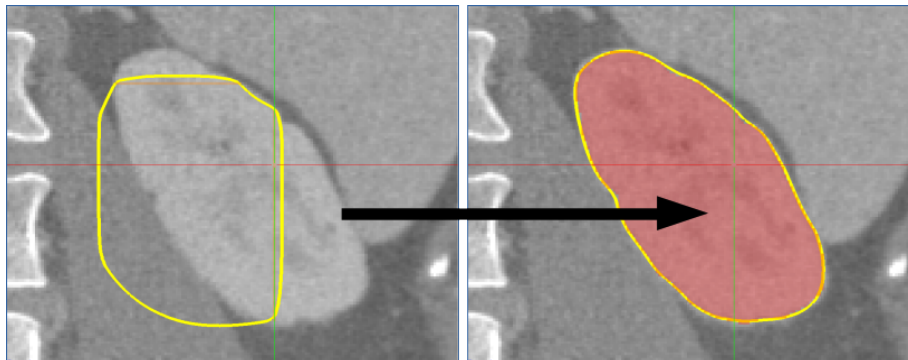


Fig. 3. The correction of the interpolation result in a slice with significant deviation. In the left window one can see that the yellow intersecting contour of the interpolated surface deviates distinctly from the kidney in this slice, which is caused by the fact that at this stage of the segmentation, insufficient contour information is provided in this area of the image. The right window shows the surface intersection after an additional contour was provided. The surface now runs closely along the kidney border.

pipeline for different structures, regularly comparing the results with reference datasets within a continuous integration system. In addition, within the Workbench release process [2], several checklists are manually run through to verify the correct interplay of the manual segmentation tools with the interpolation.

4 Evaluation

In this section we evaluate how our 3D segmentation performs regarding both the computational time and the accuracy compared to expert segmentations. The evaluation was performed on an Apple iMac with 3.4 GHz i7 processor and 16 GB memory. Finally we present use cases in which the proposed 3D segmentation was already applied successfully.

4.1 Results

For the evaluation we extracted automatically 2D slices from expert segmentations of respectively three livers and gall bladders. The extraction was done at four equidistant positions for each of the three reference anatomical planes resulting in 12 slices added together. We then extracted the contour points from each of the slices and used them as input for the 3D interpolation. The interpolation result was compared with the expert segmentations using the metrics introduced by Heimann et al. [13]. The results are displayed in Table 1. The positions of the extracted contours are shown in Fig. 4.

As we can see even a rather low number of 12 contours is sufficient for a reasonable interpolation result with a volumetric overlap error less than 10%.

Table 1. The results of the evaluation compared to expert segmentations of three different livers and gall bladders. For the interpolation automatically four slices for each of the three reference anatomical planes were extracted at equidistant positions. Additional to the number of points (n_{points}) including the off-surface points also the computation time was taken. Compared to the expert segmentations we evaluated the mean and maximum surface distance and the volumetric overlap error.

Structure	ID	n_{points}	Computation time [s]	Mean Dist. [mm]	Max. Dist. [mm]	Vol. Overlap Error [%]
Liver	1	1716	3.81	1.43	17.15	9.28
	2	1944	5.26	2.00	41.10	9.0
	9	2265	7.89	1.27	29.03	7.40
Gallbladder	1	537	0.47	0.44	4.12	6.65
	2	453	0.33	0.41	4.35	8.01
	9	351	0.29	0.39	3.00	9.96

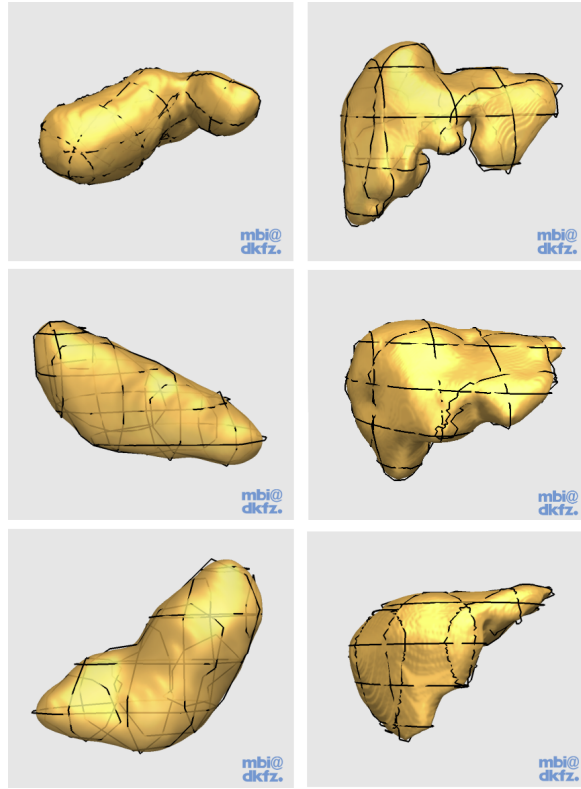


Fig. 4. The positions of the contours, which are used for the interpolation during the automated evaluation. From top down the datasets 1,2 and 9 are displayed with the gall bladder on the left-hand side and the liver on the right. The contours are displayed as black lines.

4.2 Use cases

In this section we want to present use cases in which the proposed 3D segmentation improved existing workflows.

First, it is a very efficient tool for the generation of ground truth segmentations, a necessary prerequisite for the validation of semi- or fully automated algorithms. Given the fact that these have to be created by experts with often limited time, a high level of efficiency and usability can be an advantage.

One use case in which the proposed 3D interpolation is deployed is outlined in the work of Mueller et al. “Mobile augmented reality for computer-assisted percutaneous nephrolithotomy” [12]. In the proposed workflow 3D objects have to be generated from pre-operative imaging in order to evaluate the intra-operative guidance of the surgeon during the needle insertion. Using our interactive 3D segmentation tool the kidney can be segmented reliably in 8 minutes, sufficiently fast for the workflow.

Another case of application is *mint Lesion*⁵ an FDA approved MITK based product which facilitates the assessment of the effectiveness of a cancer therapy. It provides a tool called *Interpolated Volume*, which is re-using our publicly available open source implementation.⁶

5 Discussion

Table 1 demonstrates that our method achieves good results. Unlike pure 3D based segmentation algorithms our approach relies completely on user defined 2D contour information for interpolating a 3D surface. This gives the user full control over the segmentation result at any time. Working on arbitrary orientations, our approach provides great flexibility for both the contour position and the shape of the structure itself. As a result our method is independent from the underlying imaging modality and applicable for basically any anatomical structure, independent of its contrast or intensity values within the image. In general the interpolation benefits from well placed contours, which describe the shape of the region of interest in an optimal way. However Table 1 shows that even with equidistantly placed contours a good result can be achieved.

In order to present immediate interpolation feedback several optimizations were implemented like the reduction of the contour points. However since we have to solve the linear equation system in (2) for interpolating the distance function the memory consumption of $\mathcal{O}(n^2)$ constitutes the limiting factor for the number of contour points that can be used for the interpolation.

Moreover the computational costs for solving the equation system are high. As we can see from Table 1 the interpolation slows down for an increasing number of contour points. While the interpolation for the gall bladder with $n_{\text{points}} < 600$ is pretty fast with approximately 0.5s, the interpolation for the liver needs noticeably more time with 8s for $n_{\text{points}} > 2000$ contour points. Since the global

⁵ *mint Lesion*TM: Mint Medical GmbH, www.mint-medical.de, Heidelberg, Germany.

⁶ www.mitk.org

interpolation has to be recalculated all over again after refining or adding contours this constitutes a limitation regarding the usability. A way to address this could be the usage of a more optimized solver for the linear system like it was investigated by Heckel et al. [8]

6 Conclusion and future work

We introduced a fast, robust and versatile method for an interactive 3D segmentation. During an in-house usability survey we identified certain key requirements for an image segmentation application. These requirements influenced the design and implementation of the proposed method.

The 3D interpolation on the one hand is seamlessly integrated into the segmentation workflow of MITK. On the other hand it is completely independent from the tools used for providing contour information, which facilitates the usage of any slice-based segmentation tool. Since simultaneously to the contour information also the binary segmentation is used we can even interpolate hollow structures. The fact that the user delineates the contour information manually makes the proposed method independent from the underlying imaging modalities and applicable to a large variety of anatomical structures.

Thanks to the introduced measures of usability like the position tracking of the contours, undo/redo and the convenient way for navigating through the datasets the user can easily find areas of significant deviation or amend existing contours.

Although computation time increases for bigger structures the proposed method already proved to be useful in various use-cases and workflows. The regular validation with unit tests and manual checklists during the MITK Workbench release process guarantees a constant quality of the implementation.

Future work will include a feasibility evaluation for both, using the interactive 3D segmentation as an initialization method for non-interactive algorithms like level-sets or shape-based techniques and as an efficient correction tool for the results of automatic segmentation methods.

Furthermore we could improve the usability of our tool in different ways: First, a possibility for pausing or explicitly triggering the interpolation would be useful. That could address the drawback of the automatic recalculation of the interpolation after every contour amendment. Second, since the quality of the interpolation result depends on how well the user places the contours another possible improvement could be a guidance for placing the contours similar to TurtleSeg’s *Spotlight*.

Disclaimer and acknowledgment

This work was carried out with the support of the German Research Foundation (DFG) as part of project S01, SFB/TRR 125 Cognition Guided Surgery.

References

- [1] Maleike, D., Nolden, M., Meinzer, H.-P. and Wolf, I.: Interactive segmentation framework of the Medical Imaging Interaction Toolkit. *Computer Methods and Programs in Biomedicine* 96, 72–83 (2009)
- [2] Nolden, M., Zelzer, S., Seitel, A., Wald, D., Mueller, M., Franz, A. M., Maleike, D., Fangerau, M., Baumhauer, M., Maier-Hein, L., Maier-Hein, K. H., Meinzer, H.-P. and Wolf, I.: The Medical Imaging Interaction Toolkit: challenges and advances. *Int. J. Computer Assisted Radiology and Surgery*, 604-620, (2013)
- [3] Pieper, S., Halle, M. and Kikinis, R.: 3D Slicer. *IEEE International Symposium on Biomedical Imaging*, 632–5, (2004)
- [4] Yushkevich, P. A., Piven, J., Hazlett, C., Smith, H., Smith, G., Ho, R., Ho, S., Gee, J. C. and Gerig, G.: User-Guided 3D Active Contour Segmentation of Anatomical Structures: Significantly Improved Efficiency and Reliability. *Neuroimage* 31, 1116–1128, (2006)
- [5] Wimmer, A., Soza, G. and Hornegger, H. Two-Staged Semi-automatic Organ Segmentation Framework using Radial Basis Functions and Level Sets. *3D Segmentation in Clinic*, 179-188, 2007
- [6] Hamarneh, G., Yang, J., McIntosh, C. and Langille, M.: 3D live-wire-based semi-automatic segmentation of medical images. *Proceedings of SPIE Medical Imaging: Image Processing* 5747, 1597-1603, (2005)
- [7] Andrew Top, Ghassan Hamarneh, and Rafeef Abugharbieh. Spotlight: Automated Confidence-based User Guidance for Increasing Efficiency in Interactive 3D Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention Workshop on Medical Computer Vision (MICCAI MCV)*, pages 204-213, 2010.
- [8] Heckel, F., Konrad, O., Hahn, H. K. and Peitgen, H.-O. Interactive 3D medical image segmentation with energy-minimizing implicit functions. *Computers & Graphics* 35, 275 - 287, 2011
- [9] Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques, SIGGRAPH*, 6776. ACM, (2001)
- [10] Douglas DH, Peucker TK. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10, 112122, 1973
- [12] Mueller, M., Rassweiler, M.-C., Klein, J., Seitel, A., Gondan, M., Baumhauer, M., Teber, D., Rassweiler, J., Meinzer, H.-P. and Maier-Hein, L. Mobile augmented reality for computer-assisted percutaneous nephrolithotomy. *Int. J. Computer Assisted Radiology and Surgery*, 663-675, (2013)
- [13] Heimann T, van Ginneken B, et al. Comparison and Evaluation of Methods for Liver Segmentation From CT Datasets. *IEEE Trans Med Imaging*. 2009;28(8):12511265.