NEW METHODS FOR LEAK DETECTION AND CONTOUR CORRECTION IN SEEDED REGION GROWING SEGMENTATION

T. Heimann, M. Thorn, T. Kunert, H.-P. Meinzer
German Cancer Research Center, Heidelberg, Germany
Email: t.heimann@dkfz.de

KEY WORDS: Medicine, Application, Segmentation, Algorithms, Correction

ABSTRACT:

Segmentation, i.e. the labelling of objects in image data, is a crucial step in many medical imaging processing tasks, e.g. operation planning, radio therapy or diagnostics. Seeded region growing is a basic yet effective method for semi-automatic segmentation. Its major drawback is that poor contrast at the organ edges may result in leaks, letting the area grow far beyond the region of interest. We developed a new method to reliably detect the origins of these leak regions and correct the segmentation results. Our approach is based on the observation that leaks are normally characterized by a narrow bottleneck connection to the seed area. We detect this bottleneck by specifying a single point somewhere within the erroneous area and tracing a path back to the seed point, moving along the skeleton of the segmented region. The point on the skeleton with the minimal distance to the contour marks the bottleneck. To cope with minor deviations from the desired result (not featuring a characteristic bottleneck), we optimized our freehand tool to minimize the necessary user interaction. Instead of separately cutting or merging parts of the segmentation, the new tool allows modifications by replacing entire parts of the contour in a single step.

KURZFASSUNG:

Segmentierung, d.h. die Markierung bestimmter Objekte in Bilddaten, ist ein kritischer Schritt bei vielen Aufgaben in der medizinischen Bildverarbeitung, wie Operationsplanung, Strahlentheraphie oder Diagnose. Das Regionenwachstumsverfahren ist eine grundlegende, aber wirkungsvolle Methode zur halbautomatischen Segmentierung. Der größte Nachteil liegt darin, dass niedriger Kontrast an Organrändern zum Auslaufen führen kann und die segmentierte Fläche weit über den interessierenden Bereich hinaus wächst. Wir haben eine neue Methode entwickelt, um den Ursprung dieser ausgelaufenen Regionen zuverlässig zu finden und das Ergebnis zu korrigieren. Unser Ansatz basiert auf der Beobachtung, dass die ausgelaufene Region normalerweise nur eine schmale Verbindung zur ursprünglichen Fläche besitzt. Dieser Flaschenhals kann identifiziert warden, indem man einem beliebigen Punkt innerhalb der ausgelaufenen Fläche ein Pfad zum Ursprung entlang des Skeletts der Segmentierung berechnet wird. Der Punkt auf dem Skelett mit der niedrigsten Distanz zur Kontur markiert den Flaschenhals. Um kleinere Abweichungen vom gewünschten Segmentierungsergebnis zu korrigieren, haben wir unser Freihandwerkzeug optimiert, um die notwendige Benutzerinteraktion zu minimieren. Anstatt einzelne Teile der Region hinzuzufügen oder zu löschen, ersetzt das neue Werkzeug komplette Abschnitte der Kontur in einem einzigen Schritt.

1. INTRODUCTION

During the last years, a growing effort in medical imaging research has been put into the development of fully automatic methods for segmentation to be able to cope with the ever growing amounts of data modern CT and MRT scanners are able to produce. While this trend is important and paving the way into the future, it is often overlooked that in clinical practice, the methods of choice are still classic semi-automatic tools like freehand segmentation, snakes or region growing. One reason for this situation might be that the automatic methods are not yet robust enough for the clinical practice, another one that radiologists prefer more personal control and flexibility when working with their images. But whatever the reasons may be: To support the clinical workflow of today, it is essential to enhance and improve the tools which are actually used.

This article presents two examples of such enhancements that have been developed for use in the surgery department of the university clinic of Heidelberg. Here, radiologists utilize the teleradiology and PACS system Chili (Engelmann et al, 2000) for image access. This software implements a plug-in concept for specialized applications, one of which is the segmentation of structures of interest that is performed with the Segmenta plugin (Kunert et al, 2004). Currently, this plug-in is mainly used for segmenting CT images for liver operation planning (Meinzer et al, 2002), a task which has been fully integrated into the clinical workflow. The most time-consuming part of the planning is the segmentation of liver tissue, which is performed on a slice by slice basis using semi-automatic tools. In order to optimize the liver operation planning workflow, a selection of these tools was reengineered with a main focus on the concepts of usability and ergonomics (Nielsen, 1994). An important source of input for enhancements and simplifications was a user survey that has been conducted with experts who have been using the older version of the segmentation software extensively.

In section 2 of this article, the new developments are presented from a technical point of view. Section 3 exposes the results of a comparison study between the old and the new versions of the tools and section 4 closes the article with a conclusion.

2. MATERIALS AND METHODS

All tools use the Segmenta plug-in as a common platform for event managing and graphical display. The plug-in itself is implemented in C++ using QT (Dalheimer, 2002) and OpenGL (Woo, 1999).

2.1 Extensions to the region grower

The employed region grower tool uses the basic techniques of seeded region growing (Gonzalez and Woods, 2002): From a user-defined seed point, the segmentation grows while neighbouring pixels lie within a specific grey value interval. Although this method can deliver excellent segmentation results with minimal interaction when image contrast is good, the tool has rarely been used in the old version of our segmentation software. The user survey revealed that this objection was caused mainly by the complicated parameter input, which was designed to enter the grey value interval with the keyboard and confirm it by pressing the *Enter* key. We therefore strived to make the use of the region grower as intuitive as possible. Furthermore, all additional filtering should become obsolete. In the case of liver tissue segmentation, a common filter to be employed after region growing is a closing operator to fill holes inside the segmented area.

2.1.1 User interaction scheme: The adjustment of the grey value interval in which the region should grow was implemented using two vertical sliders instead of text input. In this constellation, the upper slider represents the upper border and vice versa. The bigger the distance between the two sliders, the larger the grey value interval for region growing. Both upper and lower border are relative values to the mean grey value in the area around the seed point. In order to supply the user with a direct feed-back of what the current settings produce, a preview of the segmentation result is displayed after every slider movement.

When specifying the seed point, the user can hold the mouse button down and drag the mouse to control both sliders symmetrically. Thus, the grey value interval for region growing can be increased or decreased on the fly. Releasing the mouse button fixes the segmentation results in the image. In images with good contrast, this method is completely sufficient to segment liver tissue.

In case where it fails, both sliders can be operated independently after toggling an *Adjust* button at the controls: Releasing the mouse button after specifying the seed point does not fix the result in this mode, enabling the user to try different upper and lower interval borders and different seed points until getting a satisfying result. To remind the user that he has to toggle *Adjust* again to accept the result, the preview is displayed in a different colour as long as this mode is active.

Another button that has turned out to be very useful is the *Fix* button, which allows freezing the current absolute upper and lower interval borders for subsequent seed points. Once the optimal values have been found for a CT data set, this allows an exact reproduction in all slices.

2.1.2 Contour extraction and smoothing: The preview of the segmentation result is displayed using only the outline of the region. This has the advantage that the closing operator to fill holes does not need to be executed every update. Moreover, rendering a line is much faster than creating an overlay texture of the corresponding region, which is beneficial for the real-time constraints of the feed-back. When the segmentation is fixed, the contour can be used to produce the final segmentation result by scan conversion (Heckbert, 1990), thus eliminating the need for an additional closing operation.

In contrast to practically all contour extraction algorithms presented in literature, e.g. (Pavlidis, 1982), our algorithm delivers the exact boundaries which lie between the actual pixel positions. Since each pixel has four sides, there are four possible directions the contour can take. For the following example, we will assume that the contour is traced below the object pixels to the right. As figure 1 demonstrates, there are four possible cases for the next contour point. In the situation presented, contour following has reached the second dark pixel, the black points on the contour have already been recognized and stored. The white arrows show the course of the extraction until now and the next possible positions and directions. If there is another object pixel below the current position (A), the contour direction is rotated to the right and this pixel becomes the next to work with. Otherwise it is tested if the object continues to the right (B): If yes, the white point is stored as a new contour point and the position is shifted to the right. If this is not the case, the lower right pixel is queried (C): If it is part of the object, it becomes the next pixel to work with, the white contour point is stored and contour direction turns right. Else, case D applies: contour direction turns left, position does not change and a new point is added.

The presented algorithm allows a fast contour extraction on binary data. To reduce the effects of noise, we included the option to filter the image data before contour extraction, using a 5x5 pixel median filter as default.

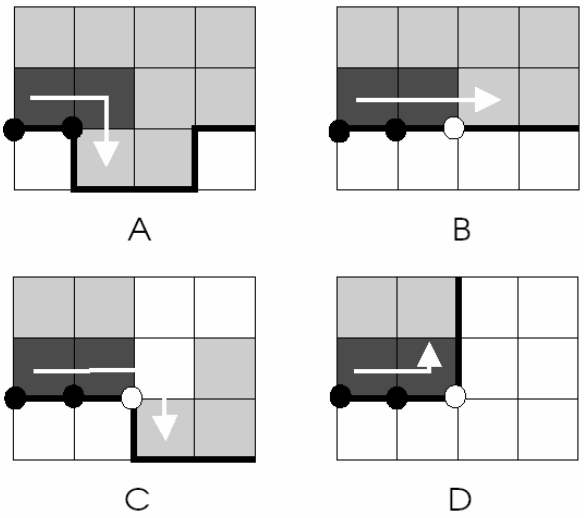

Figure 1. The four cases encountered in contour extraction. All images show a part of the lower section of the object of interest: Pixels already completed are coloured dark; the contour to be extracted is displayed as a black line.

2.1.3 Leak detection: Even when using an optimal grey value interval, the region growing algorithm can leak out if contrast at the object boundary is not sufficient. In case of the liver, especially the regions next to stomach and kidney are susceptible to leaking, since these organs have similar grey values as the liver itself. Often, the origin of the leak is only a narrow connection between neighbouring tissues. Since the time region growing algorithms are employed in segmentation, solutions that are both fast and easy to control have been searched to solve this problem. Often, the segmentation software allows the user to draw a line which blocks the region growing and thus prevents leakage. However, the most user-friendly approach would require just one mouse click in the leak area to remove it. The only proposed solution with this interaction scheme (Sekiguchi and Sano, 1994) has a major draw-back: it only works with very narrow connections.

We propose a new method using distance transformation to identify leak regions and remove the additional area. The basic idea is as follows: From an arbitrary point within the erroneous area, we calculate the path to the seed point which always maintains the maximum possible distance to the surrounding edges. The point on this path which after a local maximum has the shortest distance to the contour marks the narrowest bottleneck to the chosen region and thus the origin of the leak. The constraint that a local maximum has to be passed first prevents that the starting point of the path is marked as origin if the user clicked at a position too close near the edge.

A direct implementation of this idea uses the skeletonization or medial axis transformation (Lee, 1982) of the object of interest. There are several different definitions for the MAT of a shape, one being the following: assuming the shape is of a burnable material and is set on fire simultaneously on all sides, it will burn down towards the center. Where several fronts of fire meet, they extinguish themselves and the resulting line graph is the skeleton of the shape.

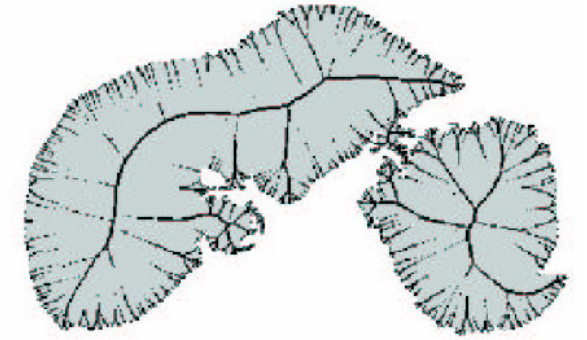

Figure 2. Skeletonization of a leaked liver shape by thresholding the gradient magnitudes of the distance image. All pixels with a gradient magnitude lower than 0.95 are marked black.

When the path follows this skeleton as much as possible, the maximum distance to the object border is maintained automatically. A fast implementation to test if points belong to the skeleton or not is based on the gradient magnitudes of the distance image of the shape, which can efficiently be calculated using the algorithm presented in (Arya, 1998). All points where this magnitude is zero belong to the skeleton. Due to inaccuracies in the distance transformation of quantized images, it is better in practice to test if the gradient magnitude lies below a specific threshold. For an example of this method see figure 2.

Being able to determine if a pixel is part of the skeleton or not, the next step consists of finding the shortest path over this skeleton. An extensive graph search on the entire skeleton is the most obvious solution to this problem, but to save the inherent costs of computation, we opted for a different method. Our algorithm makes use of the fact that the path always leads from the periphery towards the seed-point. During the region growing phase we count at which step each pixel is added to the region, defining a geodetic distance function. During the path search, we only consider pixels with a geodetic distance lower or equal to the distance at the current position and chose among those the one with the lowest gradient magnitude. In case the lowest magnitude is lower than 0.5 (i.e. the path has not reached the skeleton yet), we alternatively chose the pixel with the largest distance to the contour.

For every pixel in the path, we check its distance to the contour for a local minimum to find bottlenecks of the shape. The smallest local minimum is marked as the origin of the leak. Subsequently, we search the two nearest, opposing points on the contour at the acquired position and separate the segmented area along a line between these two points. The part of the area where the user clicked is then removed from the segmentation. An example of this operation is shown in figure 3.

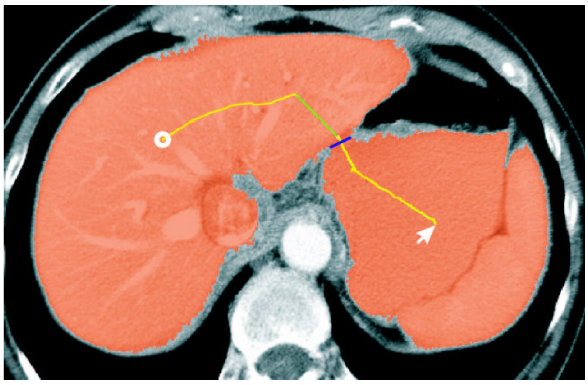

Figure 3. The region grower with seed point in the liver (white circle) has leaked out into the stomach. The path calculated by the heuristic from the point of correction is displayed yellow-green. The blue line shows where the contour will be separated at the leak origin.

2.2 The new contour correction tool

The presented leak detection for the region growing tool works in cases with a bottleneck between leaked area and origin. If this bottleneck is not present or the deviation is too small the method will not work. Additionally, as we found out in the user survey of our segmentation software, demand for an efficient correction tool is very high. The basic freehand segmentation, i.e. adding or subtracting an enclosed area, is not sufficient: Changing between the two modes is time-consuming and error-prone, moreover, it is cumbersome to revolve the entire correction area.

2.2.1 Basic Idea: To remedy these problems, the new correction tool should work directly on the contours of the segmentation. Figure 4 points out the difference to the old approach: To delete a leaked region, it should be sufficient to draw the cut line. In a similar way, the extension of an existing segmentation should be possible by just drawing the new contour. Furthermore, it should be possible to combine these operations in a single interaction, i.e. by drawing one line, the user should be able to cut the region at one place and extend it at another.

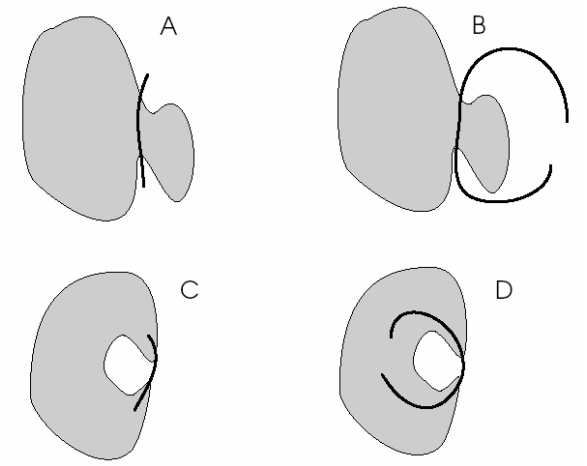

Figure 4. Required correction lines: cut operation with the new (A) and the old approach (B), merge operation with the new (C) and the old approach (D).

2.2.2 Heuristics: In order to implement this idea, we developed a set of heuristics to make the necessary decisions automatically. The decision if a cut or a merge operation should be executed is made on the existing segmentation: If the user draws a line in an unsegmented area, the region should be extended up to that line, else it should be cut at that line. One line sector is defined by two crossings with the old contour. Thus, the first steps after drawing a line with the correction tool are to find the points of intersection with the contour, split the line into the corresponding sectors and classify each sector as cut or merge.

The second heuristic has to decide which part of the region is to be modified. Both operations leave two possibilities to interpret the new contour. In figure 4a, e.g. it is not clear if the area to the left or to the right of the line should be deleted. However, since corrections generally are minor adjustments on the current result, the new contour is always interpreted in the way that results in the smallest area of change. Thus, for a cut operation always the smaller part of the region is deleted and for a merge operation the smaller part is added. If the two parts do not differ significantly, no operation is performed.

2.2.3 Implementation:

To measure the area on both sides of the drawn line, a modified region grower is used which counts all added pixels. It should be noted that there are several different possibilities to separate the two areas. Using the line drawn by the user can lead to problems in the area measurement, since areas may be split in several separated regions: In the example shown in figure 5a, the area above the line would be split in a five and a two pixel region. To avoid the resulting problems, we use two different lines to separate the areas, as shown in figures 5b and c. The dark-grey colored separators in theses figures are defined as direct neighbors of the drawn correction line, sorted by the side they belong to. To determine this side, we use an inductive procedure: Originating from an arbitrary neighbor pixel, the adjacent neighbors are marked until both ends of the line are reached. At that point, all pixels on one side of the line are known, the rest is assigned to the other side.

Finally, to determine the size of the area, region growing algorithms can be started from every point of the user-drawn line and their results are added. In most cases the entire area is captured by the first region growing, but in cases where the correction line runs close to the border of the object, several runs might be necessary. In the example in figure 5, the results are an area of 11 pixels for the upper side and 22 pixels for the lower (only for the shown section). Thus, the upper side of this line sector is filled. For the next sector in the example, the lower pixels will be removed and 5d shows the final result of the operation.

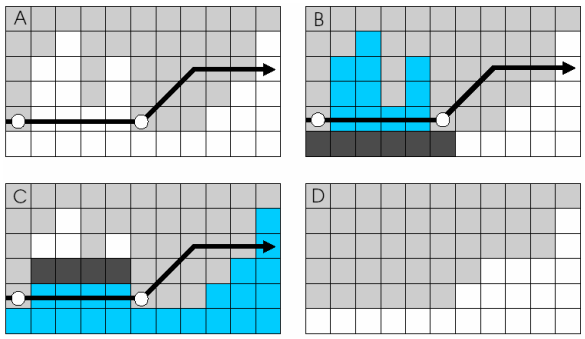

Figure 5. Procedure for a contour correction of the light-grey object: The black arrow shows the user-drawn line of correction in the original image (A), in B and C the areas of the regions on both sides of the line are determined, D shows the final result.

3. RESULTS AND DISCUSSION

We evaluated the new tools quantitatively in a study with 12 medical students on five different CT datasets. Six students worked with the old version of the segmentation tools, six with the new version. In a randomized sequence, every dataset was segmented four times by each test person. The times to create a complete segmentation of the liver with the interactive region grower and the contour modifier are on average 10% lower than the ones created with a set of standard region growing, local threshold and freehand segmentation (see figure 6). Dataset 4, the image with the lowest contrast, is the only one where the test users working with the old tools were faster. This is because in this case most of them did not bother to use the old region grower at all, which they regarded as too complicated, and used other tools instead.

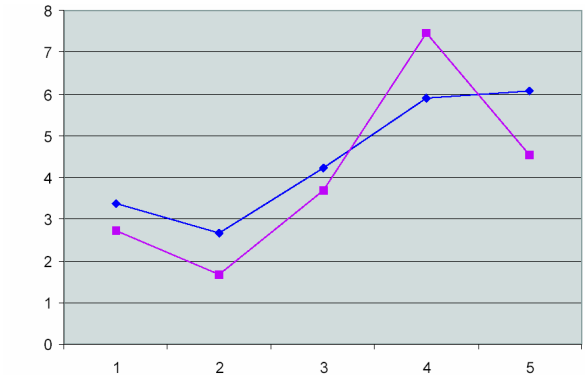

Figure 6. Average times in minutes needed for a complete segmentation of five different CT datasets. The blue line with diagonal squares shows the results for the old version of the tools, the other for the new version.

To compare the reproducibility of the segmentation results, we calculated the intra-observer accuracies by means of the Tanimoto coefficient (Tanimoto, 1958). This coefficient is 100 percent if two segmentations are completely identical and 0 if they do not overlap at all. As figure 7 shows, the intra-observer accuracies are on average maintained at the same high level as with the reference tools.

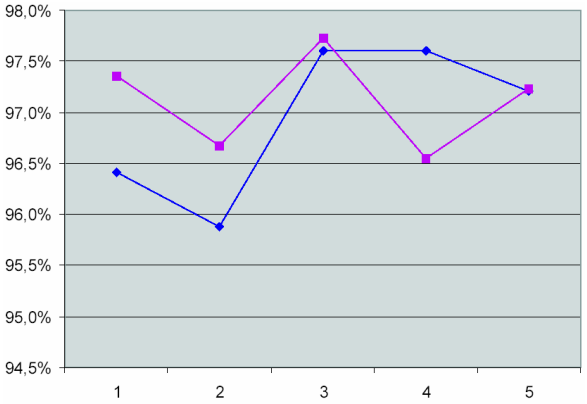

Figure 7. Intra-observer variabilities for five different datasets, measured with the Tanimoto coefficient. The blue line with diagonal squares shows the results for the old version of the tools, the other for the new version.

Additionally to these quantitative results for time and accuracy, we logged the usage of the new region grower tool to reveal more information regarding the preferred modes the users have been working with. Figure 8 shows the results of this analysis, where each mouse click with the region grower was counted as one event. As can be seen, the users took advantage of the *Fix* button to work with fixed borders of the grey value interval quite frequently. According to the diagram, only about half of all correction attempts were successful. In reality, the number where a leaked region cannot be removed is much lower, but users tended to try the removal repeatedly and clicked several times in these cases.

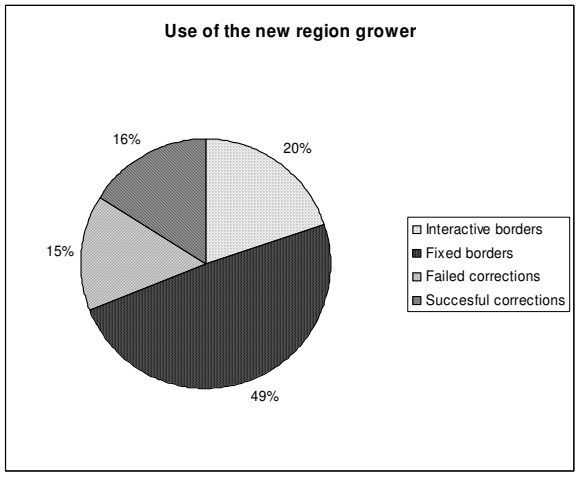

Figure 8. Use of the new region growing tool.

4. CONCLUSIONS

We introduce a new, interactive approach to detect and correct leaks in region growing segmentation that is suitable for two- and three-dimensional datasets alike. Moreover, we present a novel tool to manually correct the contours of segmented objects which minimizes the necessary user interaction.
The new user interface employed in these tools, featuring a direct feed-back of parameter changes, is approved by the users and leads to a higher acceptance of the region growing tool. The times needed for a liver segmentation in the clinical workflow could be reduced noticeable using the methods described in this paper.

REFERENCES

Arya, S., D.M. Mount, N.S. Netanyahu, R. Silverman and A.Y. Wu, 1998. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45(6), pp. 891-923.

Dalheimer, M.K., 2002. *Programming with Qt*. O'Reilly, 2nd edition.

Engelmann, U., A. Schröter, M. Schwab, U. Eisenmann, L. Bahner, S. Delorme, H. Hahne and H.-P. Meinzer, 2000. The Linux-based PACS project at the German Cancer Research Center. In: *Proc. CARS 2000*, Elsevier, Amsterdam, pp. 419-424.

Gonzalez, R.C. and R.E. Woods, 2002. *Digital Image Processing*. Addison Wesley, 2nd edition.

Heckbert, P., 1990. Concave Polygon Scan Conversion. In: Glassner, A. (editor): *Graphics Gems*. Academic Press, Boston, pp. 87-91.

Kunert, T., T. Heimann, A. Schröter, M. Schöbinger, T. Böttger, M. Thorn, I. Wolf and H.-P. Meinzer, 2004. An Interactive System for Volume Segmentation in Computer-Assisted Surgery. In: *Proc SPIE Medical Imaging 2004*.

Lee, D.T., 1982. Medial axis transformation of a planar shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(4), pp. 363-369.

Meinzer, H.-P., M. Thorn and C.E. Cárdenas, 2002. Computerized planning of liver surgery – an overview. *Computers & Graphics*, 26, pp. 569-576.

Nielsen, J., 1994. *Usability Engineering*. Morgan Kaufmann, San Francisco.

Pavlidis, T., 1982. *Algorithms for Graphics and Image Processing*. Computer Science Press.

Sekiguchi, H. and K. Sano, 1994. Interactive 3-Dimensional Segmentation Method Based on Region Growing Method. *Systems and Computers in Japan*, 25(1), pp. 88-97.

Tanimoto, T.T., 1958. *An Elementary Mathematical Theory of Classification and Prediction*. Technical report, IBM Research.

Woo, M., J. Neider, T. Davis and D. Shreiner, 1999. *OpenGL Programming Guide*. Addison Wesley, 3rd edition.

ACKNOWLEDGEMENTS

This work was funded by the Tumorzentrum Heidelberg/Mannheim.