Efficient Semiautomatic Segmentation of 3D Objects in Medical Images

Andrea Schenk, Guido Prause, and Heinz-Otto Peitgen

MeVis – Center for Medical Diagnostic Systems and Visualization
Universitaetsallee 29, 28359 Bremen, Germany
{schenk,prause,peitgen}@mevis.de

Abstract. We present a fast and accurate tool for semiautomatic segmentation of volumetric medical images based on the live wire algorithm, shape-based interpolation and a new optimization method.
While the user-steered live wire algorithm represents an efficient, precise and reproducible method for interactive segmentation of selected two-dimensional images, the shape-based interpolation allows the automatic approximation of contours on slices between user-defined boundaries. The combination of both methods leads to accurate segmentations with significantly reduced user interaction time. Moreover, the subsequent automated optimization of the interpolated object contours results in a better segmentation quality or can be used to extend the distances between user-segmented images and for a further reduction of interaction time. Experiments were carried out on hepatic computer tomographies from three different clinics. The results of the segmentation of liver parenchyma have shown that the user interaction time can be reduced more than 60% by the combination of shape-based interpolation and our optimization method with volume deviations in the magnitude of inter-user differences.

1 Introduction

Fully automated segmentation is still an unsolved problem for most applications in medical imaging due to the wide variety of image modalities, object properties and biological variability. On the other hand the most general approach, manual contour tracing, is time-consuming, inaccurate and unacceptable for large three-dimensional data sets.

To overcome these problems, a lot of work has been invested in semiautomatic segmentation methods. A popular group attracting considerable attention over the last years are the two-dimensional active contour models or *snakes* introduced by Kass et al. [1]. These algorithms try to minimize an energy functional based on contour deformation and external image forces. For initialization an approximation of the object boundary is required which is usually drawn manually by the user or in some cases is generated from a priori knowledge. However, active contours are sensitive to the settings of their numerous parameters and the quality of the initial contour which limits their applicability to medical images in the clinical routine.

Another promising approach to interactive boundary detection in two-dimensional images is the *live wire* algorithm, also known as *intelligent scissors*, which was first introduced in 1992 [4, 5]. With live wire the segmentation process is directly steered by the user who has immediate control over the automatically suggested object contours. The contours are found as minimal paths with respect to a cost function similar to the external energy function of active contours.

Semiautomatic segmentation of three-dimensional objects in volumetric medical images is often based on two-dimensional methods which is prohibitive when each slice needs to be segmented in a larger data set. Although snakes have a direct three-dimensional extension introduced as *balloons* [2] the propagation of an initial two-dimensional snake over the slices and subsequent adjustment is preferred in most applications for reasons of robustness and practicability. For live wire only one approach for segmentation of three-dimensional images has been published so far [3] based on the projection of few interactively defined contours onto orthogonal cross-sections through the volumetric image. With this approach it is difficult to keep track of the topology of complex shaped objects without major user interaction.

The goal of our work was to extend the two-dimensional live wire algorithm to an efficient and easy to use approach for semiautomatic segmentation of three-dimensional objects in medical images. This is achieved by reducing the number of interactively segmented slices and automatic calculation of all missing intermediate contours by an optimized interpolation method which is robust to topological changes. The resulting approach is validated in five hepatic CT scans from three different clinical sites.

2 Methods

2.1 Live Wire

The live wire algorithm is a user-steered segmentation method for two-dimensional images based on the calculation of minimal cost paths by dynamic programming [6] or Dijkstra's graph search algorithm [7]. Several modifications of the basic approach including quality studies have been published [8–11], proving the high accuracy, efficiency, and reproducibility of the algorithm.

The version of live wire we use in our approach is comparable to the algorithm in [11] and we resume the fundamental ideas of this method. The two-dimensional image is transformed into a directed and weighted graph. Vertices of the graph represent image pixels while edges connect neighboring pixels in two directions. The edges are weighted with local cost functions, related to the external image forces of active contours. The cost function used in our implementation is a weighted sum of different gradient functions, the Laplacian function and the gray values at the object boundary.

After the calculation of the cost graph the user starts with the segmentation by setting a first seed point on the boundary and moving the mouse along the outline of the object. Shortest paths – in the sense of paths with lowest accumulated costs – from the seed to at least the current mouse position are computed

using Dijkstra's algorithm. Computation and display of the resulting live wire boundary segment is achieved in real-time even on larger images. The live wire path snaps to the boundary while the user moves the mouse over the image and a new seed point has to be set before the path starts to deviate from the desired contour. New shortest paths are computed from the new seed point and the procedure is repeated. A final closing leads to a controlled and piece-wise optimal segmentation result.

Our live wire approach is provided with the additional concepts of boundary snapping, data-driven cooling for automated generation of seed points and learning of cost function parameters [6, 8].

2.2   Shape-Based Interpolation

Interpolation techniques [12] are used for many applications in medical imaging, e.g. image generation, resampling and visualization. They can be broadly divided into two categories: scene-based and object-based interpolation. While scene-based techniques determine the interpolated values directly from the intensities of the given image, object-based interpolation uses additional information about the objects in the image.
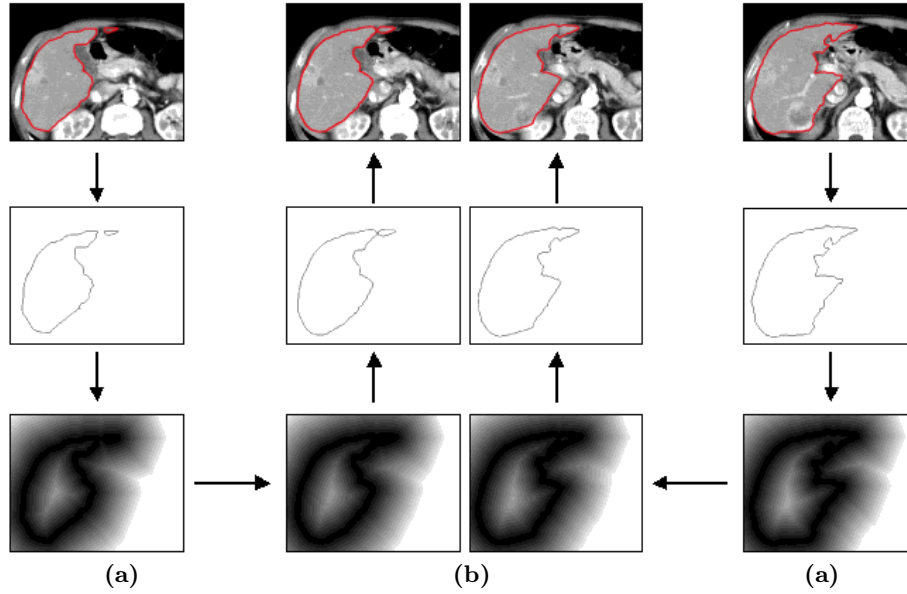

Fig. 1. Shape-based interpolation: **(a)** user-defined contour in CT images (top), binary scenes (middle), distance images (bottom); **(b)** interpolated distance images (bottom), binary scenes (middle), interpolated contours in CT images (top).

The *shape-based interpolation* method we use in our approach is the first object-based interpolation approach published in the literature [13] and it interpolates between binary images of a three-dimensional data set.

Shape-based interpolation consists of the steps illustrated in Figure 1. In the first step a binary scene is generated from a given object contour. Subsequently, the distance to this boundary is mapped into a new gray-level scene with positive distance values inside and negative distance values outside the object (shown as absolute values in Figure 1). In the third step the distance images are interpolated employing a conventional gray-level interpolation technique such as linear or higher-order interpolation. The interpolated gray-level scenes are converted back to a binary contour image by identifying the zero-crossings.

The distance transformation can be calculated efficiently with a version of the city-block distance [13] or, as in our implementation, with two consecutive chamfering processes [14, 15] realized with 3×3-kernel operations. The chamfering method leads to more accurate results since this transformation is a better approximation of the Euclidean distance.

2.3 Combination of Live Wire and Interpolation

For the segmentation of volumetric objects we have combined the user-steered two-dimensional live wire segmentation and the fully-automated shape-based interpolation. The user starts with the live wire algorithm on individually selected slices. If contours on at least two slices are available, all contours on slices in between can be computed utilizing the shape-based interpolation.

The user has free choice of applying the two methods: either live wire and shape-based interpolation in an alternating fashion, or – in a two-stage approach – interactive contour tracing first and interpolation subsequently. The only restriction is that the first and the last slice of the object of interest must be segmented interactively with the live wire algorithm.
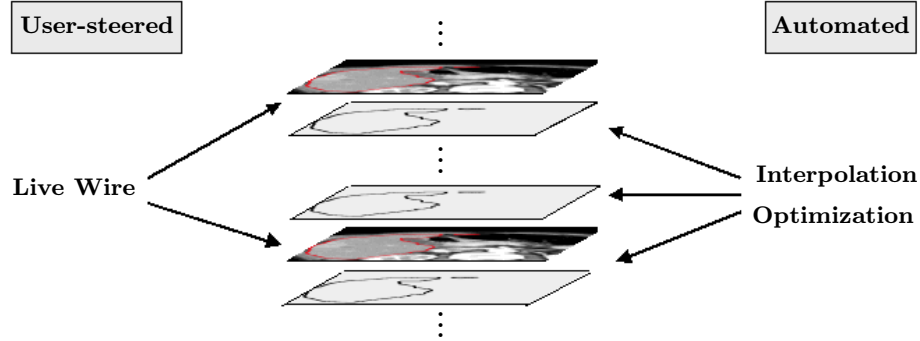

Fig. 2. Combined segmentation scheme: User-steered segmentation of selected slices followed by automatic interpolation and optimization of the intermediate contours.

If the variations of the object boundary between user-selected slices are significant, the interpolation result can be unsatisfying. In this case there is a need for interactive or automatic optimization methods which can be also used for further reduction of the number of interactively segmented image slices.

2.4 Optimization of Interpolated Contours

For the optimization of the interpolated contours we use the basic method of the live wire algorithm and recalculate contour segments as optimal cost paths between seed points with Dijkstra's algorithm.

After the shape-based interpolation the zero-crossings of the interpolated distance images define the boundary points of the new contours. To bring these points into the correct order and to identify separate contours we apply a path search algorithm. Starting with a first contour point (e.g. lower right) we follow neighboring pixels in counterclockwise direction (Figure 3a) until the first point is reached again. The algorithm is repeated until all boundary points are assigned to a contour path and it is provided with a backtracking method to deal with cases of contours connected by a single pixel line (Figure 3b).
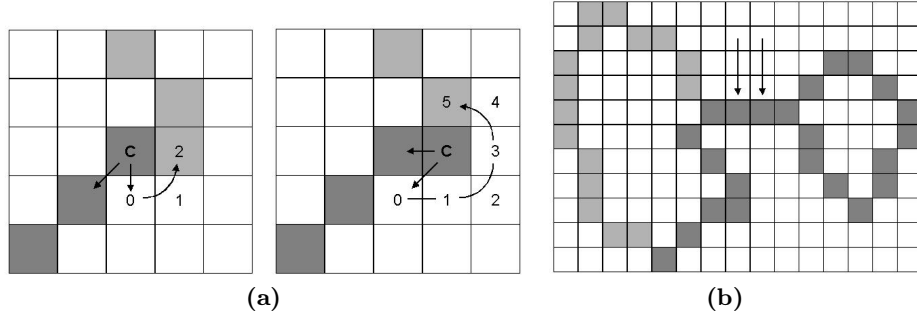

(a)   (b)

Fig. 3. Tracing the contour in the zero-crossings (gray) of the interpolated distance images. **a)** From the current pixel C of the marked path (dark gray) the next contour point is found as first (light gray) neighbor in counterclockwise direction (numbers describe the search order in the 8-neighborhood). **b)** If two boundaries are connected via a single pixel line a backtracking algorithm separates the single contours and deletes the connecting pixels (arrows).

To establish a basis for the minimal path search, the seed points from the two adjacent user-defined contours are projected onto the slices with interpolated boundaries. For every user-defined seed point the nearest point on the boundary path is determined. If two points are closer together than a given minimal distance they are merged to one central seed point. The utilization of the user-defined instead of e.g. equally distributed seeds along the contour is suggestive, since these points represent the knowledge of the user in regions of high curvature or weak edges.

The automatic contour optimization starts on all interpolated slices with the calculation of the cost matrix. Based on this data structure and the approximated seed points all contour segments linking two seeds are computed as minimal cost paths.

In the interactive optimization mode the user moves, deletes or inserts new seed points and the piece-wise optimal paths concerning these seeds are recomputed and displayed in real-time.
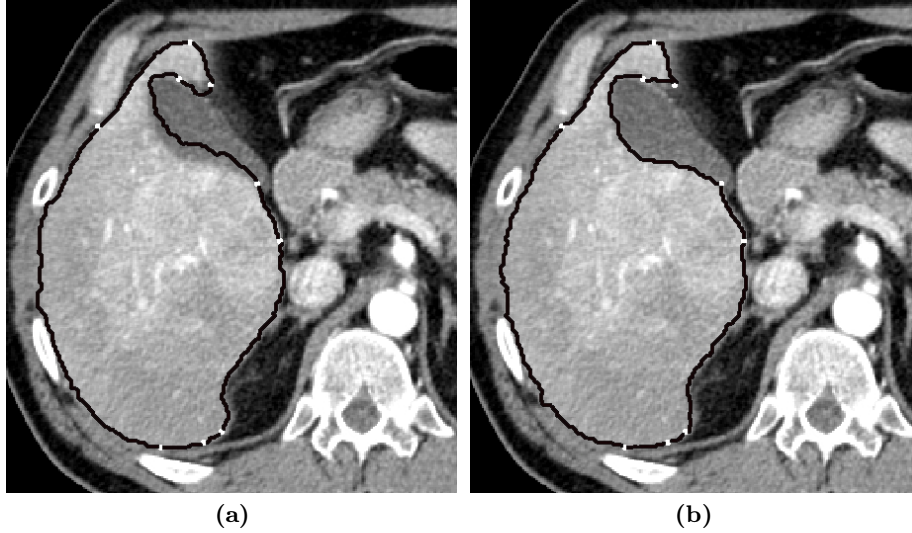

(a) (b)

Fig. 4. Liver CT with segmented parenchyma: **a)** Image with shape-based interpolated contour and approximated seed points from the two adjacent user-defined slices; larger deviations occur at the gall bladder (dark gray) **b)** contour after automated optimization.

3   Experiments and Results

Our approach has been successfully applied to CT and MR images of the liver and the lung for interactive segmentation of the parenchyma. It is part of an ongoing clinical project for the approximation of patient-specific organ segments and preoperative planning of surgery [17, 18]. Compared with the other image processing steps of this project the segmentation is the most time-consuming part.

The test presented here was designed to quantify the volumetric deviations that have to be expected when the interactive segmentation is reduced and gradually replaced by automated interpolation. For the experiment we used five diagnostic liver CT data sets (I1-I5) from three different hospitals. The underlying examinations are helical CTs during arterial portography (CTAP) (I1-I4)

and one biphasic spiral CT scan (I5, data set from the portal venous phase). All images have 512×512 pixels in plane and consist of 34–41 (I1-I4) or 86 (I5) slices with a distance of 4 mm (I1-I4) or 2 mm (I5). Three users segmented the liver parenchyma with the live wire algorithm on all slices and the time needed for the segmentation was recorded. For one image slice an average time of 6-11 seconds and 4-6 seed points were needed depending on the user. The cost function used for the test was identical for all images and was based on a Sobel-like gradient function (feature f_5 in [10] extended to 8-neighborhood) and the gray values at the object border with relative weights of 0.45 and 0.55, respectively. Parameters for both functions (mean and deviation) were determined experimentally.

Table 1. Results for the test images: inter-user variability, volume deviation for the SBI and OPT interpolation methods and saved interaction time (in %).

	Inter-user variability	Automated interpolations		Saved interaction time
		SBI	**SBI+OPT**	
Image 1	1.68			
8 mm		1.24	0.79	51.0
12 mm		2.03	1.20	65.5
16 mm		3.20	1.95	72.7
Image 2	3.47			
8 mm		2.28	1.83	47.4
12 mm		2.70	1.65	62.3
16 mm		5.70	4.35	72.8
Image 3	1.43			
8 mm		1.57	1.19	51.9
12 mm		2.87	1.98	65.3
16 mm		3.81	2.69	75.0
Image 4	1.24			
8 mm		1.34	1.17	50.1
12 mm		1.89	1.30	67.5
16 mm		2.50	1.92	75.2
Image 5	1.26			
8 mm		1.91	1.85	74.5
12 mm		2.20	1.67	82.9
16 mm		3.17	2.30	86.5

The interactively segmented slices were reduced to data sets with slice distances of 8, 12 and 16 mm and the omitted intermediate slices were automatically calculated with shape-based interpolation (SBI) and, additionally, with the cost-based optimization described above (OPT).

We defined the user-dependent reference volumes as the results of the interactive segmentation on all slices. The differences between two reference volumes (inter-user variability) or between an interpolated volume and its reference vol-

ume were calculated with the formula

$$1 - \frac{2|V_i \cap V_j|}{|V_i| + |V_j|}, \qquad i \neq j,$$

which is equal to 1 minus the similarity index [16]. These relative volume differences were averaged over the three users and listed as inter-user variability, respectively volume deviations for the automated interpolations in Table 1.

The results for the sequences I1-I4 show that we can save more than 50% of interaction time with the shape-based interpolation alone if we tolerate deviations in the order of the inter-user variability. Moreover, we can use the optimization method (OPT) for a further reduction of the differences (up to one half) or for further savings of user interaction time.

With an optimized interpolation (SBI + OPT) between all 12 mm slices we have comparable deviations to the SBI method for 8mm slices, but interaction time savings of approximately 66% instead of 50%. The interpolation on the 16 mm distant slices lead to deviations of 2.5-5.7%. With automated optimization these results can be improved, but the inter-user variability can only be reached with manual corrections which require extra interaction time.

With the biphasic image sequence I5 we achieved similar results to those of I1-I4, but due to the smaller slice distance higher savings of interaction time. The interpolation leads to higher deviations compared to those of the other four images, which is also caused by more than twice the number of intermediate slices. Here we have for example a volume deviation of 1.67% and interaction time savings of 82.9% with the shape-based interpolation and automated optimization on 12 mm distant slices (set of every sixth slice) compared to the interactive segmentation on all slices.

If the user verifies the automatically generated contours, the supported manual optimization can be used for further improvements. Corrections of the SBI + OPT results on few slices where the automatically determined lowest cost paths considerably deviate from the object boundary (Figure 5, right), usually lead to a significant reduction of the volume differences.
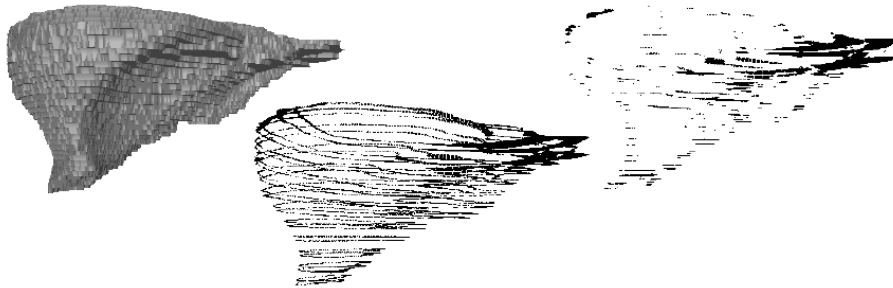

Fig. 5. Surface rendering of the user-segmented volume in image I1 (left), deviations between this volume and the 12mm SBI volume (middle), deviations for 12mm SBI + OPT (right).

4 Conclusions

In this paper we presented a new approach for the semiautomatic segmentation of three-dimensional objects in medical images. The combination of the live wire algorithm and shape-based interpolation builds an effective and accurate segmentation tool which we extended by an automated optimization method.

The live wire algorithm provides direct control to the user, is easy to use and offers high flexibility regarding image modalities and segmentation objects. For the segmentation of three-dimensional objects we combined this slice-based method with shape-based interpolation, leading to a significant reduction of interaction time compared to the segmentation on all slices. The applicability of this combination depends on the properties of the object contours, it is less appropriate for strongly undulated boundaries (e.g. white/gray matter in the brain), but a good choice for structures with a more regular surface (e.g. liver, lung). The shape-based interpolation is especially suited for image sequences with a large number of image slices like multi-slice CTs.

We showed in experiments with five hepatic CTs that we can considerably improve the interpolation results or decrease – with constant level of accuracy – the number of user-segmented slices with an automatic optimization method based on the ideas of the live wire algorithm. In the validation we have used equal distances for the interactively segmented images, but we expect even better results with slices individually selected by the experienced user who rates the boundary changes during the segmentation process and specifies the images for contour interpolation.

While the shape-based interpolation and the path-finding algorithm can be calculated efficiently in a few seconds for all interpolated slices, the automated optimization requires more computational time primarily due to the calculation of the cost graph on every interpolated image. Therefore one focus in our current work is the local restriction of cost computation.

Further improvements regarding the accuracy of the interpolated contours can be expected from a better adjustment of the approximated seed points to the underlying image data.

Acknowledgments

References

1. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision* 1(4), pp. 321–331, 1988.

2. D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence* 36(1), pp. 91–123, 1988.

3. A. X. Falcao and J. K. Udupa. Segmentation of 3D objects using live-wire. In *SPIE on Medical Imaging*, vol. 3034, pp. 191–198, Newport Beach, CA. 1997

4. E. N. Mortensen, B. S. Morse, W. A. Barrett, and J. K. Udupa. Adaptive boundary detection using live-wire two-dimensional dynamic programming. In *IEEE Computers in Cardiology*, pp. 635–638, Durham, North Carolina, IEEE Computer Society Press, 1992.

5. J. K. Udupa, S. Samarasekera, and W. A. Barrett. Boundary detection via dynamic programming. In *Visualization in Biomedical Computing '92*, Chapel Hill, North Carolina, pp. 33–39, 1992.

6. E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *Computer Graphics (SIGGRAPH '95)*, Los Angeles, CA, pp. 191–198, 1995.

7. D. Stalling and H.-C. Hege. Intelligent scissors for medical image segmentation. In *Digitale Bildverarbeitung fuer die Medizin*, Freiburg, Germany, pp. 32–36, 1996.

8. W. A. Barrett and E. N. Mortensen. Fast, accurate and reproducible live-wire boundary extraction. In *Visualization in Biomedical Computing*, Hamburg, Germany, pp. 183–192, 1996.

9. W. A. Barrett and E. N. Mortensen. Interactive live-wire boundary extraction. *Medical Image Analysis* 1(4), pp. 331–341, 1997.

10. A. X. Falcao, J. K. Udupa, S. Samarasekera, S. Sharma, B. E. Hirsch, and R. A. Lotufo. User-steered image segmentation paradigms: Live-wire and live-lane. *Graphical Models and Image Processing* 60(4), pp. 223–260, 1998.

11. A. X. Falcao, K. Jayaram, J. K. Udupa, and Miyazawa F. K. An ultra-fast user-steered image segmentation paradigm: Live-wire-on-the-fly. In *SPIE on Medical Imaging*, vol. 3661, Newport Beach, CA, pp. 184–191, 1999.

12. T. M. Lehmann, C. Goenner, and K. Spitzer. Survey: Interpolation methods in medical image processing. *IEEE Transactions on Medical Imaging* 18(11), pp. 1049–1075, 1999.

13. S. P. Raya and J. K. Udupa. Shape-based interpolation of multidimensional objects. *IEEE Transactions on Medical Imaging* 9(1), pp. 32–42, 1990.

14. G. T. Herman, J. Zheng, and Bucholtz C. A. Shape-based interpolation. *IEEE Computer Graphics and Applications* 12(3), pp. 69–79, 1992.

15. G. Borgefors. Distance Transformations in Arbitrary Directions. *Computer Vision, Graphics and Image Processing* 27(3), pp. 321–345, 1984.

16. A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer. Morphometric Analysis of White Matter Lesions in MR Images: Method and Validation. *IEEE Transactions on Medical imaging* 13(4), pp. 716–724, 1994.

17. D. Selle, T. Schindewolf, C. J. G. Evertsz, and H.-O. Peitgen. Quantitative analysis of CT liver images. In *Computer-Aided Diagnosis in Medical Imaging*, Chicago, pp. 435–444, Elsevier, 1999.

18. B. Preim, D. Selle, W. Spindler, K. J. Oldhafer, H.-O. Peitgen. Interaction Techniques and Vessel Analysis for Preoperative Planning in Liver Surgery. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2000* (this volume).