

Interactive 3D editing tools for image segmentation [☆]

Yan Kang, Klaus Engelke ^{*}, Willi A. Kalender

Institute of Medical Physics, University of Erlangen-Nürnberg, Krauenhausstr. 12, Erlangen 91054, Germany

Received 23 October 2002; received in revised form 11 June 2003; accepted 3 July 2003

Abstract

Segmentation is an important part of image processing, which often has a large impact on quantitative image analysis results. Fully automated operator independent segmentation procedures that successfully work in a population with a larger biological variation are extremely difficult to design and usually some kind of operator intervention is required, at least in pathological cases.

We developed a variety of 3D editing tools that can be used to correct or improve results of initial automatic segmentation procedures. Specifically we will discuss and show examples for three types of editing tools that we termed: hole-filling (tool 1), point-bridging (tool 2), and surface-dragging (tool 3). Each tool comes in a number of flavors, all of which are implemented in a truly 3D manner. We describe the principles, evaluate efficiency and flexibility, and discuss advantages and disadvantages of each tool. We further demonstrate the superiority of the 3D approach over the time-consuming slice-by-slice editing of 3D datasets, which is still widely used in medical image processing today. We conclude that performance criteria for automatic segmentation algorithms may be eased significantly by including 3D editing tools early in the design process.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Three-dimensional image segmentation; Three-dimensional editing tools; Hole-filling; Point-bridging; Surface-dragging

1. Introduction

Segmentation is an integral part of image processing. Contrary to many technical applications the design of fully automated segmentation routines is extremely challenging in the medical context because of the large biological variation. Even if automatic routines do work in normal subjects, they typically fail in pathologic cases, which are often more interesting from a clinical point of view. In our experience operator interaction should always be possible although it may be used in a minority of cases, only. The design goal of full automation is not realistic. On the contrary, incorporating user interactions early in the design process may considerably facilitate the development of segmentation algorithms and guarantee lower failure rates.

On the opposite side, user interaction introduces a subjective element to image processing and analysis. Thus, in order to achieve high accuracy and precision a high degree of automation is desirable. This does not invalidate the arguments discussed in the previous paragraph, but necessitates a careful design of user interaction. A user-friendly implementation of editing tools must combine intuitive and easy handling with acceptable performance. Slice-by-slice editing of 3D datasets does not meet these criteria; instead, 3D editing tools should be preferred. This is the topic of the work presented here.

Two principle modes of user interaction can be distinguished. In the first mode the user interactively selects a region or volume of interest (ROI or VOI) in which subsequently an automated operation is performed. A typical well-known example is the selection of a seed point to start a region growing process. In contrast, the second mode is iterative and requires extended user interaction, for example, an interactive change of a contour. This mode requires real-time visualization, which is demanding if a large amount of data is affected, such as in 3D datasets.

[☆] Supplementary data associated with this article can be found at doi: 10.1016/j.media.2003.07.002.

^{*} Corresponding author. Tel.: +49-9131-8522829; fax: +49-9131-8522824.

E-mail address: klaus.engelke@imp.uni-erlangen.de (K. Engelke).

Apart from interactive segmentation procedures that use a manual placement of a seed point to start a volume growing process (Adams and Bischof, 1994), the literature on editing tools in general and on 3D tools in particular is sparse (Montanat et al., 2001; Olabarriaga and Smeulders, 2001). One reason may be that scientific publications on segmentation mostly emphasize the automatic part while necessary manual corrections are considered as flaws in the automated process that could probably be avoided by an improved description of the underlying theory. In practice many image processing programs do offer user interaction tools but most of them are slice-based, for example, they let the operator manipulate 2D contours (Barrett and Mortensen, 1997; Einstein et al., 1997; Lockett et al., 1998; Mortensen and Barrett, 1998; Wu and Len Yencharis, 1998).

In this paper we present several 3D editing tools for interactive corrections of segmentation results that are independent of the specific segmentation algorithm. Tools for both modes discussed above were developed. We will, in particular, stress the importance of a fully 3D approach compared to 2D slice-based methods. The

tools were developed within an ongoing project on segmentation of CT datasets of the proximal femur that is described in more detail in a separate publication (Kang et al., 2003). Therefore, the applications of the interactive tools discussed in this paper are exemplified using the femur but the tools are general in nature and can be used for other anatomical sites as well. Automatic segmentation of bone often fails if either the cortical thickness or the bone mineral density or both are low. As a consequence artificial elements that look like holes or cavities in mountains are generated in the surface as shown in Fig. 1. This is the background for the development of our interactive editing tools.

2. Three-dimensional editing tools

All editing tools developed in the context of this work are 3D tools, although in the following, results will often be demonstrated using orthogonal multi-planar reformations (MPRs) or even single 2D slices. Two-dimensional approaches applied to 3D datasets such as



Fig. 1. Typical problems after an automated 3D segmentation of bone structures. 'Holes' in the cortical shell and 'cavities' in the trabecular bone are frequently generated. In the example a volume growing algorithm with adaptive local thresholds and a subsequent morphological closing operator was used for automatic segmentation (Kang et al., 2003).

a slice-by-slice segmentation may give satisfactory results in single planes, e.g. in the transversal direction. But when viewed from an orthogonal direction often staircase artifacts are visible. This is also true when using interactive editing tools. 3D editing tools therefore appear indispensable.

Altogether we will present three different tools. The implementations of tool 1 belong to the first category: after selecting a VOI to limit their spatial extensions they proceed either automatically or after initialization by a seed point. Thus, for tool 1 user interaction is limited to a minimum. Tools 2 and 3, which belong to the second category and require real-time updates after each user interaction, will be introduced in Section 2.2.

2.1. Editing tools that proceed automatically or with seed point initialization after interactively selecting a VOI

2.1.1. Selection of VOIs

In our case we use spherical VOIs although this does not limit generality. In the MPR view the sphere is visualized by circles that result if the sphere is cut by the corresponding MPR planes (Fig. 1). The operator can select a circle with the mouse on any of the three MPRs and move or resize it. The other two MPRs are instantly updated after each mouse movement. Of course, the usual interactions with the MPRs like zooming, panning, or slicing also update the circles. Thus, the user can position the VOI quickly and exactly.

2.1.2. Tool 1: hole-filing

For tool 1 we developed three different hole-filling approaches. In addition to the interactive location of the holes using a VOI as discussed above, one of the methods requires a seed point the other two proceed automatically.

2.1.2.1. Three-dimensional morphological closing. Three-dimensional tools based on morphological operations have been used for some time (Höhne and Hanson,

1992). Morphological closing is one of the easiest methods that can be applied within the selected VOI to close holes. The success depends on the hole size. Large holes require large structuring elements that also have a considerable and often unwanted smoothing effect as illustrated in Fig. 2.

Another unwanted effect of large structuring elements is the increase in computation time. This is in particular a limit when using 3D editing tools. Therefore, we developed a two-step procedure to considerably speed up morphological operations:

1. The morphological operations are only applied to the voxels that are located on a bone surface.
2. For spherical structuring elements only the shells, not the solid spheres are used.

Table 1 shows the relative increase in computation time with increasing sizes of the structuring elements for the example given in Fig. 2. The computation time for a solid sphere with a radius of 6 mm was 81 times higher compared to a sphere with a radius of 1 mm. Using only the outer shell (shell thickness 0.6 mm) as structuring element, the computation time could be significantly reduced, up to a factor of 4 for a radius of 6 mm.

If morphological closing and opening operations are applied locally, e.g. within our VOI, typically significant

Table 1

Relative computation times of a morphological closing for spherical structuring elements normalized to a solid element with a radius of 1 mm

Radius of structure element	1 mm	2 mm	3 mm	6 mm
Normal	1	3.76	10.57	81.10
Accelerated	0.95	2.71	5.33	20.76

In the 'normal' mode, a solid sphere is used, in the accelerated mode only the shell of the sphere is used as structuring element (shell thickness 0.6 mm, corresponding to two voxels in the x and y directions and to 1 voxel in the z direction in the digital datasets). The voxel sizes in our anisotropic datasets were $0.3 \times 0.3 \times 1.0 \text{ mm}^3$. Obviously the larger the radius the larger the savings in the accelerated mode, up to a factor of 4 for a radius of 6 mm.

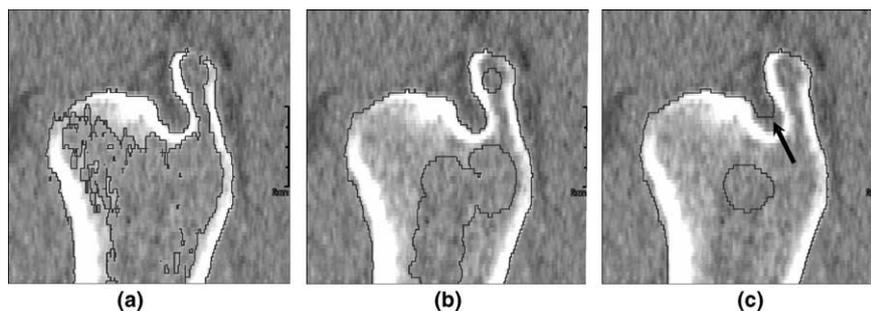


Fig. 2. 3D morphological closing with spherical structuring elements (SEs) of different sizes: (a) result after automatic segmentation; (b) 3D closing with SE radius of 2 mm; (c) 3D closing with SE radius of 4 mm. The voxel size in the dataset is $0.3 \times 0.3 \times 1.0 \text{ mm}^3$. An SE of 2 mm here appears to offer the optimum performance, as it does not introduce significant new errors as indicated by the arrow on (c). However considerable holes remain in (b).

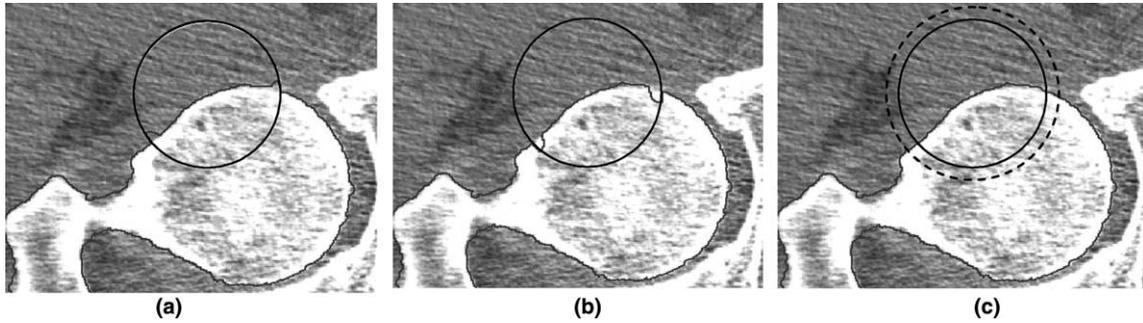


Fig. 3. Edge artifacts of local 3D closing and opening operations: (a) closing; (b) opening; (c) no artifacts using our improved closing procedure (see text). The dashed VOI is shown here for demonstration purposes, only. The morphological operations are calculated inside this dashed VOI but are applied to the user defined VOI represented by the solid circle, only.

edge effects as illustrated in Fig. 3 occur because the basic operations of dilation and erosion are not strictly inverse procedures. We solved this problem by calculating the morphological operations inside a VOI that was slightly larger than the user-defined VOI. However, the results are only applied inside the user-defined VOI so that no boundary discontinuities occur (Fig. 3(c)). The larger VOI (dashed circle) is shown in Fig. 3(c) for demonstration purposes only.

2.1.2.2. Three-dimensional morphological weaving. With normal morphological closing, holes larger than $2R$, where R denotes the radius of the structuring element, cannot be filled. Therefore we developed a new algorithm to fill holes larger than $2R$. We termed it “morphological weaving” according to its mechanism. As structuring element, a line of length L is used. When we apply a closing operation with this structuring element we only generate an effect in the direction along the line.

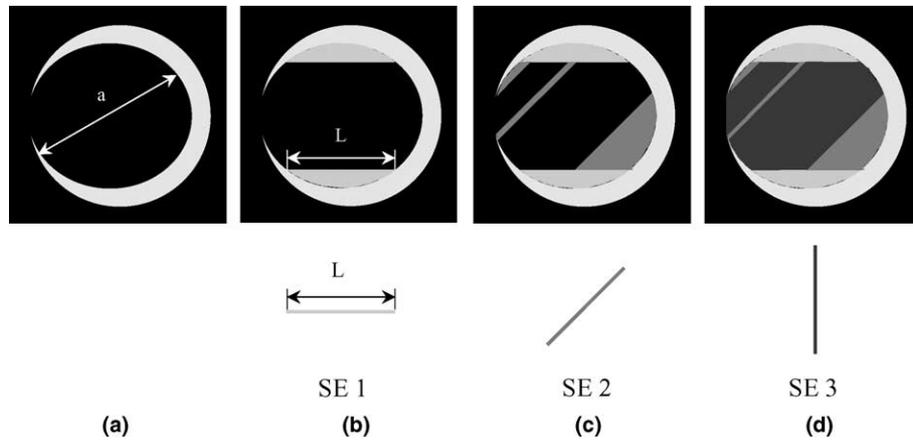


Fig. 4. 2D scheme of 3D morphological weaving: (a) hole with size a to be corrected; (b) weaving using line element of length L ($L < a$) in horizontal direction; (c) and (d) weaving using line elements in two additional directions. In the simple case illustrated, no further directions are required. In 3D space 13 directions are used.

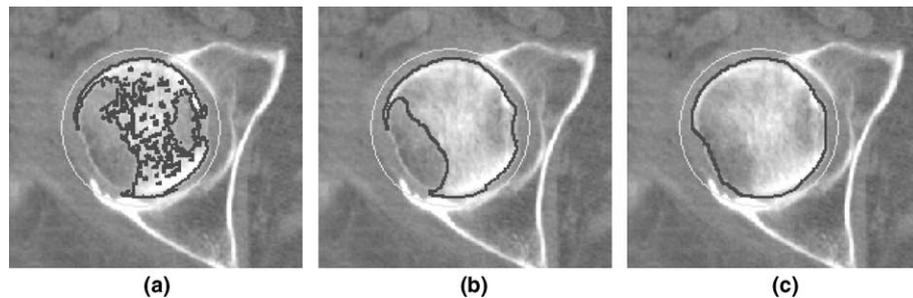


Fig. 5. Morphological weaving compared to morphological closing: (a) automatic volume growing using local thresholds; (b) 3D closing with spherical structuring element (radius = 4 mm); (c) 3D weaving with line structuring element ($L = 8$ mm).

However, by repeating the closing operation with several line-structuring elements of length $L = 2R$ rotated by different angles we can fill holes larger than $2R$ (Fig. 4). We actually use 13 different directions in 3D space that account for all voxels in a 26-neighbor mode. Fig. 5 illustrates the power of 3D weaving for filling large holes compared to simple morphological closing.

2.1.2.3. Three-dimensional ray reflection model. This approach for tool 1 can be easily understood if you imagine being in a cavity holding a candle. You can judge whether your current position is inside or outside the cavity by analyzing the light reflections from the walls around you. We assume an isotropic light source and count the reflected rays from the wall representing the segmented structure (Fig. 6). If for a given voxel the amount of reflected rays is larger than 65% of those sent out, then the voxel is inside a hole otherwise it is outside. We used 26 directions in the digitized 3D image space. The 65% criterion was selected empirically.

The actual algorithm uses the well-known volume-growing algorithm and replaces the normal global threshold criterion with the reflection condition described above. Thus, each voxel within the selected VOI can be classified as inside or outside of a hole. Results are shown in Fig. 7.

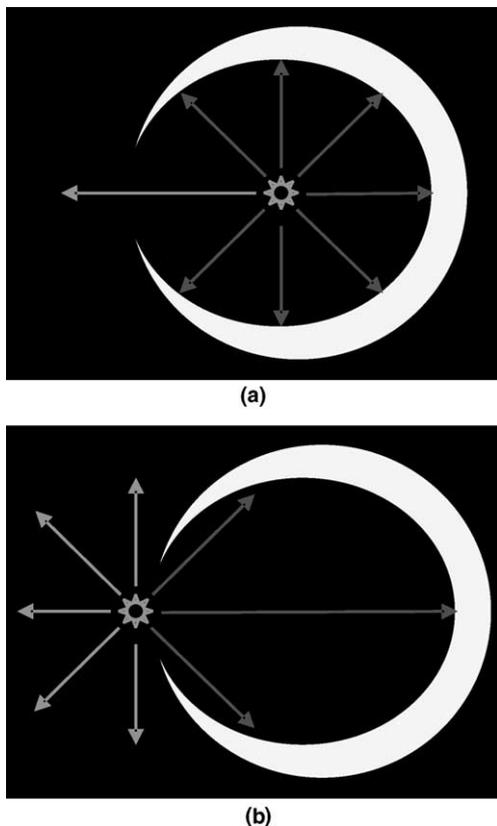


Fig. 6. 2D scheme of 3D ray reflection model: (a) 'inside' condition: reflected rays $\geq 65\%$; (b) 'outside' condition: reflected rays $< 65\%$.

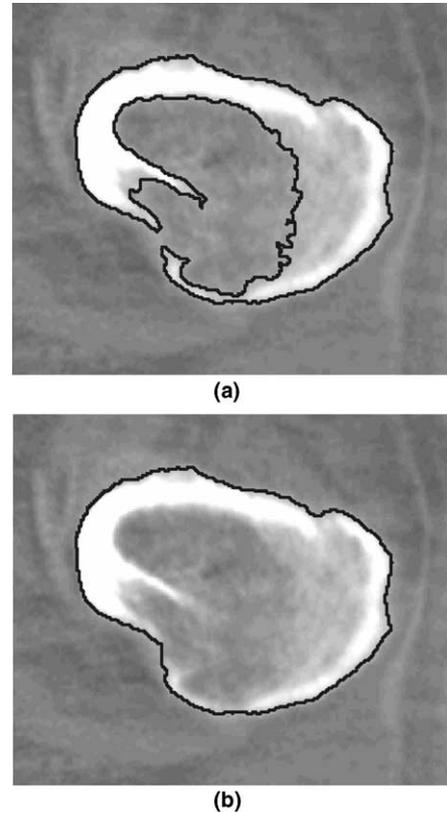


Fig. 7. Three-dimensional ray reflection applied to a realistic example.

2.1.2.4. Comparison of hole correction methods. The specific advantages and disadvantages of each of the three 3D hole-filling approaches are summarized in the upper part of Table 2. 3D morphological closing is not suited to fill large holes. For this purpose 3D morphological weaving can be used but it is relatively slow because for a given voxel 13 directions of the line-structuring element have to be calculated. Both morphological methods may suffer from the loss of detail information.

The 3D ray reflection model needs an additional seed point as input and it must be applied carefully if the segmented object has a complex shape, which may result in over-filled holes. However, it is advantageous if a large cavity with a relatively small hole in the surface needs to be filled. As can be seen from the results all three implementations of tool 1 have difficulties correcting very large boundary defects. Also at this point none of the methods described so far makes use of the CT values falsely classified as soft tissue. This is the purpose of the 3D editing tools described in the following section.

2.2. Editing tools that require real-time updates after each user interaction

The tools introduced in this section also work within a local VOI interactively selected by the user. However,

Table 2
Summary of advantages and disadvantages of the 3D editing tools presented in this contribution

Tool	Complexity	Advantages	Disadvantages	Preferred usage
Tool 1: hole-filling	low	<ul style="list-style-type: none"> can easily be integrated in automatic algorithms 	<ul style="list-style-type: none"> cannot fill holes $> 2R^*$ surface details are lost for large structuring elements 	<ul style="list-style-type: none"> filling of multiple small holes
Morphological closing	medium	<ul style="list-style-type: none"> can fill larger holes than morphological closing 	<ul style="list-style-type: none"> slower than morphological closing 	<ul style="list-style-type: none"> filling of large holes
Morphological weaving	medium	<ul style="list-style-type: none"> fast no loss of details. 	<ul style="list-style-type: none"> seed point required fails if multiple holes are present 	<ul style="list-style-type: none"> filling of small holes within large cavities
3D ray reflection	medium	<ul style="list-style-type: none"> uses gray values in VOI 	<ul style="list-style-type: none"> Threshold must be selected manually 	<ul style="list-style-type: none"> complex defects
Thresholding mode	medium	<ul style="list-style-type: none"> highly flexible 	<ul style="list-style-type: none"> t tedious to use 	<ul style="list-style-type: none"> any defect
Manual mode	low	<ul style="list-style-type: none"> preserves surface shape after dragging 	<ul style="list-style-type: none"> VOI must be selected carefully 	<ul style="list-style-type: none"> minor corrections of local surfaces
Force mode	high	<ul style="list-style-type: none"> independent of original surface shape 	<ul style="list-style-type: none"> VOI must be selected carefully 	<ul style="list-style-type: none"> corrections of local surfaces
Tool 2: point-bridging	high			
Tool 3: surface-dragging	high			
Spline mode	high			

Complexity evaluates the algorithmic efforts to implement a given tool. Advantages and disadvantages evaluate a tool from the perspective of the user. * R , radius of structuring element.

the ongoing user input requires a quick visual feedback. Thus, in order to work conveniently updates must (almost) occur in real time.

2.2.1. Tool 2: 3D point-bridging

The basic idea is as follows. (1) Identify a few voxels that are falsely classified as soft tissue and label them as bone. (2) Connect these “bridge” points with the other segmented voxels using a 3D morphological closing operation. As an example Fig. 8(a) shows a segmentation defect and the interactively placed VOI on three orthogonal MPRs. The defect occurred during the automatic segmentation of the proximal femur consisting of a multi-step procedure including volume growing with local adaptive thresholds, morphological operations and surface refinements based on grey value gradients (Kang et al., 2003). Fig. 8(b) indicates the location of the defect in the major trochanter.

For the correction, a histogram of all CT values inside the VOI is calculated and displayed (Fig. 8(c)) to guide the operator in selecting an optimal threshold for the separation of bone and soft tissue. The segmentation results within the VOI are updated and displayed in all three MPRs in real time so that the operator can adjust the threshold accordingly. In the case displayed in Fig. 8(d) most of the voxels originally labeled as bone (Fig. 8(a)) by the automatic segmentation procedure actually are below the selected threshold. However, a few points originally labeled as soft tissue are above the selected threshold. These are used as bridge points (Fig. 8(d)). Their small number suffices for a subsequent automatic 3D closing operation (spherical structuring element $R = 1.8$ mm) within the VOI to give the results in Fig. 8(e). Fig. 8(d) also shows the structuring element.

A real-time video demonstrates the interactive application of these tools along with a 3D ray reflection operation to fill the remaining hole. It can be found in the electronic annexes (doi: 10.1016/j.media.2003.07.002). Alternatively this tool can be used without the histogram. In this mode the user has to manually place the bridge points, afterwards the tool proceeds as above.

2.2.2. Tool 3: 3D surface-dragging

As a potential alternative to point-bridging, we further developed an interactive tool that we termed 3D surface-dragging. The surface of the segmented structure that is inside the interactively placed VOI can be dragged via a so-called control point, which is repositioned interactively. In our case the control point is the center of the sphere. The changed surface is calculated and displayed in real time as contours in the three MPR views. Again, for tool 3, we implemented two different approaches, a force and a spline mode.

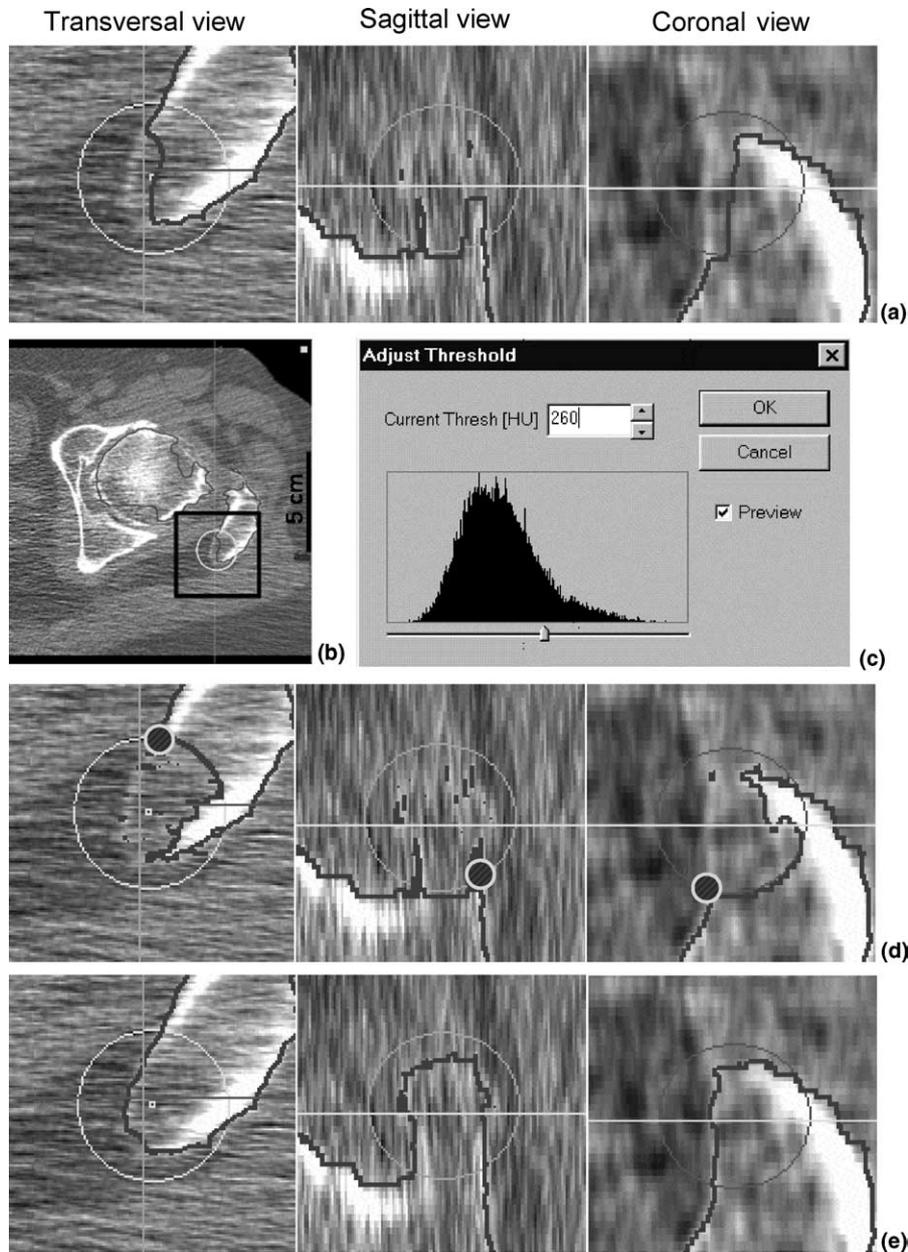


Fig. 8. 3D point-bridging: (a) original segmentation and VOI in which local correction is carried out; (b) overview indicating the location of the automatic segmentation insufficiency in the trochanter major; (c) histogram of the CT values inside the VOI; (d) morphological structuring element and bridge points (dark dots in the VOI) generated by interactive operation (see text). Note that some of these points were already detected by the automatic segmentation (see a); (e) final segmentation after morphological closing.

2.2.2.1. Force mode. The force mode is based on the concept that all surface points inside the spherical VOI are displaced by a force field. The direction of the force is parallel to the displacement vector of the control point, the position of which is interactively changed by the operator. The strength of the force is a function of the distance d of a given surface point from the control point. Obviously d varies between 0 and R , where R is the radius of the VOI (Fig. 9(a)). The strength of the force has been modeled by three different functions f :

- $f_1 = 1 - x$: linear functionality;
 - $f_2 = e^{-kx}$: exponential functionality;
 - $f_3 = e^{-kx^2}$: Gaussian functionality;
- where $x = d/R \in [0, 1]$ (Fig. 9(b)).

In practice, after shifting all selected surface points a morphological closing is used to fill gaps created by the dragging process.

2.2.2.2. Spline mode. In spline mode we used multiple cubic B-splines to perform a dragging operation on the segmented surface inside the VOI. In order to easier

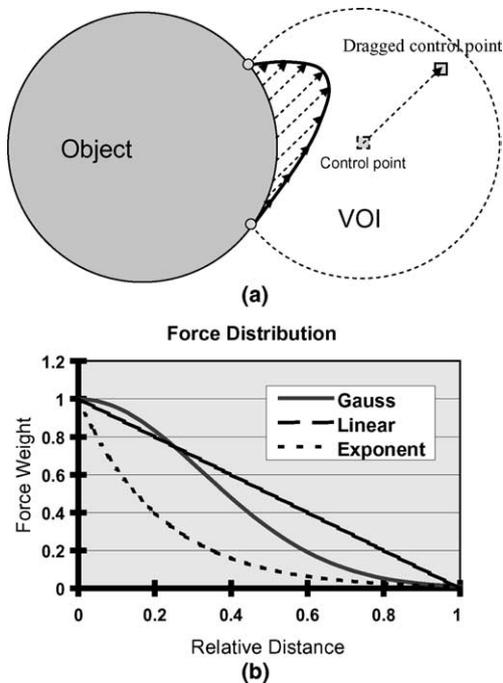


Fig. 9. Surface-dragging in force mode: (a) illustration of principle; (b) functions to model force dependence.

understand the mechanism of this tool we will first explain the 2D case.

Two-dimensional contour dragging using splines. A cubic B-spline can be described algebraically by a Bernstein polynomial of degree 3 which is called Bezier cubics (Farin, 1990):

$$z(t) = (1-t)^3 z_0 + 3(1-t)^2 t z_1 + 3(1-t)t^2 z_2 + t^3 z_3,$$

where t varies between 0 and 1. Fig. 10(a) shows a typical B-spline. From the equation above, it is evident that four points z_0, z_1, z_2 and z_3 are needed to determine

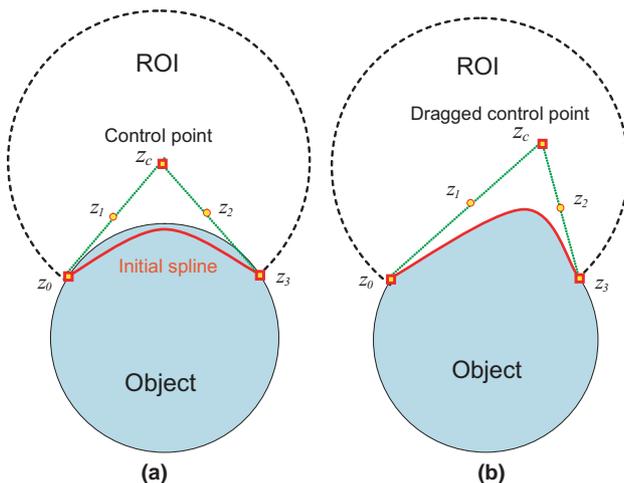


Fig. 10. 2D surface-dragging in spline mode. For simplicity we assume the contour to be manipulated to be circular: (a) before the dragging operation: original contour and initial 2D spline; (b) after the dragging operation: the contour inside the VOI is identical to the interactively changed 2D spline.

the shape of the spline curve. In our case the two end points z_0 and z_3 by definition are fixed because these are the cut points of the spline with the border of the ROI. Still we are left with two undefined points, which is inconvenient because it is difficult to interactively operate on two control points simultaneously.

In order to simplify the interactive operation, we combined the two non-end points z_1 and z_2 into a single control point z_c (Fig. 10(a)) by defining z_1 and z_2 to be the midpoints of the lines $z_0 - z_c$ and $z_3 - z_c$, respectively. Thus, once we have selected the position of z_c we can calculate the positions of z_1 and z_2 using the positions of the two fixed end-points z_0 and z_3 . This procedure defines a spline, which obviously is not identical with the initial contour within the VOI. But this does not matter as we are interested in the final result. We simply define the ‘dragged’ contour to be identical to the recalculated spline after the control point is moved. This procedure has the advantage that an exact knowledge of the potentially complicated initial contour is not required and the dragging implementation is straightforward. As control point we use the center of the ROI (Fig. 10(b)).

Extension to 3D. The equation above describing the Bezier curve is not restricted to a 2D plane. Actually, it works in 3D or higher-dimensional spaces as well. This opens a possibility to implement a surface-dragging mechanism that again is controlled by a single point. In analogy to the 2D case, we define an initial surface by multiple spline curves that all share the same control point (Fig. 11). Actually a 3D surface is constructed by connecting the voxels representing the spline curves with a morphological closing operation. After the first move of the control point we again define the ‘dragged’

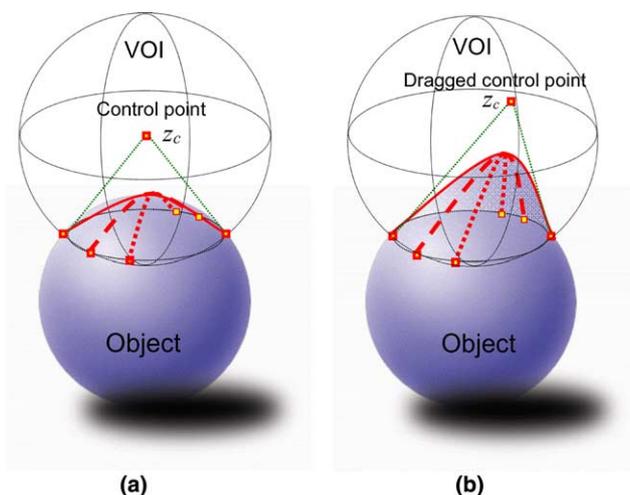


Fig. 11. 3D surface-dragging in spline mode. For simplicity we assume the surface to be manipulated to be spherical. The VOI is also a sphere but is indicated as a circle (a) before the dragging operation: original surface and multiple initial spline curves; (b) after the dragging operation: the surface inside the VOI is calculated from the interactively changed splines.

segmented surface to be identical to the updated surface calculated from the splines. Fig. 12 gives two screen shots before and after the operation of the 3D surface-dragging. A real-time video demonstrates the interactive application of these tools. It can be found in the electronic annexes (doi: 10.1016/j.media.2003.07.002).

Comparison of the dragging methods. The force mode preserves the shape of the surface as all of its points are displaced in a parallel direction. The spline mode does not show this behavior as the individual splines contributing to the surface are differently affected by the

move of the control point. Therefore, this mode is better suited to more complicated defects such as the one shown in Fig. 12 that could not be corrected using the force mode. In some situations such as the one shown in Fig. 8(a), dragging in spline mode will fail. Here tool 2, point-bridging, must be used (see also Table 2).

3. Quantitative evaluation of the 3D editing tools

We also evaluated the usability and suitability of the tools introduced in the previous section quantitatively. We started with a low noise spiral CT dataset of the proximal femur that was successfully segmented (see Fig. 13(a)) using the segmentation approach mentioned above including minor editing of the boundary in the region of the head. This segmented volume we used as the gold standard in the following analysis. We decreased the image quality by simulating a decreased radiation exposure using the same dataset. This was achieved by adding Gaussian noise to the CT projections prior to the tomographic reconstruction. The resulting images along with the results of the automatic segmentation are shown in Figs. 13(b) and (c).

As expected the automatic segmentation showed increasing limitations with increasing noise. We let two different operators (one of them more and the other less experienced in handling the tools) correct the automatic segmentation. They were free to use any combination of tools but were encouraged to vary their combinations. In addition to the tools described above the operators could place a spherical VOI and label all voxels inside as bone (VOI filling) or as soft tissue (VOI empty), they could perform an opening or closing of the whole segmented volume (global closing and opening) and they could undo the last step. As an example, Fig. 13(d) shows the final result of an interactive correction using the data shown in Fig. 13(c).

Each operator corrected both noise enhanced datasets three times with at least 1 h pause between the repetitions. We then carried out the following analyses: we monitored (1) what tools were used and (2) recorded the total correction time for each dataset. (3) We quantitatively compared each corrected segmentation with the gold standard by counting the disjunct voxels between the two segmented volumes. In Fig. 14 these voxels are marked in black for two examples. We (4) finally carried out inter and intra-operator comparisons. As shown in Table 3, there was a large variation in the number and types of tools used by the two operators. Correction times varied between 2 and 15 min. After changing his strategy even the second operator was able to correct the very noisy dataset in less than 10 min (third correction).

Despite the variety of tools applied the average difference between the interactively corrected segmentation

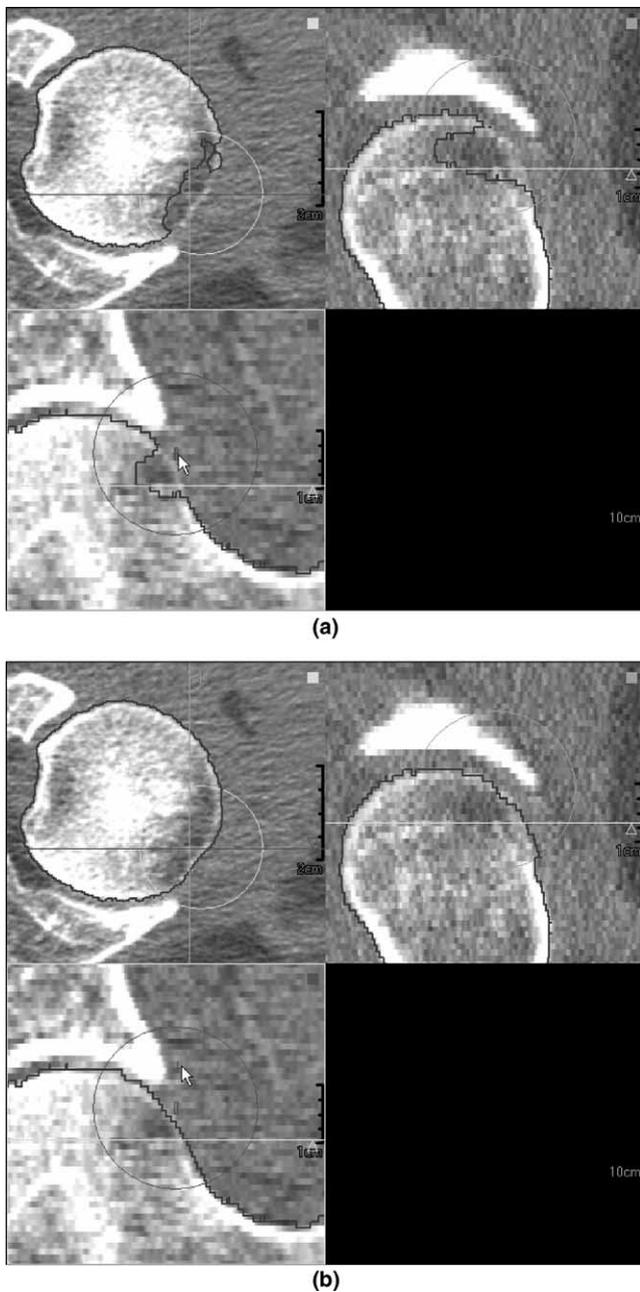


Fig. 12. Screen shots of 3D surface-dragging applied in a realistic situation: results (a) before; (b) after the interaction.

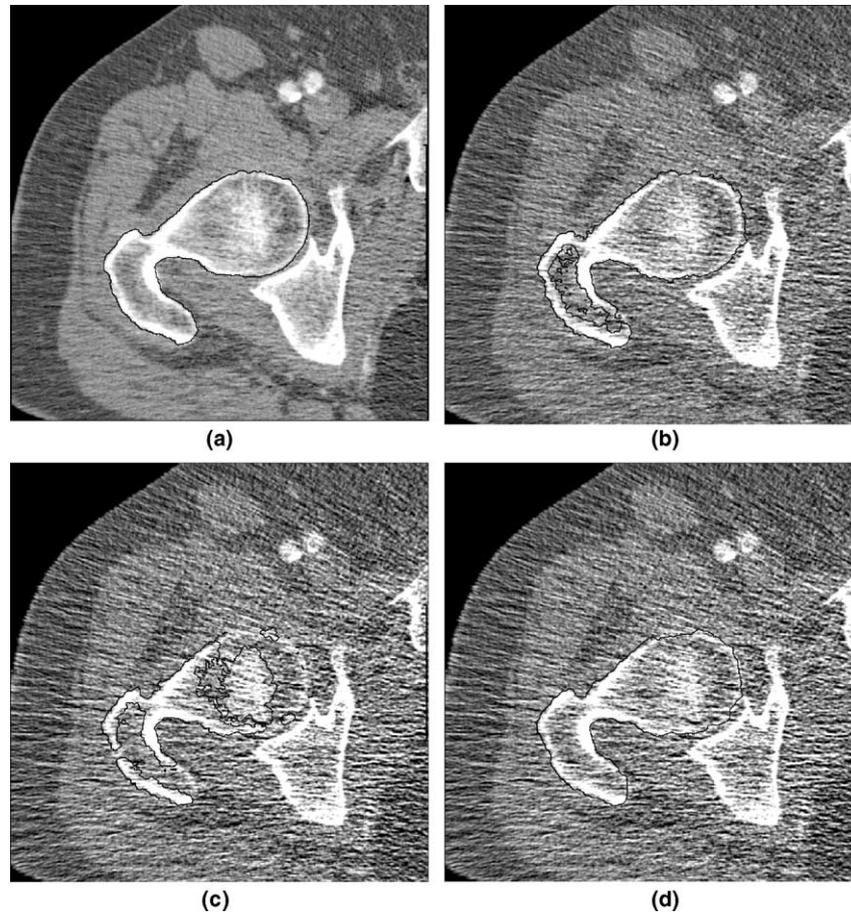


Fig. 13. Transaxial MPRs of the CT dataset used for the evaluation of the 3D editing tools. (a) Original low noise dataset segmented fully automatically; (b) simulated noise increase by a factor of 2 and (c) by a factor of 3. The noise increase was achieved by adding Gaussian noise to the CT projections prior to the tomographic reconstruction. The automatic segmentation results are also shown in (b) and (c); (d) one example of the interactive segmentation correction applied to the data shown in (c).

and the gold standard was smaller than 4%. Intra- and inter-operator differences were even smaller than 2%. Details are listed in Table 4.

4. Discussion

We developed three interactive editing tools that can be used for manual corrections of inadequate automatic segmentation results: hole-filling (tool 1), point-bridging (tool 2) and surface-dragging (tool 3). For each of the tools two or more approaches have been presented. All three tools are applied locally within a manually placed VOI for which also a user-friendly implementation was presented. In our case the use of spherical VOIs was convenient but does not pose any limitation to the results as other shapes could be used equivalently.

Although we use volumetric datasets and volumetric tools we prefer the MPR representation to 3D views such as impressive-looking volume rendered images. For a quantitative analysis the exact position of VOIs and of segmentation results must be controlled and this is easier

to perform on orthogonal MPRs. What is important for 3D image processing is the coupling and joint update of all three MPRs as was demonstrated in our results. Of course depending on the nature of the problem additional oblique MPRs may be very helpful. Potential limitations using the MPR views may exist for the surface dragging tool as in our current implementation the control points can only be moved in one of the three MPR planes but not arbitrarily in 3D space.

The three tools we developed differ in complexity and also have characteristic advantages and disadvantages as summarized in Table 2. The simplest approach of tool 1 just uses a morphological closing inside a VOI. However, even this is effective in situations where many small holes remain after an automated segmentation as shown in Fig. 5. We cannot conclude that the tools with higher complexity replace the simpler ones rather they add complementary functionality. This is also supported by the results in Table 3. All of the tools are easy to use which is most obvious for tool 1 that proceeds automatically after an interactive placement of the VOI. But also tools 2 and 3, which require real-time updates, are

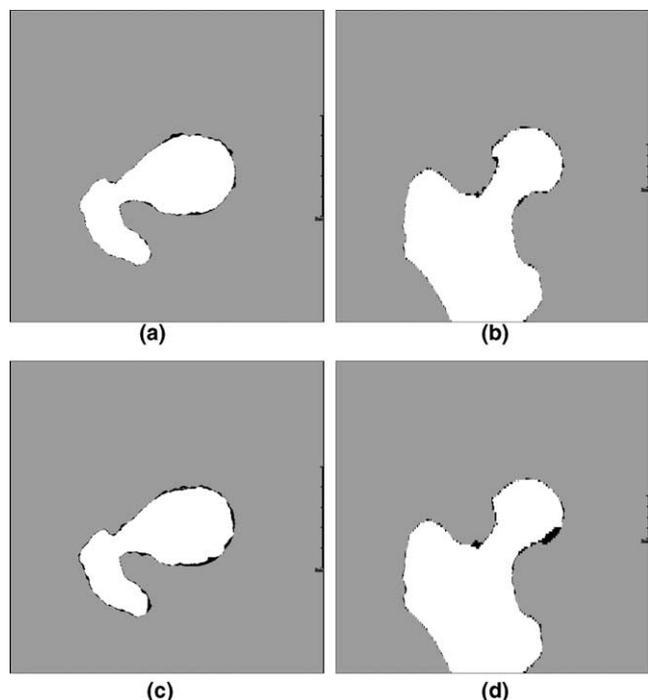


Fig. 14. Discrepancy between the gold standard and two examples of interactively corrected segmentations. The disjunct voxels are shown in black. (a), (b) Noise enhancement by a factor of 2; (c), (d) noise enhancement by a factor of 3. The sagittal view is not shown.

very convenient as can be seen in the two videos accompanying this publication.

Tool 2 is the most flexible tool we developed. However, it has to be applied more or less to each defect

separately, which, as shown in Fig. 1 is different from tool 1 when implemented with morphological closing or weaving. Contrary to tools 1 and 3, tool 2 uses the actual CT values of the voxels affected. So far we worked with CT data, but in principle the tools should be applicable to other modalities such as MR data as well.

All our results demonstrate the power of the 3D approach. This is in particular obvious in Fig. 8. Just a few bridge points as seen on each MPR are generated but the overall result is excellent. It is hard to imagine that a 2D method, for example, applied to the sagittal view, could give equivalent results. Also it must be emphasized that 2D methods should give identical results independent of the view to which they are applied.

Our tools were developed to correct automatic segmentation results of bone but they are independent of a specific segmentation algorithm. They are generic in the sense that they can be applied whenever a surface is a result of a segmentation procedure. Of course, there are numerous other segmentation problems, which require different solutions. For example, our tools are not suitable for manual segmentation; rather a prior automated segmentation is required. It must also be reemphasized that although 3D editing tools should be an integral part of most segmentation schemes in medical imaging, a high degree of automation is necessary in particular to reduce the influence of subjective operator interactions.

We do not claim that we have developed a complete suite of tools; certain areas such as contour fusion or cutting actions were not included. However, at least in our primary target, the proximal femur, which we think

Table 3
Tools used for the interactive correction of the automatic segmentation results in the noise enhanced datasets

Noise enhancement factor	2						3					
	1			2			1			2		
Operator	1	2	3	1	2	3	1	2	3	1	2	3
How often was a tool used?												
Tool 1: hole-filling	Morphological closing				2	1	1	1	3	2		2
	Morphological weaving			2			2			2	1	
	Three-dimensional ray reflection	1	1	1	2	2	1	5	5	7	1	3
Tool 2: point-bridging	Thresholding mode						1	2	2			
	Manual mode				2	1				3	5	1
Tool 3: surface-dragging	Force mode											
	Spline mode	1	1		2				1			1
Further tools not discussed in Section 3	VOI filling								1			1
	VOI empty										1	
	Global morphological opening					1	1	1	1	1		1
	Global morphological closing							1	1	1	1	
Undo last								2	1			
Total number of operations	3	3	4	4	6	4	11	10	16	12	11	7
Time per dataset (min)	2	2	2	4	2	2	7	6	9	15	13	7

Two different operators analyzed each dataset three times. There was no guidance of what and how many tools were to be used.

Table 4
Agreement of segmentation results

Noise enhancement factor	2		3	
	1	2	1	2
Operator				
Difference between interactive and automatic segmentation (%)	2.84 ± 0.02	3.20 ± 0.14	3.83 ± 0.05	3.99 ± 0.31
Intra-operator difference (%)	0.14 ± 0.03	0.42 ± 0.23	0.48 ± 0.08	1.75 ± 0.22
Inter-operator differences (%)	Operator 1 compared to operator 2 0.90 ± 0.23		Operator 1 compared to operator 2 1.21 ± 0.52	

Differences were analyzed as percentage number of voxels that were disjunct between two segmented volumes (see Fig. 14). In the table we show average results given as means ± SD. First, the differences between the interactively segmented volumes and the gold standard are listed. For each operator, results were averaged over the three corrections performed. Then, the intra-operator differences are listed. Here, the three corrections were mutually compared and results again were averaged. Finally inter-operator differences were analyzed by mutually comparing all three segmentation results from operator 1 with the three results from operator 2. Thus, here altogether nine values were averaged.

is representative for other skeletal sites, the variety described here was adequate to cope even with very noisy data and severe failures of the automatic segmentation routines. It is also an important message that the combination of a larger variety of editing tools greatly enhances their individual value.

After the interactive corrections of the noise enhanced images a difference of up to 4% of the voxels compared to the gold standard remained. Obviously, even if an automated segmentation approach were working in the noisy images it would probably also result in differences. Thus, it may be more appropriate to look at intra- and inter-operator differences which were all below 2% of the voxels. This is an excellent result given the fact that in each of the 18 interactive corrections (2 operators, 3 corrections per operator and noise enhancement level) the applied tools varied considerably. We admit that the difference of 2% of the voxels is probably higher if we analyze the effect in smaller VOIs, e.g. the neck region which also is more affected by the manual corrections than the shaft of the femur (see Fig. 14).

Still there is a lot of room for further improvements. For example, we think that 2 min for interactive segmentation corrections are acceptable but 10 min required in the noisier dataset are rather high. Correction times were not limited by computer power but were determined by the user interaction, a standard high end PC, Pentium IV, 2.8 GHz, 1GB RAM was used.

Other improvements would be the implementation of non-spherical local VOIs and the possibility to move the control point used for the 3D surface dragging arbitrarily in 3D space. Also there should be some guidance of which tool should be preferably used in a given situation.

Thus, as a conclusion considerable effort must be applied to develop appropriate 3D segmentation algorithms that minimize necessary operator interaction while simultaneously providing easy to use and intuitive editing tools.

References

- Adams, R., Bischof, L., 1994. Seeded region growing. *IEEE Trans. PAMI* 16 (6), 641–647.
- Barrett, W.A., Mortensen, E.N., 1997. Interactive live-wire boundary extraction. *Med. Image Anal.* 1 (4), 331–341.
- Einstein, A.J., Gil, J., Wallenstein, S., Bodian, C.A., Sanchez, M., Burstein, D.E., Wu, H.S., Liu, Z., 1997. Reproducibility and accuracy of interactive segmentation procedures for image analysis in cytology. *J. Microsc.* 188 (Pt 2), 136–148.
- Farin, G., 1990. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press, New York.
- Höhne, K.H., Hanson, W.A., 1992. Interactive 3D segmentation of MRI and CT volumes using morphological operations. *J. Comput. Assist. Tomogr.* 16 (2), 285–294.
- Kang, Y., Engelke, K., Kalender, W.A., 2003. A new accurate and precise 3D segmentation method for skeletal structures in volumetric CT data. *IEEE Med. Imag. 22* (5), 586–598.
- Lockett, S.J., Sudar, D., Thompson, C.T., Pinkel, D., Gray, J.W., 1998. Efficient, interactive, and three-dimensional segmentation of cell nuclei in thick tissue sections. *Cytometry* 31 (4), 275–286.
- Montanat, J., Delingette, H., Ayache, N., 2001. A review of deformable surfaces: topology, geometry and deformation. *Image Vision Comput.* 19, 1023–1040.
- Mortensen, E.N., Barrett, W.A., 1998. Interactive segmentation with intelligent scissors. *Graph. Models Image Process.* 60, 349–384.
- Olabarriaga, S., Smeulders, A., 2001. Interaction in the segmentation of medical images: a survey. *Med. Image Anal.* 5, 127–142.
- Wu, Y., Len Yencharis, 1998. Commercial 3-D Imaging software migrates to PC medical diagnostics. *Adv. Image Mag.*, 16–21.