Guided Interactive Volume Editing in Medicine

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

MSc. Alexey Karimov
Matrikelnummer 1125245

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

Diese Dissertation haben begutachtet:

_____   _____
Stefan Bruckner   Bernhard Preim

Wien, 2. Mai 2016   _____
                                                           Alexey Karimov

Guided Interactive Volume Editing in Medicine

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

MSc. Alexey Karimov

Registration Number 1125245

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

The dissertation has been reviewed by:

Stefan Bruckner	Bernhard Preim

Vienna, 2nd May, 2016

Alexey Karimov

Erklärung zur Verfassung der Arbeit

MSc. Alexey Karimov
Molkereistrasse 1 /2 /34, 1020 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 2. Mai 2016

_____

Alexey Karimov

Acknowledgements

I AM grateful to Eduard Gröller AKA Meister for supporting my research at the Technische Universität Wien and providing support during my PhD studies. This work would have never happened without a large amount of support, scientific collaboration and friendly advice from Gabriel Mistelbauer, whom I express my gratitude to. I acknowledge Thomas Auzinger for fruitful brain-storming sessions and suggestions, which had a significant impact on this work. I would like to thank the entire Vis-Group and, in particular, Andrej Varchola, Manuela Waldner, Artem Amirkhanov, Johanna Schmidt, Aysylu Gabdulkhakova, Matthias Bernhard, Michael Hecher, Nicholas Waldin, Reinhold Preiner, Peter Mindek, Andreas Weiner, and Stefan Bruckner for numerous scientific (and, sometimes, definitely not scientific) discussions. I thank Anita Mayerhofer-Sebera and Tammy Ringhofer for organizing the chaos at our institute. My special thanks go to Dr. med. Rüdiger Schernthaner, Markus Schreiner, and Dr. Timur Sharipov for their vast medical experience and knowledge, which they are open to share. I would like to thank Gerald Schröcker and Daniel Buckton for an outstanding research collaboration to strive to create breath-taking fetal ultrasound images.

I acknowledge M. Reiter and the research group XCT of the University of Applied Sciences, Upper Austria for providing the TK08 dataset. The medical datasets are courtesy of the SLiver 2007 contest (acknowledging T. Heimann, B. van Ginneken, and M. Styner), the Kaiser-Franz-Josef Hospital of Vienna, Austria, the General Hospital of Vienna, Austria, and the Republican Clinical Hospital of G. G. Kuvatov in Ufa, Russian Federation. The Electron Microscopy biological data is courtesy of the SNEMI3D contest (acknowledging I. Arganda-Carreras, H. S. Seung, A. Vishwanathan, and D. Berger), and its segmentation is courtesy of the Harvard Connectome group of J. Lichtman and H. Pfister.

I thank my family for the great deal of patience and support I got while following this somewhat hard way of being a doctoral researcher.

Kurzfassung

ILDGEBENDE Verfahren wie Computertomographie, Magnetresonanztomographie und Ultraschall sind aus der medizinischen Diagnostik nicht mehr wegzudenken; sie sind mittlerweile für die Diagnose unterschiedlichster Krankheiten unumgänglich geworden. Die Anamnese (der medizinische Diagnoseprozess) mittels bildgebender Verfahren kann durch automatische und interaktive Analysetools in hohem Maße unterstützt werden. Analysetools können für standardisierte Messungen verwendet werden, zum Bewerten von Pathologien, zum Planen von Behandlungen und Operationen, und vielem mehr. Um solche Analysetools bestmöglich einsetzen zu können, müssen seitens der Daten bestimmte Voraussetzungen erfüllt werden. Eine der wichtigsten Anforderungen ist dabei das Vorhandensein einer Segmentierungsmaske jener inneren Organe oder Strukturen, die von den Ärzten näher untersucht werden sollen. Die Segmentierung von medizinischen Daten ist nach wie vor ein fehleranfälliger Prozess. Nicht selten müssen Segmentierungsmasken, obwohl (semi-)automatisch erstellt, von geschultem Personal manuell nachbearbeitet werden, um ein brauchbares Ergebnis zu erzielen. Daraus ergibt sich eine große Nachfrage nach interaktiven, und möglichst intuitiven Techniken zum Bearbeiten von Segmentierungsmasken für medizinische Daten.

In dieser Dissertation werden drei fortgeschrittene Techniken zum Bearbeiten von Segmentierungsmasken vorgestellt. Besonderes Augenmerk wurde dabei auf die Wahl einfacher Interaktionsmöglichkeiten gelegt, mit Hilfe derer in relativ kurzer Zeit ein gutes Segmentierungsergebnis erzielt werden kann. Alle drei Techniken haben gemein, dass sie auf einer Repräsentation aufbauen welche die Topologie der Segmentierungsmaske berücksichtigt. Diese erlaubt es, strukturelle Merkmale in den Daten zu erkennen, die jeweils bestimmten Segmentierungsfehlern zugeordnet werden können.

In einem ersten Schritt wurden die möglichen Segmentierungsfehler klassifiziert, und dann jeweils Lösungswege zum Beheben dieser Fehler erarbeitet. Basierend auf der Klassifikation können Benutzern passende Techniken zum Bearbeiten der Segmentierungsfehler vorgeschlagen werden; allerdings müssen die Defekte zuvor von den Benutzern selbst detektiert werden.

Im Zuge dieser Arbeit konnte der Bearbeitungsprozess weiter verbessert werden, indem problematische Regionen automatisch detektiert und klassifiziert werden. Folglich ändern sich die Interaktionstechniken insofern, dass Benutzer nun aus einer Auswahl von vorgeschlagenen

Lösungswegen wählen können. Die Auswahl der möglichen Lösungswegse wird zuvor passend zur Klassifikation des Segmentierungsfehlers automatisch erstellt. Die Klassifikation der Segmentierungsfehler erfolgt aufgrund einer Analyse der zugrundeliegenden Unterschiede der Segmentierungsmaske. Die möglichen Lösungswege werden in einer glyphbasierten Visualisierung präsentiert, welche eine Analyse der Daten in unterschiedlichen Detaillierungsgraden erlaubt. Die Benutzer müssen dadurch nicht mehr direkt mit den Maskendaten interagieren, sondern können sich auf die vorgeschlagenen Lösungswege konzentrieren, was den Aufwand und die Bearbeitungszeit drastisch reduziert.

Als weiterer und finaler Schritt dieser Arbeit wurden weitere Techniken entwickelt, um die Benutzer während des gesamten Bearbeitungsvorganges zu unterstützen. Die Lösungswege wurden mit Hilfe statistischer Methoden bewertet, um sie danach den Benutzern in einer geordneten Form zu präsentieren. Die Ordnung repräsentiert dabei sowohl die statistische Signifikanz, als auch den passenden Detaillierungsgrad. Die Lösungswege werden in radialer Form hierarchisch visualisiert, wodurch die Benutzer zu den Regionen geleitet werden, die jeweils die größte Aufmerksamkeit erfordern. Während dem Betrachten einer bestimmten Region werden nur jene Lösungswege angeboten, die sich auch als relevant erweisen, um die Benutzer in der Darstellung nicht mit nicht benötigten Details zu überfordern.

Alle im Rahmen dieser Arbeit erarbeiteten Techniken wurden zusammen mit Radiologen evaluiert, wobei die Bedienbarkeit der Interaktionsmöglichkeiten, der Zeitaufwand und die Genauigkeit der Resultate bewertet wurden. Die Ergebnisse der Auswertung belegen einerseits die Vorteile unserer Techniken, in wenigen Interaktionsschritten ein genaues Segmentierungsergebnis zu erhalten, als auch deren Anwendbarkeit in verschiedenen Anwendungsszenarien.

Abstract

ARIOUS medical imaging techniques, such as Computed Tomography, Magnetic Resonance Imaging, Ultrasonic Imaging, are now gold standards in the diagnosis of different diseases. The diagnostic process can be greatly improved with the aid of automatic and interactive analysis tools, which, however, require certain prerequisites in order to operate. Such analysis tools can, for example, be used for pathology assessment, various standardized measurements, treatment and operation planning. One of the major requirements of such tools is the segmentation mask of an object-of-interest. However, the segmentation of medical data remains subject to errors and mistakes. Often, physicians have to manually inspect and correct the segmentation results, as (semi-)automatic techniques do not immediately satisfy the required quality. To this end, interactive segmentation editing is an integral part of medical image processing and visualization.

In this thesis, we present three advanced segmentation-editing techniques. They are focused on simple interaction operations that allow the user to edit segmentation masks quickly and effectively. These operations are based on a topology-aware representation that captures structural features of the segmentation mask of the object-of-interest.

Firstly, in order to streamline the correction process, we classify segmentation defects according to underlying structural features and propose a correction procedure for each type of defect. This alleviates users from manually applying the proper editing operations, but the segmentation defects still have to be located by users.

Secondly, we extend the basic editing process by detecting regions that potentially contain defects. With subsequently suggested correction scenarios, users are hereby immediately able to correct a specific defect, instead of manually searching for defects beforehand. For each suggested correction scenario, we automatically determine the corresponding region of the respective defect in the segmentation mask and propose a suitable correction operation. In order to create the correction scenarios, we detect dissimilarities within the data values of the mask and then classify them according to the characteristics of a certain type of defect. Potential findings are presented with a glyph-based visualization that facilitates users to interactively explore the suggested correction scenarios on different levels-of-detail. As a consequence, our approach even offers users the possibility to fine-tune the chosen correction scenario instead of directly manipulating the segmentation mask, which is a time-consuming and cumbersome task.

Third and finally, we guide users through the multitude of suggested correction scenarios of the entire correction process. After statistically evaluating all suggested correction scenarios, we rank them according to their significance of dissimilarities, offering fine-grained editing capabilities at a user-specified level-of-detail. As we visually convey this ranking in a radial layout, users can easily spot and select the most (or the least) dissimilar correction scenario, which improves the segmentation mask mostly towards the desired result.

All techniques proposed within this thesis have been evaluated by collaborating radiologists. We assessed the usability, interaction aspects, the accuracy of the results and the expenditure of time of the entire correction process. The outcome of the assessment showed that our guided volume editing not only leads to acceptable segmentation results with only a few interaction steps, but also is applicable to various application scenarios.

Contents

List of Figures

List of Tables

List of Algorithms

List of Mathematical Notations

Symbol	Meaning
$\lvert S \rvert$	Cardinality of the set S
$\mathbf{1}$	Indicator function
α	Significance level
A^2	Statistic of the Anderson-Darling test
avg	Average
$\mathcal{B}$	Bin for the ranking
B	Basin of the watershed segmentation
b	Histogram bin size
$\mathcal{C}$	Composite region
$\mathcal{D}$	Set of the data sampling locations
$\Delta d, \Delta d_1, \Delta d_2$	Delta values in the tests
Δ	Dissimilarity measure
δ_1	Dissimilarity value for detecting part defects
δ_2	Dissimilarity value for detecting layer defects
δ	Dissimilarity value of any kind
$D_\cap$	The histogram intersection metric
D_E	The Euclidean distance
D_n	Statistic of the Kolmogorov-Smirnov test
$D_S^{\max}$	Maximum of the skeleton distance
D_S	Skeleton distance
D_W	Metric for merging watershed basins
D	Distance between two skeleton voxels
$\mathbf{d}$	Set of the points for sampling the data values
d	Computed value of the statistic of the Kolmogorov-Smirnov test
$\mathcal{E}$	Number of randomized samples in the Monte Carlo test
$\hat{F}_{\mathcal{R}}$	Empirical Distribution Function of data values at the region $\mathcal{R}$
$F_{\mathcal{R}}$	Cumulative Distribution Function of data values at the region $\mathcal{R}$
Γ	Histogram bin-wise summation kernel
$\mathcal{H}$	Binned data for the Chi-Squared test
H_0	Null hypothesis

Symbol	Meaning
H_1	Histogram for detecting part defects
H_2	Histogram for detecting layer defects
IZ	Influence zone
J	The Jaccard coefficient
$\mathcal{L}$	Segmentation mask
L_B	Bottom level of the watershed basin
L_W	Watershed level between two basins
L	Layer defect in the segmentation mask
$\mathcal{M}$	Marker region
m	Number of the data dimensions
$\mathcal{N}$	Normal distribution
N_B	Number of bins in the histogram
N_G	Number of the growing iterations for the collection regions
N_R	Number of the ranks
Ω	Set of random variables
O_L	Over-estimation layer defect in the segmentation mask
O_P	Over-estimation part defect in the segmentation mask
O	Over-estimation defect in the segmentation mask
o	Outcome of the random variable
$\mathcal{P}$	Poisson distribution
P	Part defect in the segmentation mask
$Pr(E)$	Probability of the event E
pv	P-value
$\mathcal{R}$	Region, atomic region
R_1	Size of the kernel for detecting part defects
r	Randomized sample for the Monte Carlo test
$\mathcal{S}$	Space of the data sampling
S_0, S_1	Parameters for the check of the sample size
S_H	Set of histograms – argument for the dissimilarity measure
S_N	Set of neighbor voxels
$\mathbf{s}$	Location of the data samples in the space $\mathcal{S}$
sum	Summation
U_L	Under-estimation layer defect in the segmentation mask
U_P	Under-estimation part defect in the segmentation mask
U	Under-estimation defect in the segmentation mask
V	Random variable of the data value
ω	Weight in the Stouffer's Z-score method
W^2	Statistic of the Cramér-von Mises test
X	Statistic of the Chi-Squared test
Z_i	Z-score in the Stouffer's Z-score method
Z_S	Statistic of the Stouffer's Z-score method without weights
Z_W	Statistic of the Stouffer's Z-score method with weights

Glossary

Computed Tomography is a medical imaging modality that uses X-rays in order to measure the radiological density of various tissues of the human body xv, xxix, 1, 21, 54, 95

Computed Tomography Angiography is a medical imaging modality that combines conventional Computed Tomography with a contrast-enhancing agent that allows the radiologist to capture vascular structures xv, xxix, 3, 30, 52, 113

Direct Volume Rendering is a volume visualization technique that samples the data values along rays, looks up the colors from the corresponding transfer function and performs image compositing xvii, xxix, 8, 49

Electron Microscopy is a biomedical imaging modality that captures electron scattering from a specimen, physically cut into thin slices xviii, xxix, 79, 131, 134

Empirical Distribution Function is the distribution function of the outcomes of a random variable A. The outcomes are required to be sampled from identical and independently distributed random variables $A_1, \ldots, A_n \sim A$ xxix, 95

Hounsfield Unit is a unit of radiological density in Computed Tomography that is normalized to water (0) and air (-1000) media xxix

Magnetic Resonance Angiography is a medical imaging modality that combines conventional Magnetic Resonance Imaging with a contrast-enhancing agent that allows the radiologist capture of vascular structures xvi, xxix, 3, 29

Magnetic Resonance Imaging is a medical imaging modality that uses strong magnetic fields and radio-frequency pulses in order to measure atomic spin characteristics of various tissues of the human body xxix, 1, 46, 90

Non-Alcoholic Fatty Liver Disease is a liver disease that is characterized by build-up of fat deposits in the liver xv, xxix, 1, 30

Peak Signal-to-Noise Ratio is a ratio between the power of a signal and the power of noise that is often used as a measure of noise presence. It is measured in decibels xvii, xxix, 54, 120

X-Ray Computed Tomography is an industrial imaging modality which uses X-rays with higher energy levels compared to those of a conventional medical Computed Tomography

Acronyms

AD Anderson-Darling test 99, 100

CT Computed Tomography xv, xxix, 1–3, 5–7, 14, 21, 30, 33, 54, 95, *Glossary:* Computed Tomography

CTA Computed Tomography Angiography xv, xvii, xviii, xxi, xxiii, xxix, 3, 4, 7, 10, 12, 13, 30, 31, 33, 51, 52, 54, 59, 79, 80, 86, 113–115, 120, 128, 131, *Glossary:* Computed Tomography Angiography

CvM Cramér-von Mises test 99, 100

DVR Direct Volume Rendering xv, xvii, xxix, 2, 4, 8, 9, 49, 60, *Glossary:* Direct Volume Rendering

EDF Empirical Distribution Function xxix, 95, 96, 98, 100, 101, 106, 128, 131, *Glossary:* Empirical Distribution Function

EM Electron Microscopy xviii, xxix, 79, 83, 86, 131, 134, *Glossary:* Electron Microscopy

HU Hounsfield Unit xv–xvii, xxix, 2–4, 30, 44, 51, 54, 55, 57, 59, *Glossary:* Hounsfield Unit

KS Kolmogorov-Smirnov test 96, 98–100, 102, 108, 119

MRA Magnetic Resonance Angiography xvi, xxix, 3, 29, 35, *Glossary:* Magnetic Resonance Angiography

MRI Magnetic Resonance Imaging xxix, 1, 5, 14, 46, 90, 95, *Glossary:* Magnetic Resonance Imaging

NAFLD Non-Alcoholic Fatty Liver Disease xv, xvi, xxix, 1–3, 30, 31, *Glossary:* Non-Alcoholic Fatty Liver Disease

PSNR Peak Signal-to-Noise Ratio xvii, xxix, 54, 59, 61–70, 120, 128, *Glossary:* Peak Signal-to-Noise Ratio

1

Introduction

Challenges cannot possibly be good or bad. Challenges are simply challenges.

— Carlos Castaneda

A VAST variety of medical imaging techniques is used in clinical medicine on a daily basis. For example, Computed Tomography (CT) uses X-rays casted from multiple directions. As the rays are attenuated and absorbed by the patient's body, an X-ray image is formed. Using the Radon transformation, X-ray images from different scanning directions are combined into volume data. This acquisition modality captures the radiological densities of samples from various tissues. In case of Magnetic Resonance Imaging (MRI), a strong magnetic field combined with radio-frequency pulses allows the radiologist to capture spin characteristics of different atoms inside the tissues. Ultrasonic images capture echogenicity of tissues via sending an acoustic signal and analyzing the received echo. The doctors read the acquired images in order to examine pathologies, verify diagnoses, propose treatment plans, and evaluate their effectiveness. During their studies doctors extensively learn how to read medical images. The reason for this is that the human body has a complex structure, which should be reflected in the mental model of the doctors.

Many of the medical imaging techniques applied nowadays produce volumetric data, *i.e.*, a set of data values, sampled on a three-dimensional grid. In these images, various types of tissues are represented with different data values. Natural variations and inhomogeneities in tissue properties lead to local deviations of the measured data values. Combined with noise, which is inherent to all acquisition modalities, the resulting images, which the doctors investigate, are often difficult to comprehend. Given the possibility to visually inspect pathologies, the data values in the medical images often exhibit a great intra-patient and inter-patient variability. Therefore, doctors have to discriminate various types of tissues on their own, as automatic classification techniques often operate only in very restricted use-cases, *e.g.*, in exact phases of contrast agent enhancement and in the absence of severe pathologies.
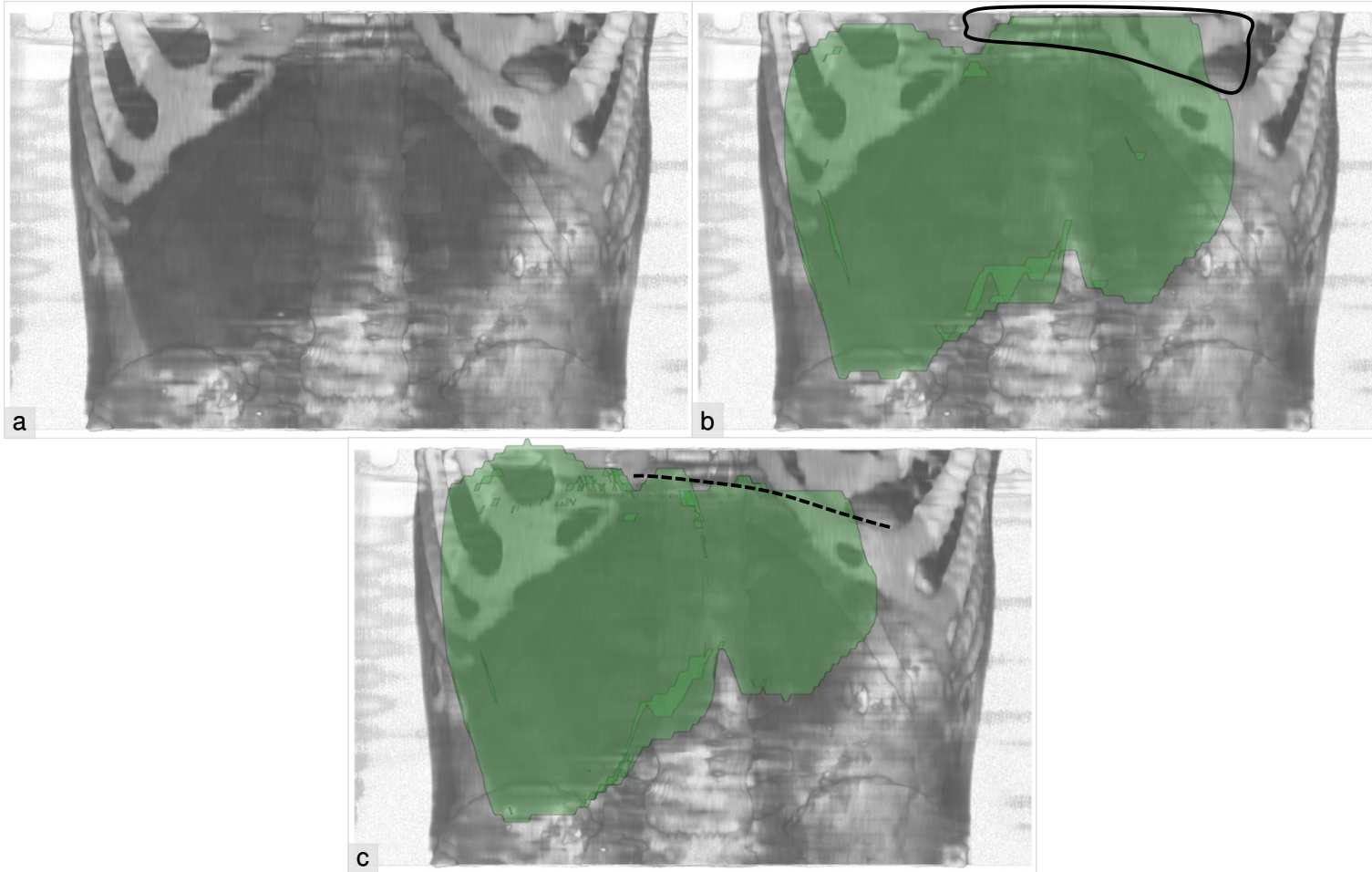
Figure 1.1: Abdominal CT dataset, depicting a severe NAFLD condition in the liver. a) Conventional DVR image with a window function of WL/WW=76/136 HU. b) Segmentation mask of the liver with a severe over-estimation at the heart, created by a fully automatic segmentation technique. c) Corrected segmentation mask of the liver without the over-estimation defect.

Medical imaging has a wide variety of application scenarios. First, doctors can find pathological regions and validate their diagnosis. For example, the Non-Alcoholic Fatty Liver Disease (NAFLD) can be detected through a CT examination. The major feature, which corresponds to a NAFLD diagnosis, is a build-up of fat deposits inside the liver. Thus, the density of the organ starts to decrease in terms of HU, getting close to that of the heart. The degree of the pathological process can be evaluated by an examination of the liver texture, performed on CT images. However, most of the automatic segmentation algorithms fail to segment the liver properly in this case, by for example including cardiac tissue into the segmentation mask. A typical case of a segmentation-algorithm failure is depicted in Figure 1.1.

Another example is vascular-structure diagnostics in Computed Tomography Angiography (CTA) and Magnetic Resonance Angiography (MRA) data. For these acquisition modalities, the doctors add a contrast-enhancing agent to the blood flow, which allows them to trace the vessels, including even small ones. Due to their complex structure, bones can be misclassified as contrast-enhanced vessels by segmentation algorithms, which leads to severe segmentation defects. In order to later proceed with analysis tools, the doctors have to edit the segmentation masks either manually or semi-automatically. As an example, the AngioVis framework [1] contains manual tools for a proper vessels/bones delineation. This application scenario is illustrated in Figure 1.2.

The segmentation masks of various organs can be used for planning an operational intervention. As an example, the liver segmentation mask is usually split into eight Couinaud segments, which rely on the major vascular structures in the liver. Then, a resection can be planned for each segment with respect to the pathology location, the vascular-structures accessibility and other factors. Volumes of the remaining liver parts are estimated in order to assess the viability of the operation. Thus, it is crucial to have available a segmentation mask of high quality.

Medical imaging also allows the doctors the comparison of organs and their pathologies over time. By examining medical images from several points in time, the doctors can judge the treatment progress as well as assess the aggressiveness of pathologies. In order to be able to employ (semi-)automatic comparative tools, though, medical images have to be co-registered and possibly segmented. Ideally, after these two preprocessing steps, the doctors would get an exact match of numerous features of the organ of interest and inspect in detail the changes that occurred between the acquisition times.

Having available the segmentation masks of organs, the doctors may perform various kinds of measurements. Often this involves the linear sizes of pathologies and their relative distance to vascular structures. Such information can be used in the tumor-malignancy assessment, which often determines the subsequent treatment.

The segmentation masks are often required by the doctors, who apply various data analysis techniques on them for diagnostic purposes. In this thesis, we discuss semi-automatic and manual segmentation techniques, as fully automatic solutions often operate only in very restricted scenarios with well-defined conditions and protocols, which sometimes are not possible to maintain during the data acquisition. Another class of segmentation techniques is represented by interactive segmentation-editing methods, which allow the user to edit the existing mask in order to rectify its deficiencies.
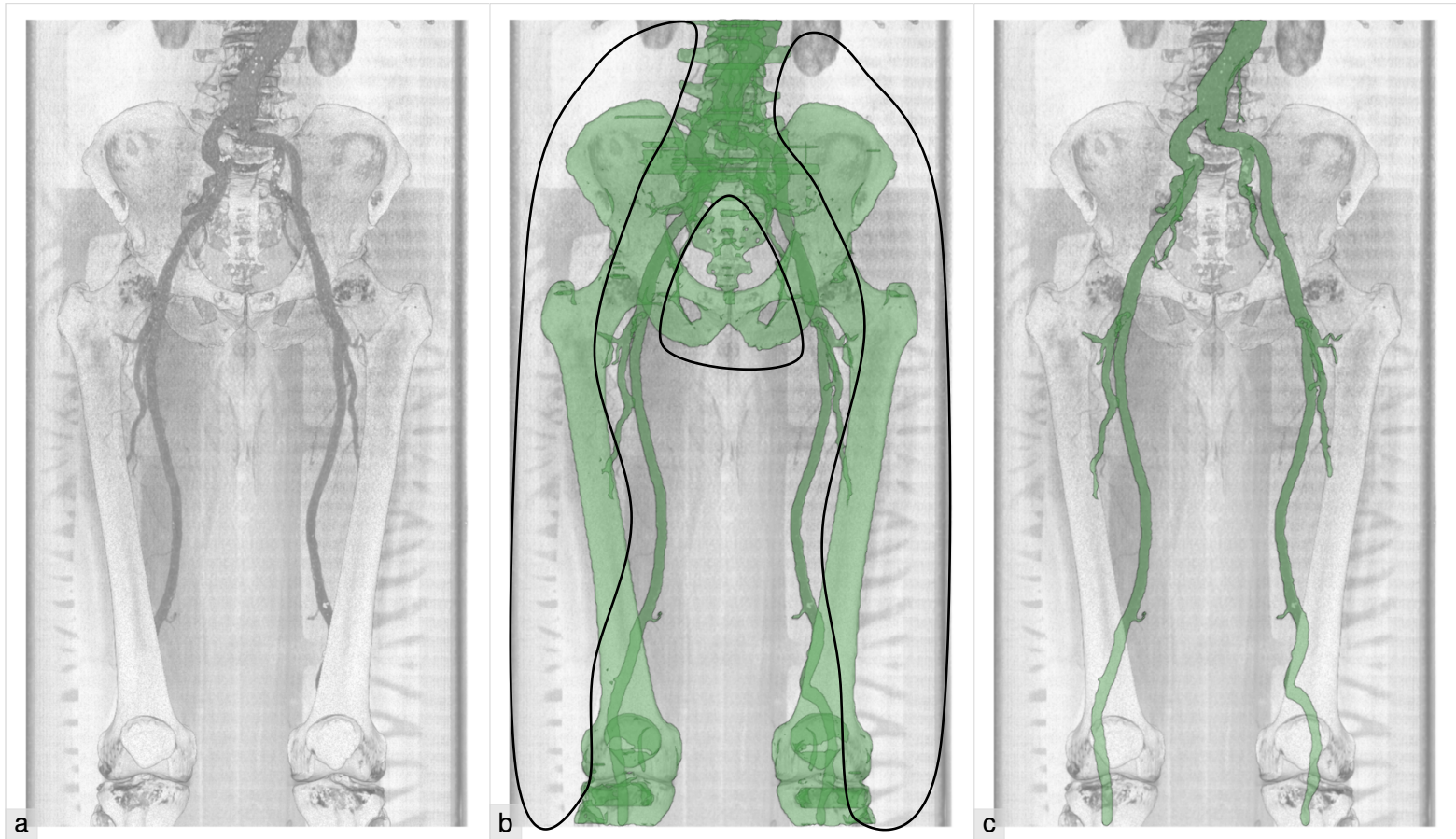
Figure 1.2: CTA dataset of the lower extremities, depicting a failure of vessel segmentation. a) Conventional DVR image with a window function of WL/WW=188/384 HU. b) Segmentation mask of the vessels, created by a fully automatic segmentation technique, with a severe over-estimation at the bones. c) Corrected segmentation mask of the vessels without the over-estimation defects.

We start with the most basic segmentation editing tools, such as the manual selection of voxels and the drawing of contours on 2D slices. These techniques provide the user with unlimited editing possibilities, as each voxel can be classified on its own. However, the flexibility comes at the price of a laborious, repetitive and error-prone process that requires an extensive domain knowledge and a long processing time.

In order to speed up the editing process, Mortensen and Barrett [2] propose semi-automatic intelligent scissors that operate on 2D slices. The user selects a "start node", *i.e.*, a pixel on the image, where the boundary of the object starts. Then the user places a "free point" where the detected boundary should end. As the user moves the free point, a "live wire" is automatically placed, *i.e.*, a curve that is automatically snapping to a boundary in the slice. To achieve such a behavior discrete dynamic programming is used to trace the optimal boundary between the start node and the free point. Finally, the user locks the free point, making it the "goal node". Then the next part of live wire is placed, assuming the last goal node to be the new start node. The boundary, composed of all live-wire segments, runs through all the nodes. The processing time strongly depends on the image resolution. The interactivity issue is addressed by Wong *et al.* [3]. They propose a "slimmed graph" in order to process large images interactively. Intelligent scissors use the image gradient, so they are sensitive to noise. The enhanced intelligent scissors proposed by Mishra *et al.* [4] use a phase-based image representation. They formulate and solve the optimal boundary extraction problem as an active contour problem. Enhanced intelligent scissors, as stated by the authors, are more tolerant to noise and require fewer nodes than conventional intelligent scissors.

An extension of live wire for volume editing was proposed by Hastreiter and Ertl [5]. They focus on brain segmentation from MRI and CT scans. The user defines contours in slices through the live-wire tool. The difference to conventional intelligent scissors is that their live-wire placement-procedure takes into account slabs, not individual slices. Then the contours are propagated to adjacent slices automatically. Falcão and Udupa [6] generalize a live-wire segmentation technique to 3D objects. The contours are placed in the slices that are either the original acquisition slices (axial slices in case of the CT and MRI acquisition modalities) or slices orthogonal to them. The control points in the orthogonal slices are taken from intersecting the corresponding contours of the original acquisition slices. In order to construct the contours in the orthogonal slices, the control points have to be properly ordered. If the ordering is incorrect, the corresponding contour would not lie at the object surface. Instead, it would run either inside or outside the object. Hamarneh *et al.* [7] and Poon *et al.* [8] extend this idea by automatically generating new live-wire contours from the ones the user specified. The user specifies the contours in planes $\pi_1, \ldots, \pi_M$. Planes $\kappa_1, \ldots, \kappa_N$ that contain the new contours are arbitrary. For each new contour in a plane κ_i the control points are computed by intersecting the user-specified contours in the planes $\pi_1, \ldots, \pi_M$ with the plane κ_i. In fact, the authors first compute intersection of the plane π_j with the plane κ_i. There are three possible outcomes for the intersection. If the planes π_j and κ_i are parallel, there is no intersection between the planes. Therefore, the corresponding user-specified contour does not contribute to the control points of the new contour. If the planes π_j and κ_i are coincident, the user-specified contour is used directly as the new contour. If the planes π_j and κ_i are concurrent, the intersection of π_j and κ_i is a line. Along the intersection line,

the algorithm determines line segments that lie inside the corresponding user-specified contour. The control points are placed at end-points of the found line segments. For the plane κ_i, the algorithm collects all the control points and the line segments from all the planes $\pi_1, \ldots, \pi_M$. The control points are then ordered according to the "turtle" L-system that moves along the found line segments and always turns left at their intersections. The latter work addresses the issue of segmenting objects with arbitrary topology. In order to do so, it detects the contours that are inscribed into another contour and marks them to correspond to topological features such as deep concavities, protrusions, non-spherical topologies, branchings, *etc.* An analogy of live wire for volume editing was presented by Armstrong *et al.* [9]. They propose a "live surface", which is defined by user interaction and refined by an optimization algorithm. The idea is the following: the user brushes "object" and "background" seeds in a specific cross-section plane. Additionally, mouse interaction on the live surface is taken into account to either get more object voxels, or remove background voxels. As this method allows the experts sequential refinements, based on their input, the user has to manually search for the defects and place seeds accordingly.

Beck and Aurich [10] approach the interactive segmentation problem as follows. The user first specifies several seed points inside the object-of-interest. An automatic flood-fill algorithm grows the selection until the built-in classification procedure finds significant differences in the data-value distribution. The intermediate result often contains over-estimated regions that are subsequently removed with a "virtual knife" interaction tool, which operates in the 3D view. Under-estimation defects are fixed by specifying additional seed points.

For graph cuts a graph is constructed from voxels serving as its vertices, and connections between adjacent voxels serving as its edges. A cost is assigned to an edge according to characteristics of both voxels. A graph cut is a graph partitioning: voxels in the first part belong to the object, and voxels in the second part belong to the background. The partitioning depends on user-classified voxels ("object terminals" and "background terminals"). The optimal cut is found with respect to the sum of the costs of the edges in the cut-set. Hard constraints ("seeds") are applied to correct the position of the graph cut locally. Both terminals and seeds are placed by sketching. The method can be utilized for segmentation correction. A drawback is that terminals and seeds have to be placed manually instead of supporting feature-based selection and editing. Liu *et al.* [11] apply the graph-cut technique to separate pre-segmented bones in CT images. Each initially specified seed point corresponds to an individual bone. As the user progressively adds more seed points, the bones become separated and properly classified. Beichel *et al.* [12] utilize graph cuts to get an initial segmentation mask. Later, the user interactively refines it with two post-processing steps: chunk-based refinement (CBR) and mesh-based refinement (MBR). The CBR stage allows the user to split the segmentation into connected components ("chunks") and interactively add or remove them from the mask. With the MBR stage, the user edits finer features of the segmentation mask. Egger *et al.* [13] focus on restricting the graph cut by additional seed points, which the user places in order to quickly rectify the segmentation defects. A survey on graph cuts can be found in the works of Boykov *et al.* [14, 15].

Sketch-based methods employ simple and intuitive sketching operations on slices, which modify the contours of the object. One of these methods is proposed by Heckel *et al.* [16], where radial basis functions are used to construct an implicit smooth surface of the object out of several

sketches. Sketching over existing contours, combined with some heuristics in order to determine the required correction operation, is the basis of the approach by Heimann *et al.* [17]. The work of Salz *et al.* [18] includes sketches over the object and the background, which are then used for a geodesic segmentation of heart and lungs.

Approaches that employ deformable 3D meshes, like the ones proposed by Bornik *et al.* [19] and Proksch *et al.* [20], provide easy to use and efficient interaction tools for volume editing. Jackowski *et al.* [21] present arbitrary surfaces in 3D with Rational Gaussian surfaces, which can be subsequently edited. As an application example, the authors present liver, spleen, kidney and brain-tumor segmentations. Timinger *et al.* [22] propose to correct the segmentation mask, represented as a mesh, via deformations by user-specified forces. Random walk methods [23, 24] classify the data based on seed points. From each voxel a random walk is performed and the probability of reaching each seed point is evaluated. Then, each voxel is assigned to the class of the seed point with the maximum probability. The watershed transformation can be the basis for an interactive segmentation procedure as well. This was demonstrated by Hahn *et al.* [25], who consider the case of removing bones from CTA images. Heckel and Braunewell [26] propose a segmentation-editing concept, which is based on an interactive watershed transformation [27] combined with an analysis of foreground and background markers provided by the user. Another possibility to interactively segment 3D objects is to build a so-called Active Appearance Model, fit it automatically to the data and then allow the user to locally modify it in order to refine the results. Beichel *et al.* [28] apply this approach to medical data and segment the following objects: thoracic cavity and diaphragm, liver, and various tumors. Neumann and Lorenz [29] review several statistical shape models and analyze their performance for segmenting vertebra from CT images.

Two recent methods [30, 31] direct the user to possible segmentation defects. Still, these methods do not suggest how to correct them. Praßni *et al.* [30] employ a random walk method and present ambiguous regions with high segmentation uncertainty in a 3D overview rendering and detailed uncertainty iso-lines in slice views. Top *et al.* [31] determine ambiguous regions of a segmentation and display them in a slice view where the user can perform corrections with live-wire editing.

Several approaches focus on improvement of the segmentation techniques from a different perspective. Torsney-Weir *et al.* [32] build up a statistical model of an arbitrary fully-automatic segmentation technique. While the model is created, the parameter space of the segmentation technique is sampled and the segmentation output is analyzed. The method then allows the user to visually explore the model with respect to the various parameters. It estimates the output segmentation using the model and computes related uncertainty information. Finally, the user arrives at "good" parameter settings that maximize the quality of the output segmentation masks. Fröhler *et al.* [33] investigate ensembles of the segmentation masks in order to determine optimal settings for an arbitrary segmentation technique. This method first performs sampling of the parameter space. The hierarchical clustering algorithm processes the output segmentation masks. The entire data is then represented by a hierarchy of the clusters that match segmentations of various spatial regions with the parameter values. As a result, the user can easily find the parameter settings that produce optimal segmentation results in certain regions of the data. For an extensive overview on interactive segmentation and volume-editing techniques in medical image

processing, we refer the reader to the survey of Olabarriaga and Smeulders [34] and the works of Heckel *et al.* [16, 35].

1.1 Segmentation Visualization

Another general topic with segmentation masks is their visualization. A segmentation mask should be incorporated into existing medical visualizations in such a way that it is, first, readable by the doctors and, second, does not introduce artifacts into the visualization. The most basic visualizations often render the segmentation masks with various semi-transparent colors on top of the gray-scale slice images serving as context. The boundaries of a segmentation mask are often highlighted to ensure easier delineation of the segmented object and the background. Such a visualization complies with both of the above requirements. However, with such approaches the doctors see only a single slice of the data at a time, which might be suboptimal if complex structures are segmented and visualized across numerous slices.

The segmentation mask can be rendered in 3D as well. The first option is to represent the surface of the mask with a polygonal structure, *e.g.*, triangles or quadrilaterals. The surface is usually calculated with Marching Cubes or Marching Tetrahedra algorithms. Then it is often post-processed, for example, to smooth unnatural shape edges or to remove staircase artifacts. The simplest solution is to apply Laplacian smoothing. This algorithm iteratively moves each mesh vertex by a vector, which is calculated from the positions of neighboring vertices. However, multiple iterations cause the mesh to shrink. Taubin's filtering rectifies this deficiency by alternating the orientation of the displacement vectors, *i.e.*, odd and even iterations grow and shrink the mesh respectively. Mönch *et al.* [36] discusses modifications to the smoothing algorithms, such as Laplacian or Taubin's filtering, which would prevent staircase artifacts as well as retain the resulting shape close to the original mask. Their modifications involve identification of faces which are related to the artifacts, and reducing the weighting of all such faces. Based on several metrics used to describe the differences in the shapes, the authors find that their modifications can effectively reduce the artifacts and preserve the remaining parts of the form.

Another possibility is to incorporate the segmentation mask into Direct Volume Rendering (DVR). Usually it is achieved by specifying different transfer functions for the mask and the background. However, artifacts may appear due to ambiguities in the binary classification which the segmentation mask provides. A typical solution is to use two-level volume rendering [37, 38]. This approach casts rays through the volume data. On a local level, each sample along each casted ray is classified in terms of rendered objects and shaded accordingly with specific rendering algorithm. Samples that belong to a single object are composited together with the same rendering algorithm. Then, on a global level, the individual objects along the ray are combined together with DVR compositing while respecting their depths and opacities. The classification of the objects at each sample is possible due to the usage of an object ID volume. This volume describes objects located at the corresponding positions of the original volume data. In order to determine the boundaries of the objects, several texture lookups are required.

Lindholm and Bock [39] address the segmentation-rendering problem by using two texture lookups and calculating class attenuation parameters. This allows the algorithm more efficient

texture access in terms of time; however, it may also result in certain artifacts. Such a scheme is between the classification that uses only a local sample and the one that operates with all neighbors around it. More advanced rendering techniques, *e.g.*, with contours by Salah *et al.* [40], may be more suitable in certain situations. In the general case, the two-level volume-rendering algorithm provides results of high quality without noticeable artifacts.

In our works, we use both approaches for visualizing the segmentation masks. In Chapter 2, the segmentation mask is rendered with its own distinct transfer function via DVR. In order to enhance the perception of the shape, we additionally extract and shade a surface of the segmentation mask. This surface is found by comparing the object ID's of the current and previous samples along the ray. Such a binary procedure is fast, however, it results in some aliasing artifacts. Therefore, in Chapters 3 and 4 we utilize mesh-based rendering. First, we extract the surface of the segmentation mask with a Marching Tetrahedra algorithm by Müller and Wehle [41]. Then, we perform a moderate Taubin filtering [42] in order to ensure a general smoothness of the resulting shape. Finally, we render the triangle mesh with a conventional order-independent-transparency algorithm, based on the fragments lists. For details on the order-independent-transparency rendering methods, we refer the readers to the work of Knowles *et al.* [43]. We enhance the shape of the object by displaying its contours. The contours are determined by analyzing depth differences in the ray profiles among neighboring screen pixels. The ray profiles contain entry and exit points of the segmentation mask.

1.2 Segmentation Defects

Motivated by the fact that most of the automatic segmentation techniques produce segmentation masks with defects in imperfect acquisition settings, we differentiate the defects by types depending on their solution procedures. The type of defect influences the interactive process of selecting the defect as well. For each class we propose its own treatment strategy. First, we distinguish between the following two types of defects of a segmentation (see Figure 1.3):

- *Over-estimation defects* (O) are regions of the volume that are incorrectly classified as belonging to the object. For instance, they can be caused by pathological conditions such as fibrosis or steatosis, which change the radiological density of the object itself. This compromises boundary identification between the object and surrounding tissues, because the classifier cannot distinguish them anymore. Another possibility for the over-estimation defects to appear is the usage of a contrast enhancement agent. While increasing the acquired data values of the blood, the agent may cause the data values to be as high as for the bones. Then, the separation between touching bones and vessels requires user intervention, as they are not separable by the data values alone. This problem requires tools, which enable the quick removal of background regions from the object.

- *Under-estimation defects* (U) are parts of the object that are misclassified as being background. It is usually due to pathological conditions like cysts or tumors, which change the radiological density of the affected regions. This forces a classifier of the segmentation procedure to judge that the regions contain another type of tissue than the object itself. In

the case of completely occluded vessels, stents or bypasses, the blood with the contrast agent will not flow through certain parts of the vessel tree. In this case, the segmentation mask will not contain those parts, as their data values would be outside the data range for most of the segmentation techniques. A way to treat such a problem is to provide volume-editing tools, which allow users to easily add the misclassified regions to the object.

These two types of defects require us to operate either on the object (O) or on the background (U) respectively.

We distinguish between defects that occur in the object's boundary layer (layer defect L) and those that affect whole parts of the object (part defect P). These two types of defects require different solution operations, *e.g.*, all voxels of P defects should be removed from the corresponding mask, and only the surface layer of L defects should be reclassified. The L defects usually manifest themselves as local artifacts in the boundary of the detected object, which occur due to noise in the data or pathological conditions like inflammation. In certain cases, the detected boundary of the object may be shifted by a constant number of voxels, which may be due to partial volume effects and a complex structure of the boundary of the object. This may result in the boundary artifacts as well. In total, we classify the defects into four different types (O_P, O_L, U_P, U_L), which are illustrated in Tables 1.1, 1.2. Table 1.1 depicts the segmentation defects in abdominal CTA datasets, and Table 1.2 gives examples for CTA of human lower extremities.

1.3 Skeletonization

As our solutions to a great extent rely on skeletonization, we discuss various skeletonization approaches. We represent a segmentation mask by its skeleton. The main idea behind such a representation is to capture structural features of the segmentation mask and allow the user to edit the defects, which are associated with these distinct structural features.

We start with thinning-based skeletonization approaches. Such techniques [44–46] iteratively remove surface voxels under geometrical and topological constraints, *i.e.*, preservation of end-points, desired skeleton width, connectedness and centered location of the skeleton [44]. These approaches are based on the definition of a *simple* point, *i.e.*, the voxel that can be removed without changing the topology of the object and its connectivity. Various definitions of simple points are proposed by different authors. Often, such definitions involve the Euler characteristic, computed in the 26-neighborhood of each voxel, *e.g.*, as in the work of Lee *et al.* [44]. End-points of open curves are simple points by definition. As a result, such a naive criterion for point deletion would reduce open curves to single voxels and misrepresent the original object. Therefore, end-points are detected and preserved in the thinning-based skeletonization methods. The requirement for each removed voxel to be at the surface of the object is checked using the 6-neighborhood. The desired skeleton width of one voxel is achieved by continuing the thinning iterations until no more voxels can be removed without changing the topology and connectedness of the object. During the iterative process the order of voxels to be removed matters. Classical approaches tend to address this issue by treating six principal thinning directions sequentially. Each direction corresponds to a non-object voxel in the 6-neighborhood: in front of, behind, on the left of, on
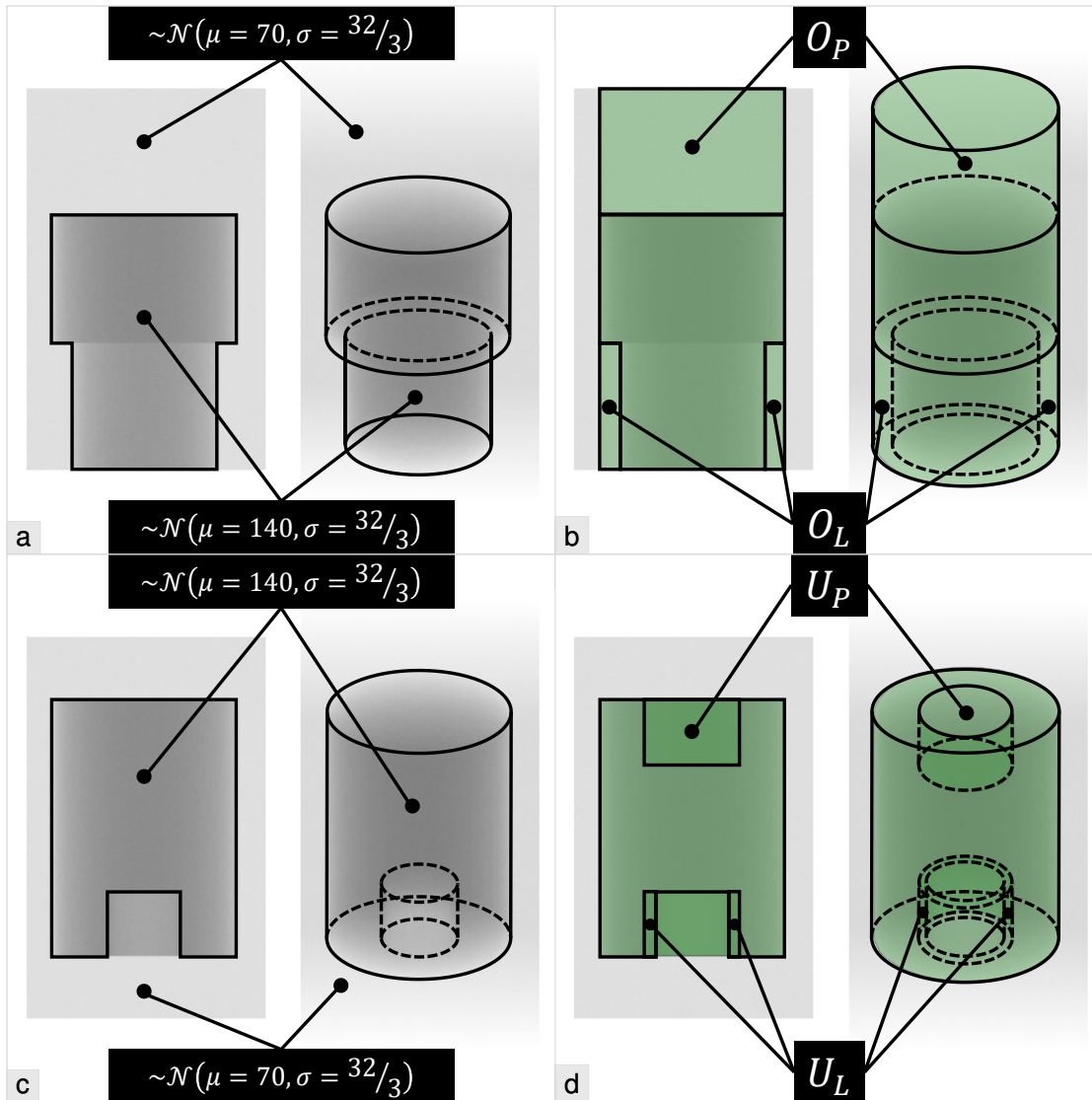
Figure 1.3: Two phantom datasets with the following segmentation defects: a) object #1 (dark gray), b) segmentation (green) of the object #1 with O_P, O_L over-estimation defects, c) object #2 (dark gray), d) segmentation (green) of the object #2 with U_P, U_L under-estimation defects. The defects are exaggerated for illustration purposes.

Table 1.1: Different classes of segmentation defects in abdominal CTA data. Objects of interest are livers, segmented with a fully automatic technique. The segmentation masks are displayed via meshes (green) atop the anatomical context (gray tones). The user interaction is required in order to achieve the desired quality level of the segmentation masks. In the first dataset, the O_P and U_P defects are fixed. The user rectifies the O_L and U_L defects in the second dataset. The result row depicts the desired segmentations.
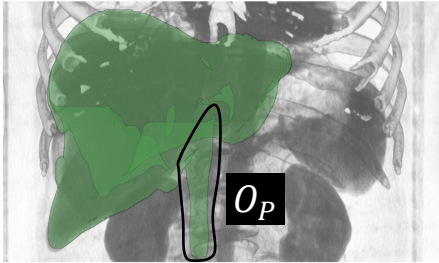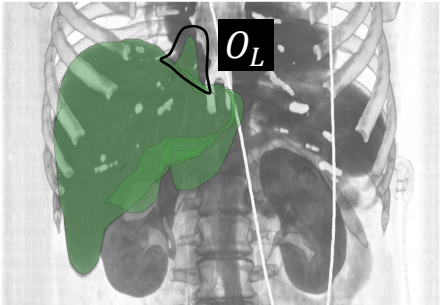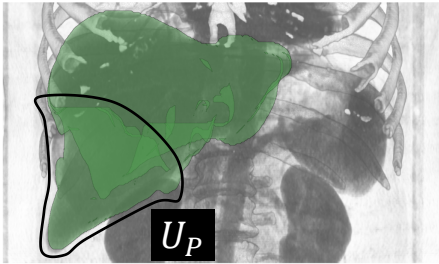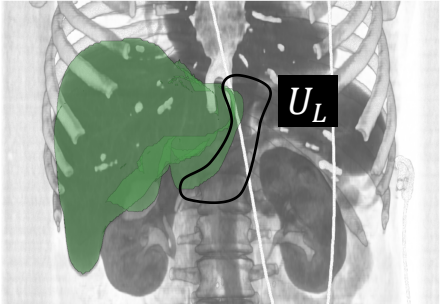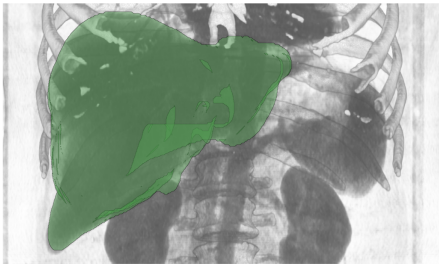
	Dataset with part defects P	Dataset with layer defects L
Over-estimation O		
Under-estimation U		
Result		

Table 1.2: Different classes of segmentation defects in lower extremities CTA data. Vessels are segmented with a fully automatic technique. The segmentation masks are displayed via meshes (green) atop the anatomical context (gray tones). The user, interested only in vessels, corrects the segmentation mask, achieving the desired quality level. In the dataset, the following defects are sequentially fixed: O_P, U_P, O_L, and U_L. The edited mask is shown in the result column.

	Part defect P	Layer defect L	Result
Over-estimation O	O_P	O_L	
Under-estimation U	U_P	U_L	
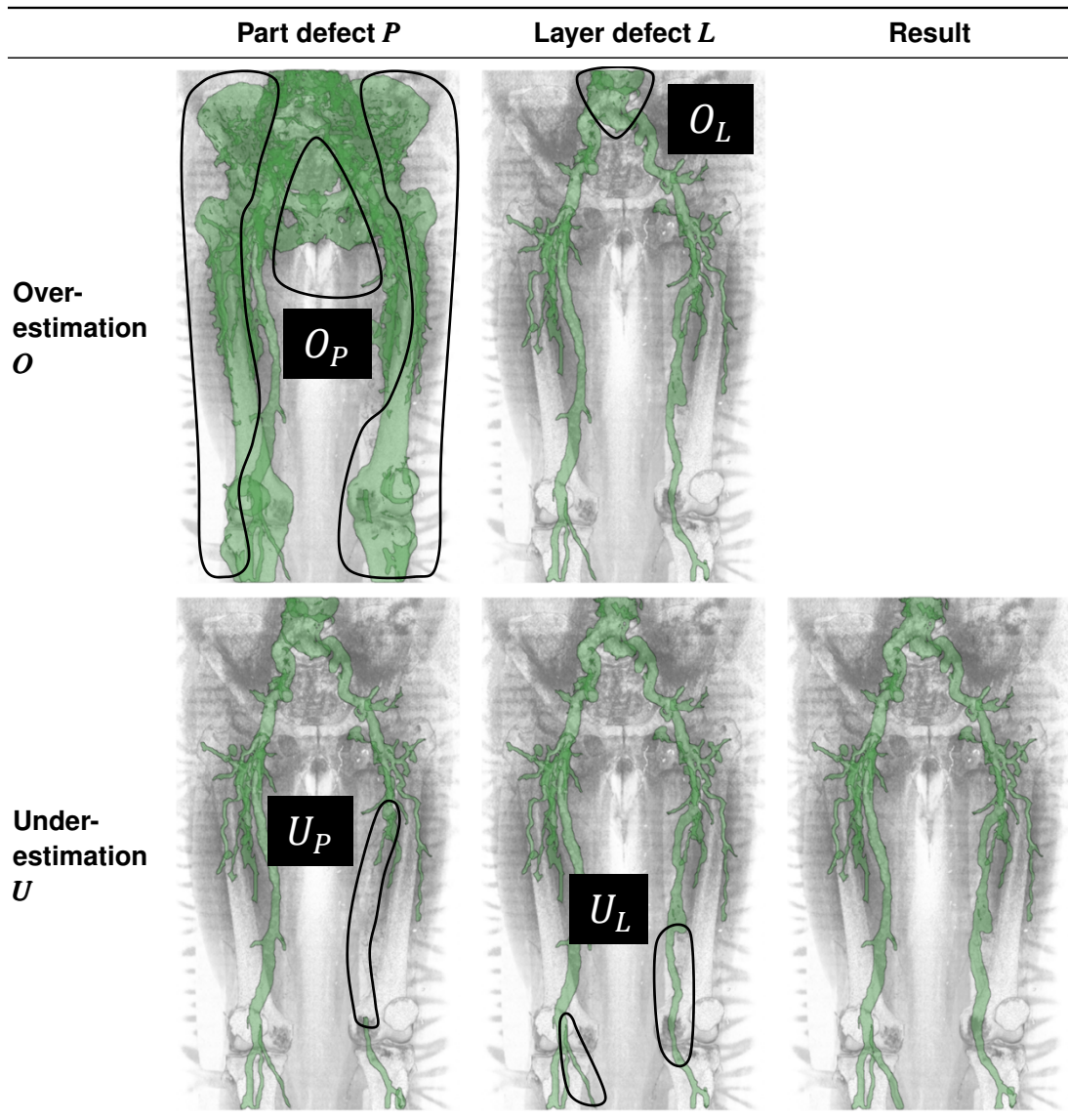
the right of, below and above. Each thinning iteration treats six directions in six sub-iterations, thinning the object equally from all sides. Pudney [47] suggests Distance-Ordered Homotopic Thinning, which orders the voxels for removal by their Euclidean or chamfer distances to the surface. As a result, the centered location of the skeleton with respect to the object is guaranteed.

Distance-field-based methods [48–53] treat ridges of a distance field as skeleton voxels. However, discreteness of the volume data often causes misclassification of the ridges of the distance field, thus, a post-processing step of reconnecting the skeleton voxels and pruning the spurious skeleton branches is required. Different types of distance fields result in different skeleton characteristics [54]. Most often the Euclidean distance is used for the skeletonization. However, its discrete approximation – the chamfer distance (also known as $\langle 3, 4, 5 \rangle$ distance) – may be used as well, *e.g.*, as done in the work of Gagvani and Silver [49]. Zhou *et al.* [48] define their own distance transformation, which propagates iteratively in the 6-neighborhoods of processed voxels. Even though such a distance metric is less precise than the chamfer distance, the authors claim that the achieved precision is sufficient to operate on medical CT and MRI volume data. As an example, the authors show extracted centerlines in colon and brain datasets.

Some approaches search for a skeleton by constructing Voronoi diagrams [55–57]. By definition, a Voronoi diagram for a given set of points consists of lines, which are equidistant to these points. If the points are placed on the surface of the mask, the corresponding Voronoi diagram will contain a skeleton of the mask. Practically, the mask is represented with polygons (polyhedra), *e.g.*, squares (cubes), and the points coincide with vertices of the polygons (polyhedra). However, the diagram also contains many spurious skeleton branches that come from inaccuracies, introduced during the polygonal (polyhedral) approximation of the mask. A regularization of the Voronoi diagram is required in order to remove the spurious branches and retain the relevant skeleton branches that represent the mask. Ogniewicz and Ilg [58] suggest four residual functions that control the regularization process: potential residual, circularity residual, bi-circularity residual, and chord residual. The skeleton, extracted from the corresponding Voronoi diagram with these four residual functions, is guaranteed to be connected. The shortcoming of the Voronoi-diagram-based skeletonization methods is a long computation time, as the number of the processed points often exceeds a million.

The curve skeleton is a simplified one-dimensional representation of a three-dimensional object, consisting only of curves that are equidistant to the object's surface [59]. Such representation is different from a conventional topological skeleton, introduced by Blum [60] and generated by thinning-based, distance-field-based, and Voronoi-diagram-based approaches, because the topological skeleton of a three-dimensional object may contain surface patches. Cornea *et al.* [61, 62] generate curve skeletons using the vector field of a repulsive force. The force is generated by distributing electric charge over the object's surface. Critical points of the vector field are identified and classified into attracting points, repelling points, and saddle points. From each critical point, a path-line is started, following the vector flow. The path-lines combined with the critical points form the core skeleton, which is later improved by including low divergence points and high curvature points, detected via thresholding. This method provides a classification of the object's voxels with respect to separate lines of the curve skeleton, split by the critical points. Such a classification is achieved by following the vector field from each voxel towards the skeleton.

As a result, the object is decomposed into parts. The authors compare this component-wise decomposition to a manual "skinning" process, applied in animations of mesh-based models. The proposed decomposition scheme may be applied to the segmentation-editing task, where severe over-estimation defects would be represented by their own skeleton lines. However, the editing would be limited to the level of the object's parts, provided by the decomposition procedure, and without the possibility to fine-tune the selected defects. Cornea *et al.* [61] mention that there is no optimum for the divergence and curvature thresholds. The presence of segmentation problems may require tuning these thresholds during the interactive editing process.

In order to produce the curve skeletons, Reniers *et al.* [59, 63] propose a collapse measure, based on an advection process. The setup for the skeletonization is the following: Masses are placed at the surface of the object. During the skeletonization, they are moved towards the skeleton and then follow the skeleton lines. The collapse measure represents the amount of mass that was advected through the corresponding point in space. In order to handle objects of different sizes appropriately, the collapse measure is normalized beforehand. The skeleton is computed by thresholding the normalized collapse measure. As masses move from the surface towards the skeleton, this skeletonization technique maps object voxels to the corresponding skeleton voxels. This provides users the possibility to select parts of the object by interactively choosing some of the skeleton voxels and propagating the selection by the given mapping towards the surface. Such functionality can be employed in an interactive segmentation-editing framework. However, the resulting skeleton depends on the threshold for the collapse measure. Tuning of this threshold during the interactive editing process may be tedious, as the users first have to identify the segmentation defects and then check whether the set threshold and the corresponding skeleton are optimal for correcting the found defects.

For a discussion on skeletonization, we refer the reader to the works of Prohaska [64], Reniers [65], and Saha *et al.* [66]. In our works we use thinning-based techniques for skeletonization. Having the representation of open paths and the corresponding end-points in the skeleton, we can detect and edit severe over-estimation defects. For under-estimation defects, we operate on the skeleton of the complement of the object in a volume. While automatically detecting or manually tuning selections of defects, we require the connectedness of the skeleton, such that we have a fine-grained control over the selected parts of the object. Without additional post-processing, the connectedness of the skeleton is not guaranteed in case of distance-field-based methods. As our solutions are supposed to be interactive, we avoid time-consuming Voronoi-diagram-based approaches. Still, one can generalize our concepts to operate on top of the distance-field-based and Voronoi-diagram-based methods, if such techniques are of preference for specific objects or data acquisition modalities.

We provide skeletonization examples for two different classes of medical objects in Figure 1.4. First, we consider the skeleton of the liver segmentation mask that contains a severe over-estimation defect. In Figure 1.4a,b parts of the skeleton that correspond to the defect are circled. The segmentation masks of vessels and bones are skeletonized in Figure 1.4c,d. We start with the mask that contains both vessels and bones. There, bones and vessels have separate skeleton branches that meet at branching points of the skeleton, where vessels and bones touch. Still, it is
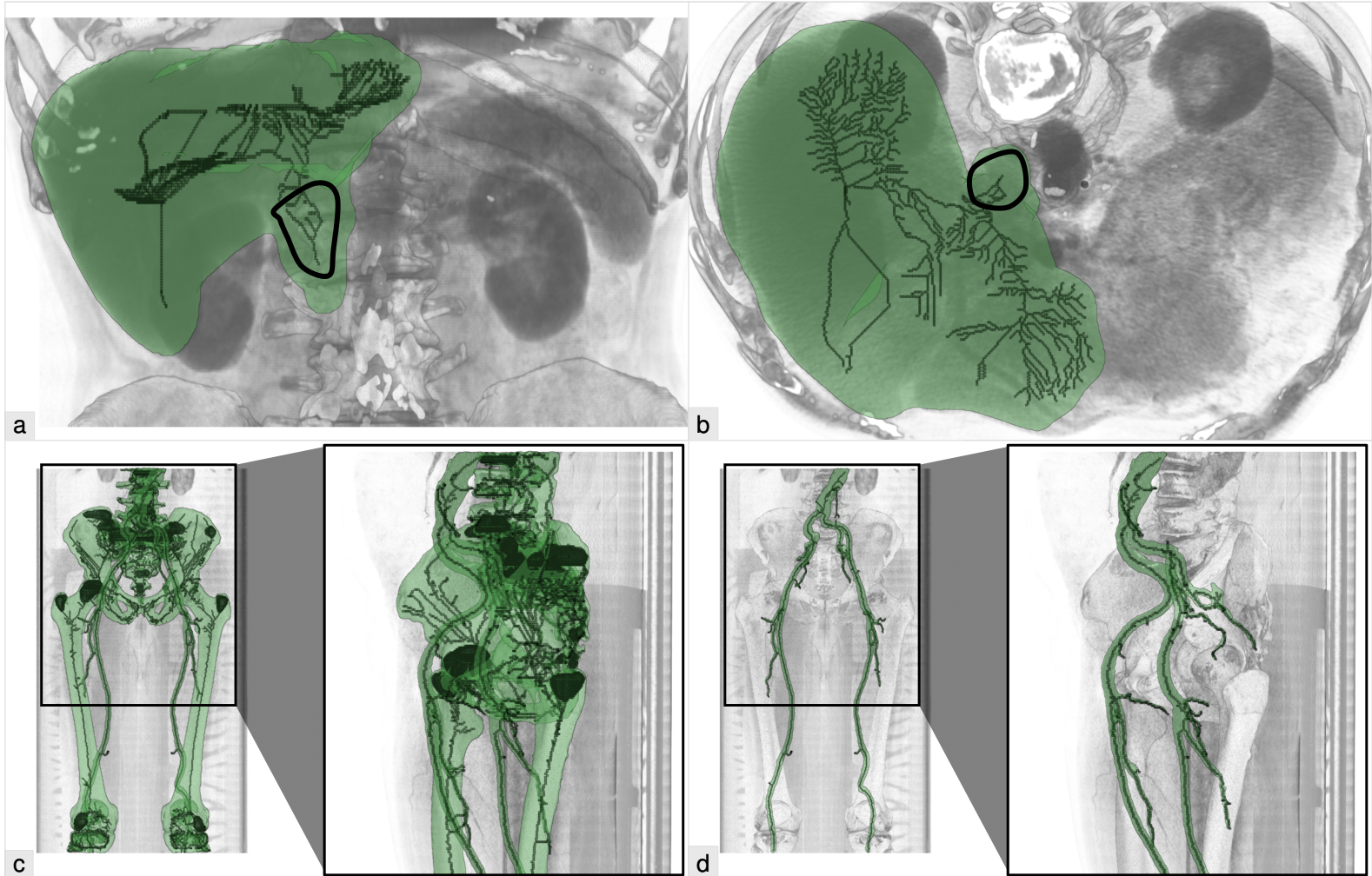
Figure 1.4: Skeletons of segmentation masks of the following medical objects: a,b) liver, c) bones and vessels, d) vessels only. Parts of the skeleton that correspond to the O_P defect are circled in (a,b).

possible to separate vessels from the bones, as demonstrated in Figure 1.4d. There, we display the skeleton of the vessel mask only.

1.4 Contribution

While dealing with segmentation editing, we identified the following research questions:

- **Q1**: How to select the defects prior to the correction?

- **Q2**: How to correct the selected defects?

- **Q3**: How to detect certain types of defects?

- **Q4**: How to present detected defects to the user?

- **Q5**: How to design the interactive editing process?

Our works contribute to answering these questions, aiding the user in achieving better segmentation quality in a shorter time, compared to existing segmentation-editing techniques. The overview of our contributions is presented in Table 1.3. Questions **Q1**, **Q2** and **Q5** apply to all segmentation-editing methods. The pro-active search for defects and the manual specification of the corresponding correction operations require considerable interaction effort from the users.

Several segmentation-editing approaches automatically detect defects or ambiguities in the segmentation mask and focus the user on rectifying them. Questions **Q3** and **Q4** are related to such additional functionality. For example, we propose such a technique in Chapter 3. Our method detects differences in data-value distributions, which are characteristic to segmentation defects, and generates suggestions to fix the found defects. Praßni *et al.* [30] employ uncertainty-visualization methods to guide the user towards ambiguously segmented regions of volume data. The ambiguity is determined per voxel from a probability, available from an underlying random walk algorithm. Top *et al.* [31] evaluate the stability of contours of the segmentation mask by perturbing paths locally in a live-wire procedure. This technique then suggests the user to validate the most unstable contours.

This thesis is organized as follows: In each chapter we describe in detail the approaches we developed in order to aid the user during the segmentation-editing tasks. First, we introduce a novel approach to correct the segmentation mask, which is represented by the skeleton. This way, we allow the user to easily select the segmentation defects via rather easy interaction operations on the skeleton. We adopt the classification of the segmentation defects from Section 1.2 in order to provide all necessary correction operations. For correcting the selected defects the users have to choose the operations manually.

Having the topological representation of the segmentation mask, we next aim at finding the regions of the mask that potentially contain segmentation defects. These are detectable by differences in data-value distributions. The idea here is to automatically detect all such regions and present them to the user, who will judge if defects are present and edit the mask if necessary.

Table 1.3: Overview of our contributions per chapter of this thesis.

Question	Chapter 2	Chapter 3	Chapter 4
Q1: *selection of defects*	Direct interaction on the skeleton	From the suggested correction regions and the corresponding operations	From the suggested regions with required dissimilarity characteristics
Q2: *editing operation*	User-specified operation	Suggested correction operation	Suggested correction operation
Q3: *detection of defects*	Not available	Via histogram dissimilarity metric	Via statistical significance of dissimilarities in data values
Q4: *visualization of defects*	Not available	Via glyphs	Via radial tree view and paths in it
Q5: *interaction process*	Manual selection of the defect and its treatment	Find necessary correction among large number of suggestions	Find necessary correction along the suggested path of exploration

Using our defects classification, we define for each type of defect the proper correction operation. To improve the interaction, we also focus on the visual encoding of the found defective regions and the corresponding correction operations. We assess the interactive editing pipeline we built with domain experts in order to verify the benefits of using our approach.

Finally, we present a statistical method for comparing various regions of the data according to the similarity of the underlying data values. With this approach we allow the user to correct segmentation masks of rather complex objects, such as entire vessel structures in the human lower extremities. We statistically evaluate the segmentation defects and offer the corresponding solution procedures at different levels of detail. We streamline the exploration of the solution procedures by the user with paths, which denote the most severe defects and their solutions. The evaluation conducted with domain experts demonstrates that our approach delivers results with a level of quality, that is comparable to a domain-specific clinically employed technique, with less interaction efforts and in a shorter time.

Skeleton-based Volume Editing

Intelligence is not to make no mistakes, but quickly to see how to make them good.

— Bertolt Brecht

Volume segmentation is important in many applications, particularly in the medical domain. Most segmentation techniques, however, work fully automatically only in very restricted scenarios and cumbersome manual editing of the results is a common task. In this chapter, we introduce a novel approach for the editing of segmentation results. Our method uses structural features of the segmented object to enable intuitive and robust correction and verification.
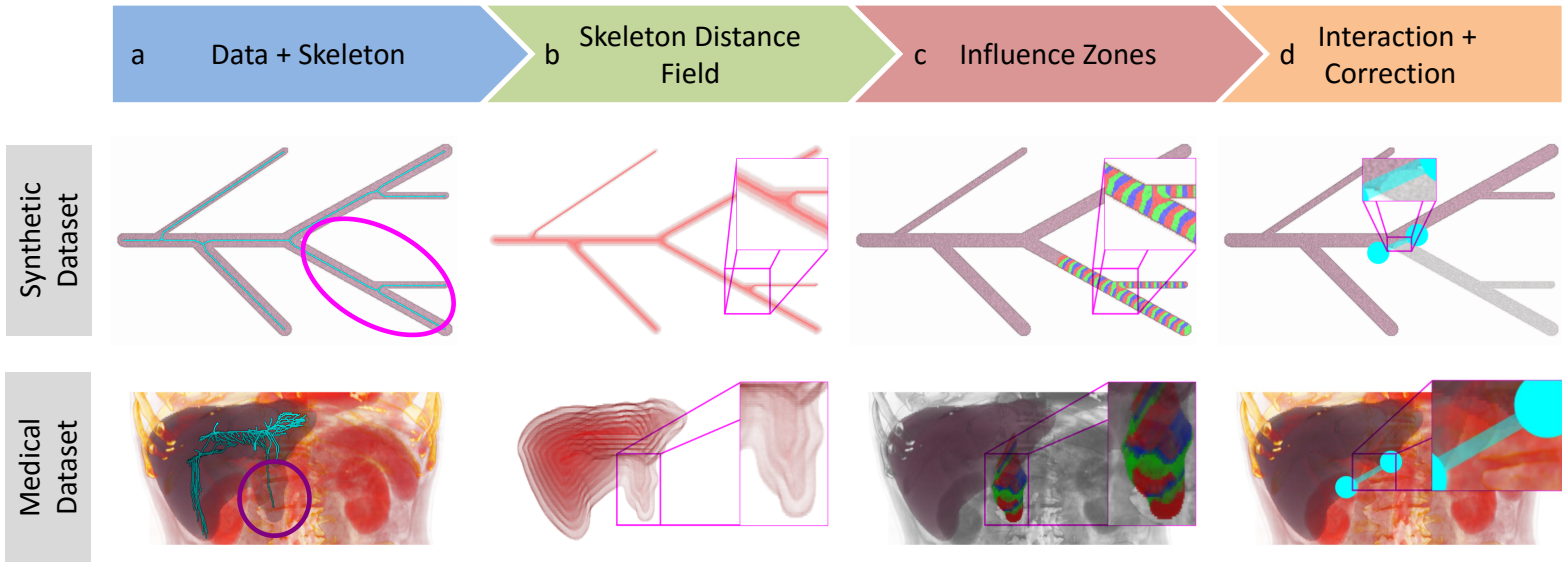
Figure 2.1: Example of the correction for a phantom dataset (top row) and a medical dataset (bottom row) using our new approach: a) features for the correction are circled, b) iso-surfaces of the skeleton distance field are shown, c) each color band represents a certain influence zone, d) user interaction is shown together with the correction results.

*C*URRENT volume editing approaches suffer from several issues. The main problem is an accurate selection of the features the user wants to correct. In the two-dimensional case, the user can work on a pixel level, achieving a high segmentation quality. In 3D, however, the additional dimension makes volume editing on a voxel level laborious, time-consuming, and error prone. The selection should require as little interaction as possible. Thus, it is essential to provide smart tools, which ease the selection of features. Furthermore, if the selection or the correction of an isolated feature affects other features of the object as well, then the editing process is not consistent. This happens if the method cannot distinguish different features of the object. These concerns lead to the conclusion that segmentation correction requires an interactive, predictable and robust volume editing method. The selection of the features should be precise and fast. The method has to detect the individual features and to process them separately.

We propose ViviSection, a volume editing method, which uses the skeleton of the object in order to detect structural features. These structural features represent the characteristic parts interesting to the user during correction. The method is based on novel skeleton-distance-field and influence-zone concepts. The components of the technique are illustrated in Figure 2.1. While our approach is applicable to other scenarios, we specifically focus on liver segmentation in Computed Tomography (CT) data, which is of high clinical importance. Due to a substantial degree of variation across scanners and patients, particularly in the presence of pathologies, current liver segmentation algorithms frequently require manual corrections [68].

2.1 Method

Our proposed method enables volume editing based on structural features. The pipeline of the method is illustrated in Figure 2.1 for two datasets: a phantom dataset with branchings and a liver segmentation dataset. First, skeletonization of the segmentation mask of the object is performed. We use a thinning-based skeletonization [44]. During the skeletonization, we also compute a skeleton distance field, which orders the object voxels with respect to the skeleton. Then we build influence zones. Each influence zone is defined by a certain skeleton voxel and contains a set of the object voxels that are linked to this skeleton voxel. The influence zones are determined using the skeleton distance field. They represent low-level structures of the object. A structural feature is defined as follows. First, a set of the skeleton voxels is chosen. The set should contain only one connected component of the skeleton voxels. Then, we add object voxels, which are in the influence zone of any of the chosen skeleton voxels. An incorrect segmentation differs from a correct segmentation by over-estimated and under-estimated regions. The regions can be represented as structural features and the artifacts can be selected on the surface of the structural features. Operations on the skeleton and the structural features can resolve segmentation problems, as illustrated in Figure 2.2.

We present several simple selection tools and operations on the skeleton and the structural features. The purpose of these tools is to select part of the skeleton based on the user input. The structural features, defined by the selected skeleton voxels, are then corrected according to the selected operation. The correction is previewed, so the user may refine it by additional input.
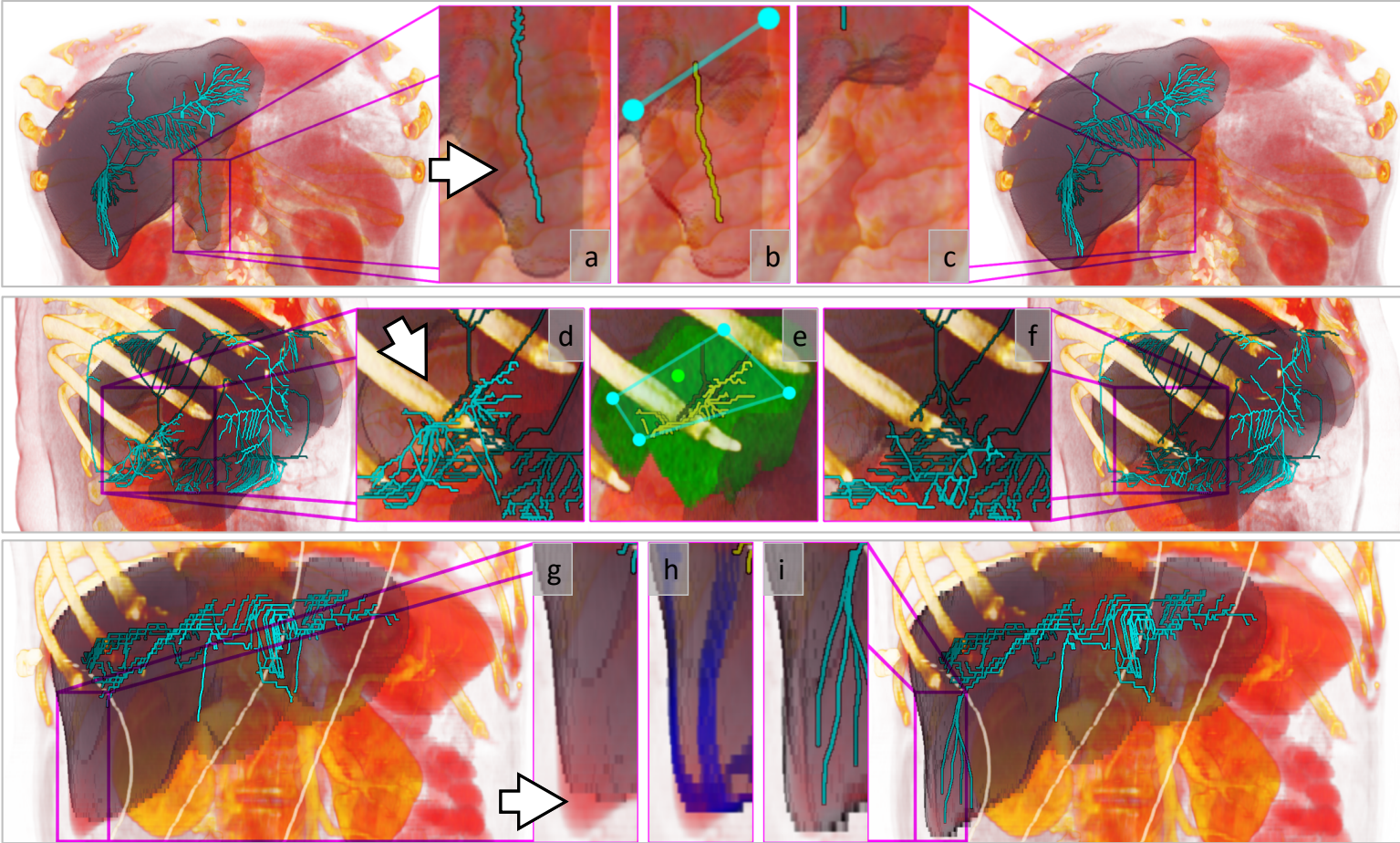
Figure 2.2: Manipulation on the skeleton and the structural features for the correction: a) a region with over-estimation O_P, b) manipulation on (a), c) correction of (a) by result of (b), d) a region with under-estimation U_P, e) manipulation on (d), f) correction of (d) by result of (e), g) a region with layer defect U_L, h) manipulation on (g), i) correction of (g) by result of (h). (a-c, g-i) depict the skeletons of the objects, (d-f) depict the skeletons of the complements of the objects in the volumes.
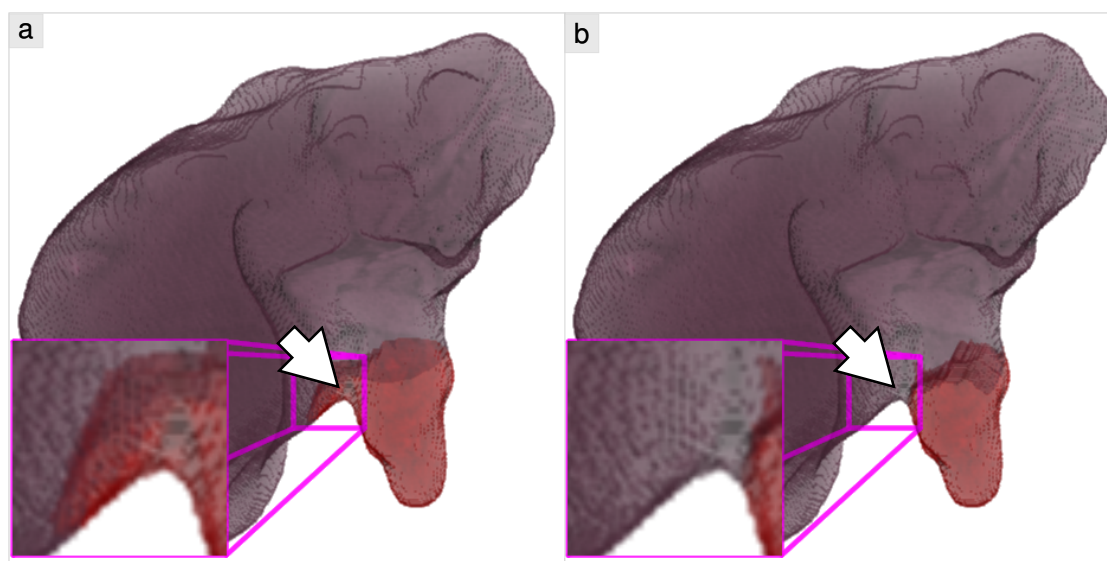
Figure 2.3: Comparison of (a) the Euclidean and (b) the skeleton distance fields for the influence zones calculation. The features are selected by the same set of skeleton voxels in both cases. Note the "leak" of the selection if the Euclidean distance is used.

2.1.1 Influence Zones

The influence zones can be determined in various ways. The criterion is that each object voxel should either be a skeleton voxel or belong to one and only one influence zone, *i.e.*, be linked to just one skeleton voxel. To determine influence zones we may use the Euclidean distance D_E. An object voxel o is linked to the skeleton voxel p only if the distance $D_E(p,o)$ is less than the distance to any other skeleton voxel. This link definition does not take the structures of the object into account, as it simply uses the shortest path between the two voxels. Thus, such a definition may lead to wrong influence zones in regions, where two or more object structures are presented. If some voxels of the first structural feature are incorrectly linked to the second feature due to wrong influence zones, the selection may "leak" leading to undesired result as illustrated in Figure 2.3a. As the selection should not interfere with other features, unless explicitly specified to do so by the user input, we use a skeleton distance field to include the delineation, based on the skeleton lines, into the definition of influence zones. Calculation of the influence zones from the skeleton distance field allows us to select part of the object without the undesired "leak", as shown in Figure 2.3b.

2.1.2 Skeleton Distance Field

The skeleton distance field can be computed during any thinning-based skeletonization. The main idea is that thinning iterations order the object voxels. By assigning the iteration number to each currently removed voxel, we receive a scalar field defined on the volume grid. We assume that the field is zero for voxels outside of the object. As skeleton voxels are never removed by the

Algorithm 2.1: Calculation of the skeleton distance field using the iteration numbers.

```
1 forall voxel v ∈ Volume do
2   if It(v) > 0 then                              // inside mask
3     D_S(v) ← It^max − It(v)        // D_S(v) is the skeleton distance of voxel v
4   else                                          // outside mask
5     D_S(v) ← −1
```

skeletonization, we assign them the value $It^{\max}$, which equals to the number of thinning iterations plus one. In order to construct the skeleton distance field, we perform arithmetic operations on the stored iteration numbers $It(\cdot)$, as explained in Algorithm 2.1. We denote as $D_S^{\max} = It^{\max} - 1$ the maximal skeleton distance, which is assigned to the surface voxels. The skeleton distance of each voxel v is denoted as $D_S(v)$.

The influence zone for a voxel o is determined by inspecting its neighborhood. If o has a neighbor s, which is a skeleton voxel, then the voxel o is in the influence zone of s. If o with a skeleton distance $D_S(o) = a$ has a neighbor r with a skeleton distance $D_S(r) = b$ and $b < a$, then the voxel o is in the same influence zone as the voxel r. With these two rules the influence zones grow from the skeleton voxels into the object voxels. The farther away the voxels are from the skeleton, the earlier they are removed during thinning and the higher their respective skeleton distances are. The influence zones grow from voxels with lower skeleton distances to voxels with higher skeleton distances. This is the reason why the $b < a$ condition is sufficient to determine the influence zone in the absence of skeleton voxels in the neighborhood.

Algorithm 2.2 provides details about the influence zones calculation using the skeleton distance field. Each voxel v has the following attributes. $State(v)$ stores a state of the voxel: *None* if the voxel is initialized, *Ready* if the voxel is enqueued for processing, *Processed* if the voxel is processed. $IZ(v)$ stores a link to the skeleton voxel, to whose influence zone voxel v belongs to. $S_N(v)$ returns a set of neighboring voxels of voxel v. Sets F and F_{new} ("fronts") store the voxels, whose influence zones still propagate. Sets Q and Q_{new} ("queues") store the voxels for processing. List L stores the influence zones in the neighborhood of voxel v and returns the most frequent one by MostFrequentElement(L). Voxel v is assigned to this influence zone.

The difference between the skeleton distance field and the Euclidean distance field is caused mainly by the delineation based on the skeleton lines. The Euclidean distance is propagated from a voxel to its neighbors. The skeleton distance depends on the skeleton lines, which represent object structures. If only one object structure is located in the region, then the skeleton distance is propagated in the same way as the Euclidean distance. This case is illustrated in the left close-up views of Figure 2.4. There the Euclidean and the skeleton distances change between layers, parallel to the surface of the object.

If two or more object structures are located in the region, then their delineation comes from the skeleton lines. This case is illustrated in the right close-up views of Figure 2.4. The Euclidean distance field does not consider the skeleton lines, so the Euclidean distance propagates between voxels of different object structures. Then the delineation of the resulting structural features fails

---

Algorithm 2.2: Calculation of the influence zones using the skeleton distance field.

```
 1  forall voxel v ∈ Volume do                                    // initialization of
 2   if D_S(v) = 0 then                                            // skeleton voxels
 3    State(v) ← Processed; IZ(v) ← v; F ← F ∪ {v}
 4   else                                                          // and non-skeleton voxels
 5    State(v) ← None
 6  d ← 1                                                          // begin at the skeleton of the object
 7  while d ≤ D_S^max do                                           // end at the surface of the object
 8   Q ← ∅; F_new ← ∅
 9   forall voxel v ∈ F do                                         // fill the queue Q
10   forall voxel w ∈ S_N(v) do
11    if State(w) = None then                                      // if unprocessed neighbor w found
12    if D_S(w) = d then
13     State(w) ← Ready; Q ← Q ∪ {w}                               // then enqueue w for processing
14    else
15     if D_S(w) ≠ −1 then F_new ← F_new ∪ {v}                     // or keep v
16   repeat                                          // process all voxels with skeleton distance equal to d
17    Q_new ← ∅
18    forall voxel v ∈ Q do                                        // process the queue Q
19     L ← ∅
20     forall voxel w ∈ S_N(v) do
21      if State(w) = Processed then L ← L ∪ IZ(w)                 // store IZ of processed neighbor w
22      if State(w) = None then                                    // if unprocessed neighbor w found
23      if D_S(w) = d then
24       State(w) ← Ready; Q_new ← Q_new ∪ {w}                     // then enqueue w for processing
25      else
26       if D_S(w) ≠ −1 then F_new ← F_new ∪ {v}                   // or keep v
27     IZ(v) ← MostFrequentElement(L)                              // assign IZ to v
28    forall voxel v ∈ Q do State(v) ← Processed
29    Swap(Q, Q_new)                                               // swap the queues
30   until Q = ∅
31   Swap(F, F_new); d ← d + 1                       // swap the fronts, increase the distance counter
```

---

as illustrated in Figure 2.3a. The skeleton distance field uses information on the skeleton lines, thus the skeleton distance does not propagate between voxels of different object structures. Due to this, the skeleton distance field allows the correct delineation of the resulting structural features as illustrated in Figure 2.3b. Here the selection of one feature does not "leak" into other features.
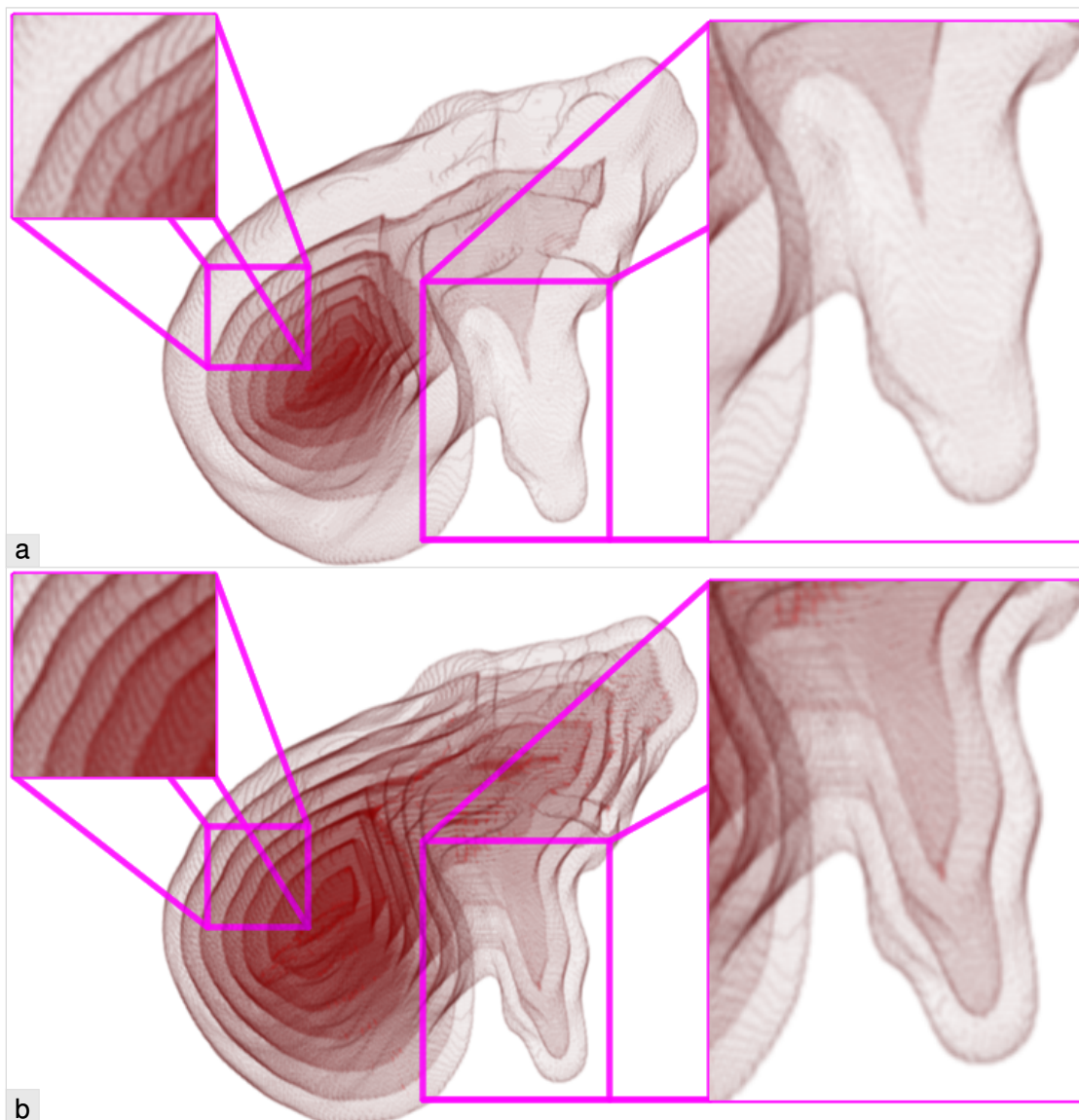
Figure 2.4: Comparison of (a) the Euclidean and (b) the skeleton distance fields. Iso-surfaces of the same values are shown in both cases. The left close-up views show a region with only one object structure. No strong differences appear. The right close-up views show a region with two object structures. Their delineation, based on the skeleton lines in the skeleton distance field, causes strong differences.

2.1.3 User Interaction

Selection tools and operations are the two main components of our user interface. The selection tools provide an interface for the selection of features to be corrected. Each selection tool defines a set of graphic objects, which are manually placed by the user on the projection of the volume.

The operations implement common volume editing tasks. If the user selects a different operation or selection tool or interacts with the selection tool, the selected feature for the correction is computed and displayed interactively. Additionally, the user can inspect the feature and results of the operation in slice views. The selection tools are illustrated in the part "selection tools" of Figure 2.5.

The **needle** is the simplest tool. The user specifies a point at the surface of the object in 3D space. Then, the skeleton voxel of the influence zone the point belongs to is located. The skeleton voxel splits the skeleton into two or more parts. The smallest part of the skeleton (according to the number of voxels) defines the structural feature selected for correction.

The **scalpel** tool operates on the structural features of the object or the background during the selection instead of defining a simple geometrical delineation as it is done in similar tools of other editing methods. This allows accurate feature selection from the detected structures in two mouse clicks instead of a cumbersome allocation and refinement of a separation plane or surface in slices. The user specifies two points in screen space, which define a line in screen space and a plane in 3D space. The skeleton voxels are split into two parts according to the negative and positive half-spaces of this plane. The smallest part of the skeleton (according to the number of voxels) defines the structural features selected for correction.

The **lasso** is the most powerful tool, but requires more interaction than the needle and the scalpel. The user specifies a selection point and a selection polygon in screen space. The polygon may consist of an arbitrary number of points and can be either convex or concave. It defines a prism in 3D space. The skeleton voxels inside the prism define the candidate structural features. The selection point defines a ray in 3D space to discriminate the features. If the ray crosses one of the candidate features, that feature is exclusively selected. Otherwise all the candidate features are selected.

The choice of the smallest part of the skeleton for the subsequent correction in the needle and scalpel selection tools is motivated as follows. With these tools, the skeleton is split into several parts. In order to proceed with the correction, we need to choose one part. We assume that most voxels of the object and the background are correctly classified. Therefore, most skeleton voxels and their corresponding influence zones do not require correction. Consequently, the smallest part of the skeleton (according to the number of voxels) is chosen, as the remaining parts contain most of the skeleton voxels.

The operations "add part" and "remove part" are designed for the correction of large-scale structural features. Small-scale structural features can be considered as local artifacts of the boundary of the object. To correct these artifacts "remove layer", "add layer", and "smooth" operations can be used. These three operations modify the boundary of the selected structural feature. To connect or disconnect parts of the object after the operations post-processing is used. The operations are illustrated in the part "operations" of Figure 2.5:

- The **add part** operation resolves under-estimation problems by adding to the object the selected feature from the complement of the object in the volume. Morphological closing with a kernel size of six voxels is applied as a post-processing step.
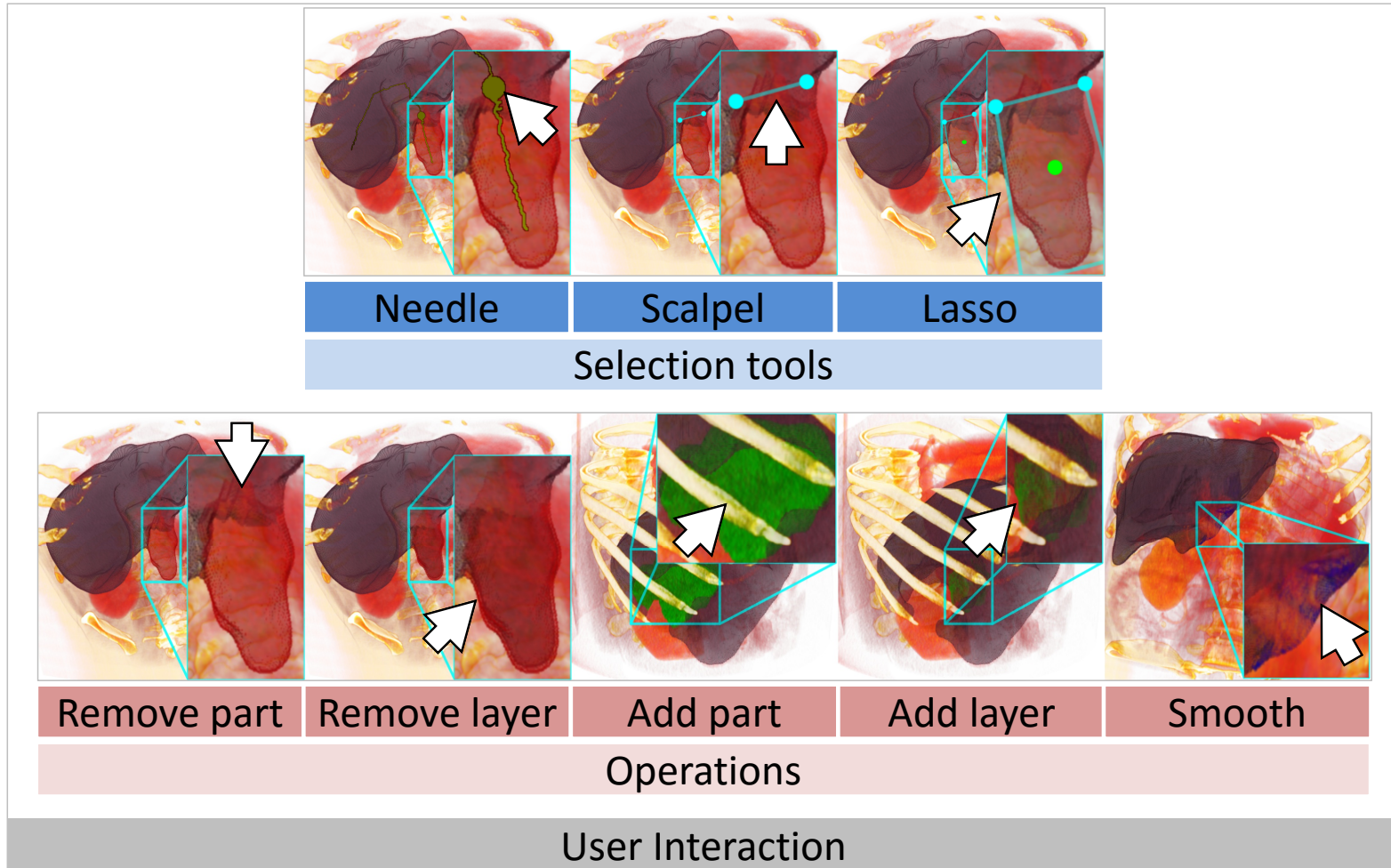
Figure 2.5: The selection tools and the operations available during user interaction. Previews of changes are shown.

- The **remove part** operation treats over-estimation problems by removing the selected feature from the object. Morphological opening with a kernel size of six voxels is applied as a post-processing step.

- The **add layer** operation grows a new layer, which has a thickness of $\sqrt{3}$ voxels, on top of the surface of the selected feature. Morphological closing with a kernel size of three voxels is applied as a post-processing step.

- The **remove layer** operation peels away the surface layer of $\sqrt{3}$ voxels thickness from the selected feature. Morphological opening with a kernel size of three voxels is applied as a post-processing step.

- The **smooth** operation replaces the selected feature with its smooth version. We use Gaussian blurring with a kernel size of 11 voxels and a threshold of 0.5 in order to compute a binary representation of the smooth version of the feature.

2.2 Performance and Scalability

Influence zones are pre-calculated to provide an interactive selection of the structural features during individual operations. For the performance evaluation we measured the following: pre-calculation time (generation of the skeleton, the skeleton distance field, and the influence zones), preview generation time (transformation of the user input into the selected structural features) and post-processing time (morphological operations).

We investigated twenty datasets in a quantitative evaluation. Pre-calculation times range from 2.8 to 16.7 seconds (average 8.2 seconds) depending on the volume size and the object's complexity. We use the skeletonization algorithm by Homann [69] that is based on the work of Lee *et al.* [44]. Parallelization is done via OpenMP on the CPU for up to eight parallel threads. The skeleton and the skeleton distance field are calculated simultaneously. The generation of the influence zones can take up to 25% of the pre-calculation time. Preview generation times are in range from 0.1 to 0.25 seconds depending on the skeleton complexity. Post-processing takes from two to five seconds depending on the volume size. Post-processing is executed via OpenCL on the GPU. All reported timings were measured on an Intel Core i7-2600K 3.4 GHz CPU equipped with 16 GB of RAM and an NVidia GeForce GTX 680 GPU with 4 GB of video memory.

Scalability is evaluated with respect to the skeleton complexity, measured as the number of individual skeleton lines. Magnetic Resonance Angiography (MRA) datasets with a resolution of $512 \times 512 \times 512$ voxels and containing brain-vasculature segmentation are used. Time measurements are given in Table 2.1. As the skeleton complexity decreases, both pre-calculation and preview generation times decrease. The morphological operations, which are the part of the post-processing time, are independent of skeleton complexity.

Every time the segmentation mask changes during interaction, the skeleton, the skeleton distance field, and the influence zones need to be updated. As this information is global, local partial updates in the region of changes are not feasible.

Table 2.1: Measurements for the scalability analysis. As previews of changes are generated during the user interaction, we require that the preview generation stage is interactive. The pre-calculation and post-processing stages only take place when the correction process begins or the segmentation mask is changed. These two stages are not required to be interactive.

Dataset	01	02	03	04	05
# of skeleton lines	13094	6099	2354	371	291
Pre-calculation time, sec.	10.6	6.4	5.5	5.1	4.8
Preview generation time, sec.	0.9	0.8	0.7	0.4	0.3
Post-processing time, sec.	6.8	5.5	5.1	5.1	4.9

2.3 Evaluation and Results

During the evaluation an automatic liver segmentation technique was used for initial segmentation. It is a region-growing procedure that operates on the segments from a 3D watershed segmentation. The initial seed point for the growing process is found automatically using the following assumption from general anatomy: liver is located below right lung lobe. The seed point is located by stepping a fixed distance down from the automatically segmented right lung lobe. Then radiological density distributions are used to grow the segmented region and stop at the boundary of the liver. The technique is targeted for Computed Tomography Angiography (CTA), and not standard CT.

2.3.1 Severe Pathological Cases

To demonstrate the quality of our approach in difficult clinical scenarios we conducted an evaluation using liver segmentations for patients affected by Non-Alcoholic Fatty Liver Disease (NAFLD) [70]. The liver is hardly differentiable from other organs on CT scans. To differentiate it better contrast-enhancing agents are used in CTA. The patient was scanned with a standard CT protocol after the diagnosis confirmation. Due to the NAFLD the liver parenchyma accumulated fat, causing radiological density to drop down to $10 - 30$ HU from the typical $50 - 70$ HU [71]. Usage of the standard CT protocol in this severe pathological case causes a failure of the applied segmentation technique. The problem here is that the boundaries between the liver and other organs (like kidneys, stomach, heart and bowels) are too weak, so that even a trained operator cannot clearly distinguish them. In this case, the segmentation technique produced severe overestimation, mainly towards the heart. Severe segmentation problems may change the skeleton of the object drastically, as it happens in this case. The correction of such a segmentation provides a robustness test for our method. The domain expert at the hospital used eight operations. He spent approximately seven minutes for the corrections. The calculations required 93 seconds in total. Ultimately, the expert achieved the desired segmentation, as given in Figure 2.6.
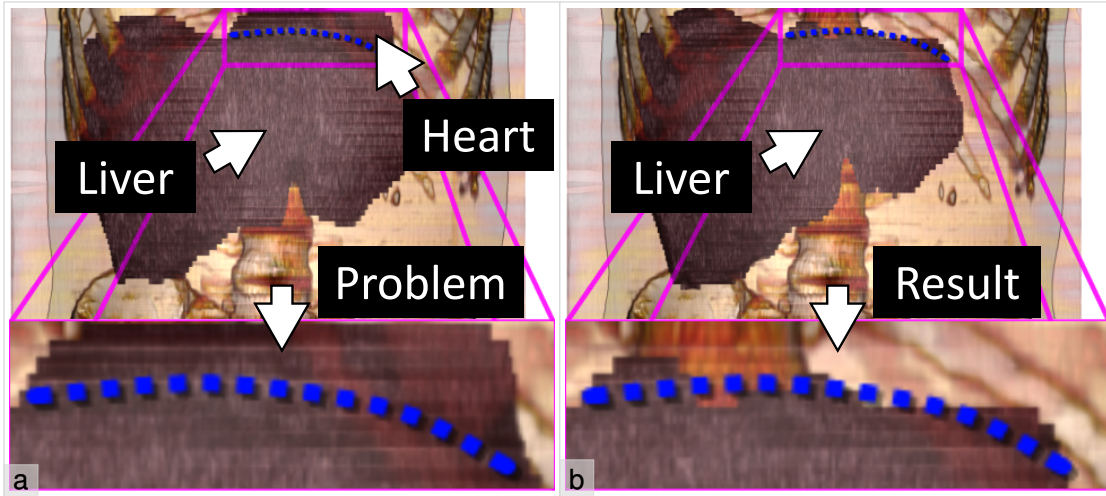
Figure 2.6: Comparison of (a) the initial segmentation and (b) the corrected segmentation of the object. The object is a liver, affected by NAFLD. The initial segmentation has severe over-estimation towards the heart. The blue dotted line shows the desired delineation of the organs.

2.3.2 Quantitative Evaluation

We compare our method with existing approaches in terms of segmentation quality and interaction time. The goal of the SLiver 2007 contest [68] was to find a liver segmentation technique that performs best and minimizes user interaction. At the contest, twenty datasets were provided together with ground truth segmentations done by domain experts. Each dataset is a CTA scan with a slice resolution of 512×512 pixels.

First, the applied segmentation technique provided an initial segmentation for each dataset. Various kinds of pathologies caused over-estimation and under-estimation. Then the segmentation was corrected by the author in ViviSection. We evaluated the quality of the segmentation before and after the correction and measured the time spent on the segmentation and the correction. The quality metric is defined as the Jaccard coefficient J:

$$J(\mathcal{L}, \mathcal{L}_T) = \frac{|\mathcal{L} \cap \mathcal{L}_T|}{|\mathcal{L} \cup \mathcal{L}_T|}, \tag{2.1}$$

where $\mathcal{L}$ is the set of liver voxels in the segmentation, and $\mathcal{L}_T$ is the set of liver voxels in the ground truth. J equals to one if the segmentation completely matches the ground truth. As the segmentation differs more from the ground truth, J decreases. J equals to zero if the segmentation does not contain any voxel from the ground truth. Evaluation results are given in Table 2.2. Figure 2.2 partially illustrates the correction of over-estimation in dataset #13 (a-c), under-estimation in dataset #16 (d-f), and layer defect in dataset #03 (g-i).

Several contest entries used interactive segmentation techniques. Beichel *et al.* [12] proposed a method built upon a graph-cut segmentation and chunk-based and mesh-based refinements.

31

Table 2.2: Measurements for the quantitative evaluation. For each dataset we state: the number of slices, the quality of the initial segmentation J_{segm}, the time of the automatic segmentation T_{segm}, the average calculation time per operation t_{calc}, the number of operations, the quality after the correction J_{corr}, and the correction time T_{corr}. The calculation times are included in the corresponding correction times.

Dataset	Slices	J_{segm}	T_{segm}, sec.	t_{calc}, sec.	Operations	J_{corr}	T_{corr}, sec.
01	183	0.9	106	8.5	3	0.92	81
02	64	0.89	45	4.4	2	0.9	37
03	79	0.82	47	3.8	6	0.9	52
04	212	0.79	118	9.8	4	0.92	106
05	319	0.89	161	13.7	2	0.93	56
06	111	0.88	61	4.7	5	0.92	44
07	251	0.89	154	9.9	4	0.92	108
08	228	0.91	158	15.9	5	0.92	148
09	210	0.9	140	9.3	3	0.92	56
10	191	0.87	126	12.2	1	0.9	38
11	388	0.76	208	19.7	11	0.94	576
12	220	0.89	150	10.7	5	0.95	134
13	145	0.88	96	6.3	6	0.92	115
14	129	0.87	73	3.8	9	0.92	103
15	394	0.91	239	14.4	3	0.95	113
16	151	0.79	118	10.9	4	0.95	89
17	121	0.86	81	5.3	3	0.91	47
18	245	0.8	160	11.8	3	0.95	70
19	335	0.88	215	19.7	2	0.93	58
20	183	0.81	107	5.6	2	0.9	22

Segmentation: $\mathrm{avg}(J_{segm}) = 0.86$ **Correction:** $\mathrm{avg}(J_{corr}) = 0.93$
$\mathrm{sum}(T_{segm}) = 2563$ sec. $\mathrm{sum}(T_{corr}) = 2053$ sec.
$\mathrm{sum}(T_{calc}) = 837$ sec.

The initial segmentation takes thirty minutes on average. For the correction the user spends on average of six minutes to achieve a segmentation quality of 0.95. Beck and Aurich [10] proposed a region-growing-based semi-automatic segmentation method with a "virtual knife" for correcting over-estimation. Under-estimation correction requires new seed points in the region growing procedure. The segmentation and correction time for this method is seven minutes on average for a segmentation quality of 0.93. With our method the average segmentation quality rises from 0.86 to 0.93. The technique only requires a modest amount of time for the correction, *i.e.*, 103 seconds on average, including 42 seconds on average for calculations. For comparison, the fully automatic initial segmentation took 129 seconds on average. For all datasets the segmentation quality is equal or above 0.9. Before the correction the lower bound was 0.76. We conclude that ViviSection allows the user to correct the segmentation for any dataset until a certain quality is

reached (about 0.9). A further increase of the quality is possible only with voxel-level editing, which takes a significant amount of time. Manual segmentations of the entire volumes on a voxel level takes from 25 minutes to 40 minutes per dataset [72]. The time of manual editing on a voxel level strongly depends on the quality of the initial segmentation. The voxel-level editing increases the quality to 0.95-0.99.

2.3.3 Domain Experts Feedback

Three domain experts provided feedback on our method. They corrected liver segmentations from the applied segmentation technique. Both CT and CTA datasets were used.

The first domain expert, a senior clinical radiologist, was familiar with live-wire-based editing to manually obtain liver contours, which takes about seven minutes per dataset for usual cases. He commented that context information available during the editing in 3D is useful for decision making in cases of severe pathologies. He proposed to add a new selection tool like a "brush" operating on the object's surface to speed up interaction in special scenarios. He preferred live-wire-based editing to our scalpel and lasso tools because of better response times and having full control over the correction. However, in his opinion, the case of the severe pathology was easier corrected with our system than with his current software.

The second domain expert, an intern, used in her routine a slice-based editing technique with a manual drawing of liver contours, which takes about twenty minutes per dataset for usual cases. She stated that the selection based on the structural features is sufficiently precise to correct major segmentation defects. To correct small-scale features she suggested a "zoom-in" mechanism. She found the scalpel selection tool very intuitive and easy to use. She also commented that the needle and the lasso tools require more initial training. She also found that a better understanding of the data can be achieved through interactive 3D rendering. The link between 3D rendering and slice views allowed her to select occluded features, although, in her opinion, this also requires additional training. Finally, she stated that response times during the individual operations are appropriate, but response times between operations are too long due to the re-computations.

The third domain expert, a clinical radiologist, used live-surface-based and slice-based techniques for the editing. He commented that a link between the slice views and the 3D rendering is useful for the editing of partially occluded features. The main concern from his point of view are the response times. He liked the interactive preview of changes in both the 3D rendering and the slice views, as it provided him with the ability to fine-tune operations. He preferred our system over live-surface-based editing, because he can easily and efficiently correct major segmentation defects with our method.

2.4 Discussion and Future Work

An issue of our method is the dependency on the underlying skeletonization. If the feature of interest is not represented in the skeleton, our approach cannot correct this as no information about the feature is available. For example, small-scale features of the object cannot be properly detected with thinning-based skeletonization techniques. Experimentally we found that features

smaller than $5 \times 5 \times 5$ voxels in size are missing in the skeleton. They are not editable using "remove part" and "add part" operations. Treating them as layer defects the user can apply "remove layer", "add layer", and "smooth" operations.

In order to solve the issue of missing small-scale features multi-resolution skeletonization could be utilized. The idea is to merge multiple skeletons with details of different scales across different regions of the volume. As the user concentrates on some region of interest, one can create a high-resolution sub-volume of that region and do a skeletonization on it. With a higher resolution the missing features become larger, so they are detected by the skeletonization. The new skeleton substitutes part of the old one, so the skeletons are merged. Then the user has the possibility to edit the features, which were previously unavailable.

Another issue of our method is that self-obstruction and cluttering in the case of complex skeletons make the selection of structural features more difficult. They might be occluded by other features or located inside the object. However, partial occlusion can be effectively dealt with using our lasso selection tool. This is sufficient, for example, for a vessel-segmentation correction as illustrated in Figure 2.7. For future work we propose to select and temporarily hide occluding structural features. Such a solution could reveal completely occluded structural features, like vasculature, tumors, and cysts in the liver.

Cluttering of the skeleton due to noise or a large number of details could result in many small skeleton parts. To alleviate this in our method the user does not work with the skeleton directly, but rather selects a region for editing. The scalpel and the lasso tools treat all underlying structural features in the selected region at once.

Skeleton smoothing and pruning methods can simplify complex skeletons generated from noisy data. Our method can utilize them if the influence zone of each modified skeleton voxel is re-assigned to the appropriate skeleton voxel, which remains after the skeleton simplification. We propose to use the number of voxels in influence zones of each individual skeleton line as a criterion for skeleton pruning.

Our concepts of the skeleton distance field and the influence zones can be adapted to different skeletonization approaches. For the various thinning-based skeletonization approaches, the adaption is trivial, as all thinning-based methods contain thinning iterations in one way or another. Thus, the skeleton distance field can be constructed and, then, the influence zones are computed. In case of the distance-field-based techniques, the underlying distance field can be used as the skeleton distance field. The influence zones can be computed by our Algorithm 2.2 from the resulting skeleton, which is possibly post-processed. If the skeleton is constructed using a Voronoi diagram, the influence zones correspond to individual Voronoi cells. The skeleton distance field is then based on a distance metric that was used for the Voronoi diagram construction. As there is no general definition of a curve skeleton, the adaption of the skeleton distance and influence zones concepts for each curve-skeletonization approach is custom. For the definition by Cornea *et al.* [61, 62], one may trace the influence zones by following the underlying vector field from each skeleton voxel towards object voxels. The skeleton distance field can use the numbers of tracing steps, required to reach each object voxel. With the methods by Reniers *et al.* [59, 63], the skeleton distance field and the influence zones can possibly be generated during the mass

advection process. With such adaptations, the skeleton distance field and influence zones can be computed by a wide variety of the skeletonization techniques.

2.5 Summary

In this chapter, we proposed ViviSection, a novel approach for editing segmentation results based on structural features. For this purpose we introduced the notion of a skeleton distance field, which allows us to detect structural features of the object and to operate on them during a correction step. To support the accurate selection of the features our technique employs influence zones. They clearly delineate the structural features. Our approach uses a set of intuitive selection tools and operations, which allow the user rapid correction of major segmentation defects. We tested the robustness of our method under severe pathological conditions. Our results show that the presented approach can increase the precision and stability of segmentations with little user interaction.

Next, we develop a method for automatically detecting the following types of segmentation defects: O_P, O_L, U_P, and U_L. Our classification in Chapter 1 describes each of these types. The detection method utilizes the introduced skeleton distance field and influence zones in order to enable the analysis of data values at different parts of the object. This way, we provide the user with suggestions on where and how to correct the detected defects.
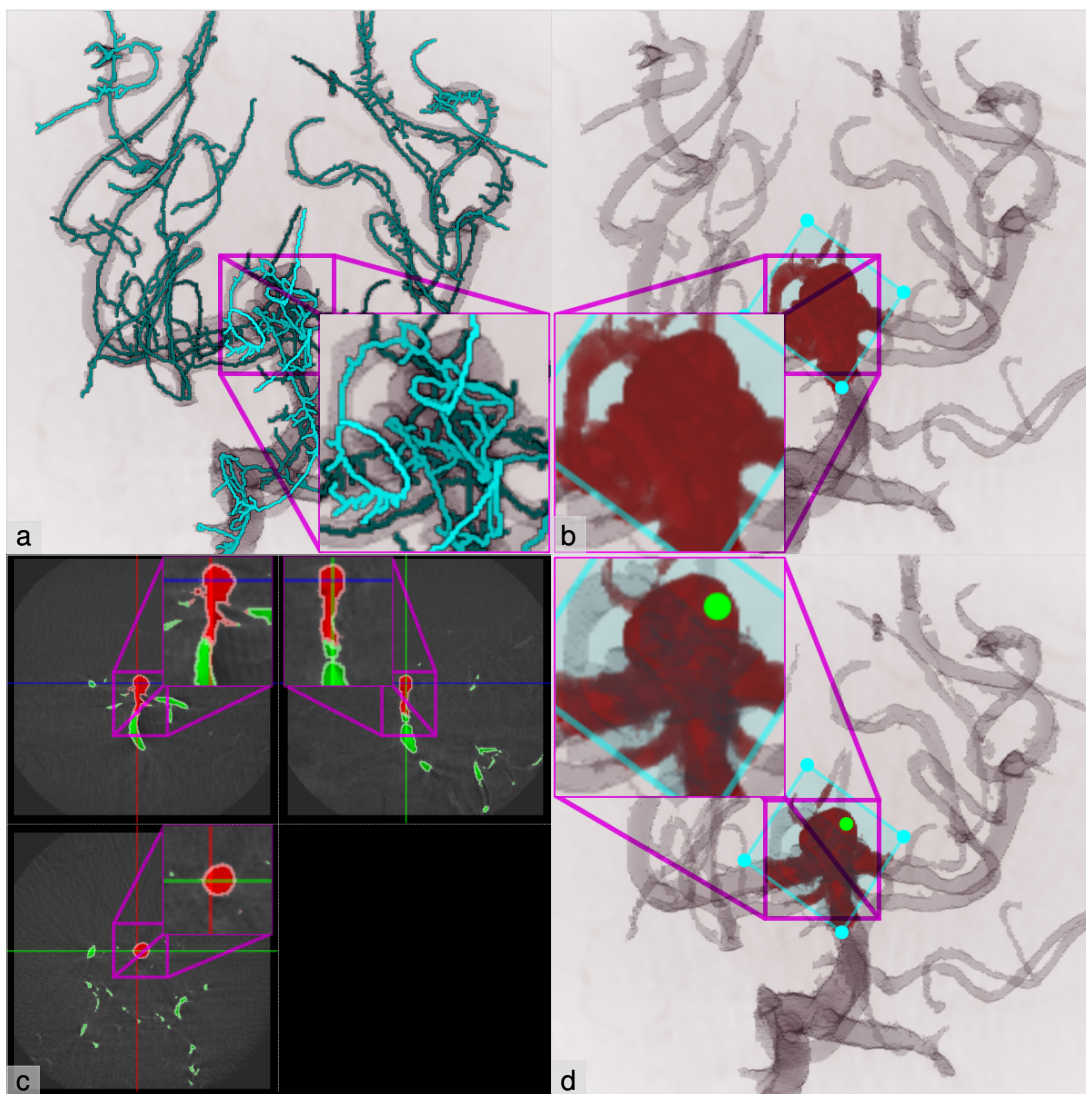
Figure 2.7: Selection of a partially occluded structural feature (vessel aneurysm in MRA): a) skeleton cluttering and self-occlusion, b) direct interaction with the lasso tool fails to select the feature without the occluder, c) a point is selected in the slice views in order to discriminate the feature from the occluder, d) the lasso tool with a link to the point in the slice views (the green circle) allows the user to select the feature.

CHAPTER 3

Guided Volume Editing based on Histogram Dissimilarity

I do not seek. I find.

— Pablo Picasso

Segmentation of volumetric data is an important part of many analysis pipelines, but frequently requires manual inspection and correction. While plenty of volume editing techniques exist, it remains cumbersome and error-prone for the user to find and select appropriate regions for editing. We propose an approach to improve volume editing by detecting potential segmentation defects while considering the underlying structure of the object-of-interest. Our method is based on a novel histogram dissimilarity measure between individual regions, derived from the topological representation introduced in Chapter 2. Based on this information, our interactive system guides the user towards potential defects, provides integrated tools for their inspection, and automatically generates suggestions for their solution.
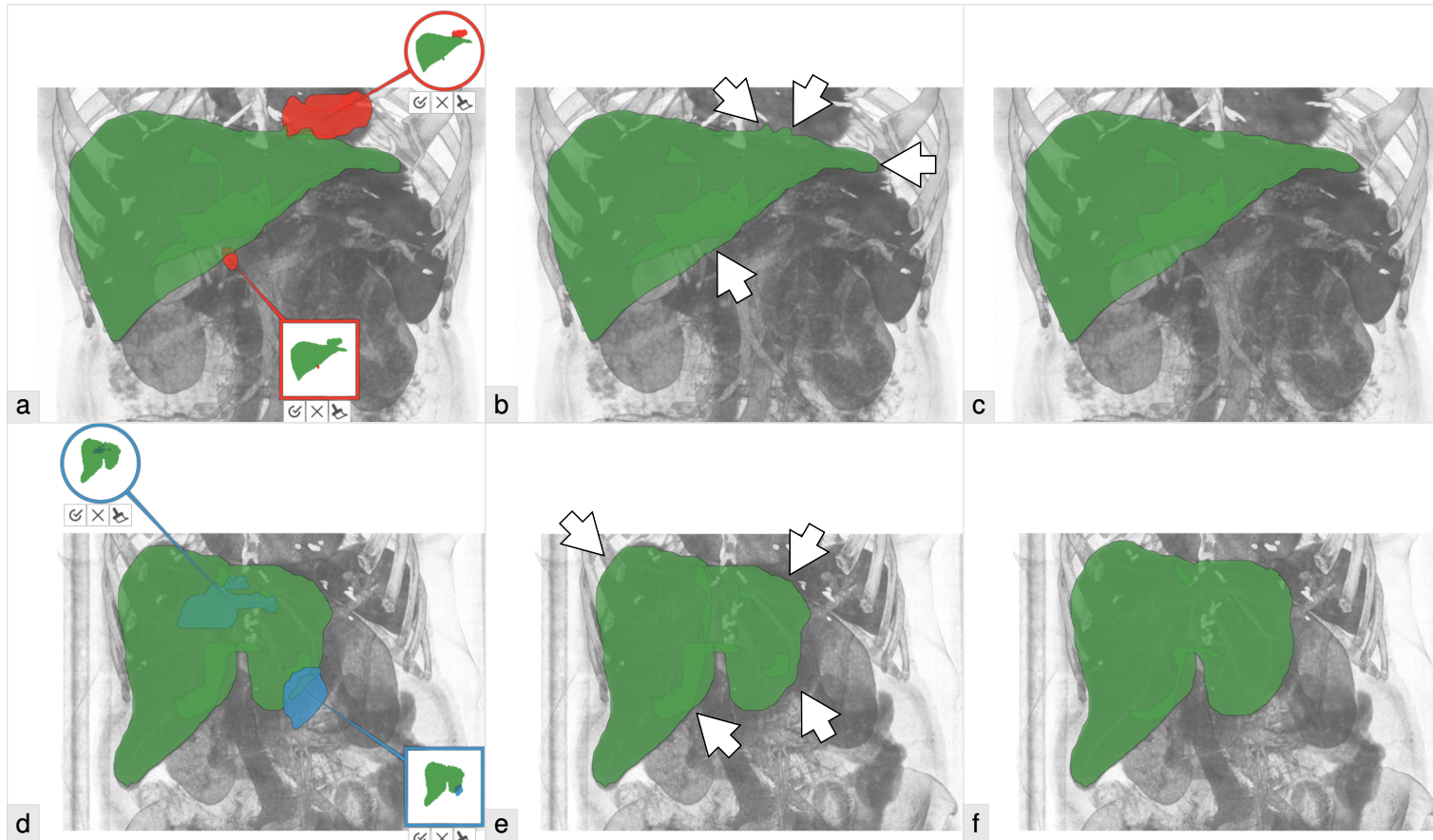
Figure 3.1: Our novel guided volume-editing approach for improving the quality of segmented medical data (Jaccard coefficient J from Equation 2.1). a) Two suggestions to rectify over-estimation defects; initial quality of 0.88, b) after applying the suggestions, segmentation quality of 0.91, c) after applying four more suggestions, segmentation quality of 0.92. d) Two suggestions to fix under-estimation defects; initial quality of 0.8, e) after applying the suggestions, segmentation quality of 0.85, f) after applying six more suggestions, segmentation quality of 0.94.

*M*ANY analysis tasks require the segmentation of volumetric data. Common examples include medical diagnosis and treatment planning, where individual organs or pathologies are labeled in order to compute quantitative information such as diameter or volume, or the segmentation of cell bodies in microscopy data. While there is a plethora of automatic and semi-automatic segmentation algorithms, they are frequently restricted to very specific scenarios and often require manual verification and adjustment. An example of typical segmentation defects in medical imaging is depicted in Figure 3.1. Most commonly, the user coordinates segmentation correction, actively searching for defects and editing them. We propose a novel approach for simplifying and accelerating correction. Our method extracts structural information from any initial segmentation and identifies potential defects automatically. These potential defects can be inspected by the user in 3D using integrated views and corrected by automatically provided suggestions. Our method enables:

- *guidance towards an overall better quality* of the result by suggestions, based on a novel histogram dissimilarity analysis,

- *simple, yet effective volume editing* by reducing and unifying user interaction,

- *structure-awareness* by accounting for the object's shape by means of the skeleton.

3.1 Related Work

Data collection (*e.g.*, into histograms) and aggregation are common ways of describing data values in spatial regions of volume data. The idea is to obtain a discriminative entity that characterizes each such region. Cubes and spheres centered at a voxel are commonly used as regions of aggregation. Generic aggregation functions, which represent the whole region by a single data value, *e.g.*, minimum, maximum and average, are used, for example, on vessel data by Mistelbauer *et al.* [74]. A more general approach is to fit the collected data values to a known model such as a normal distribution. First, the fitting procedure checks whether the collected values correspond to the model by statistical tests and then determines the parameters of the model. A potential problem of such an approach is that the distribution model may be insufficient to describe parts of the object where segmentation defects are present. Haidacher *et al.* [75] propose to collect the data within a growing sphere around each voxel and fit it to a normal distribution. The statistical test they used can distinguish between a region with a single material and a region with multiple materials and stop the growing process accordingly. The statistics, however, become biased at the boundary of the object.

3.2 Method

Motivated by the fact that volume editing is a laborious task with the last percentages of quality consuming most of the time, we propose a technique that unifies the interaction and guides the user through the editing process. The workflow of our technique is depicted in Figure 3.2. We automatically generate suggestions (*correction scenarios*) about where (*correction region*) and
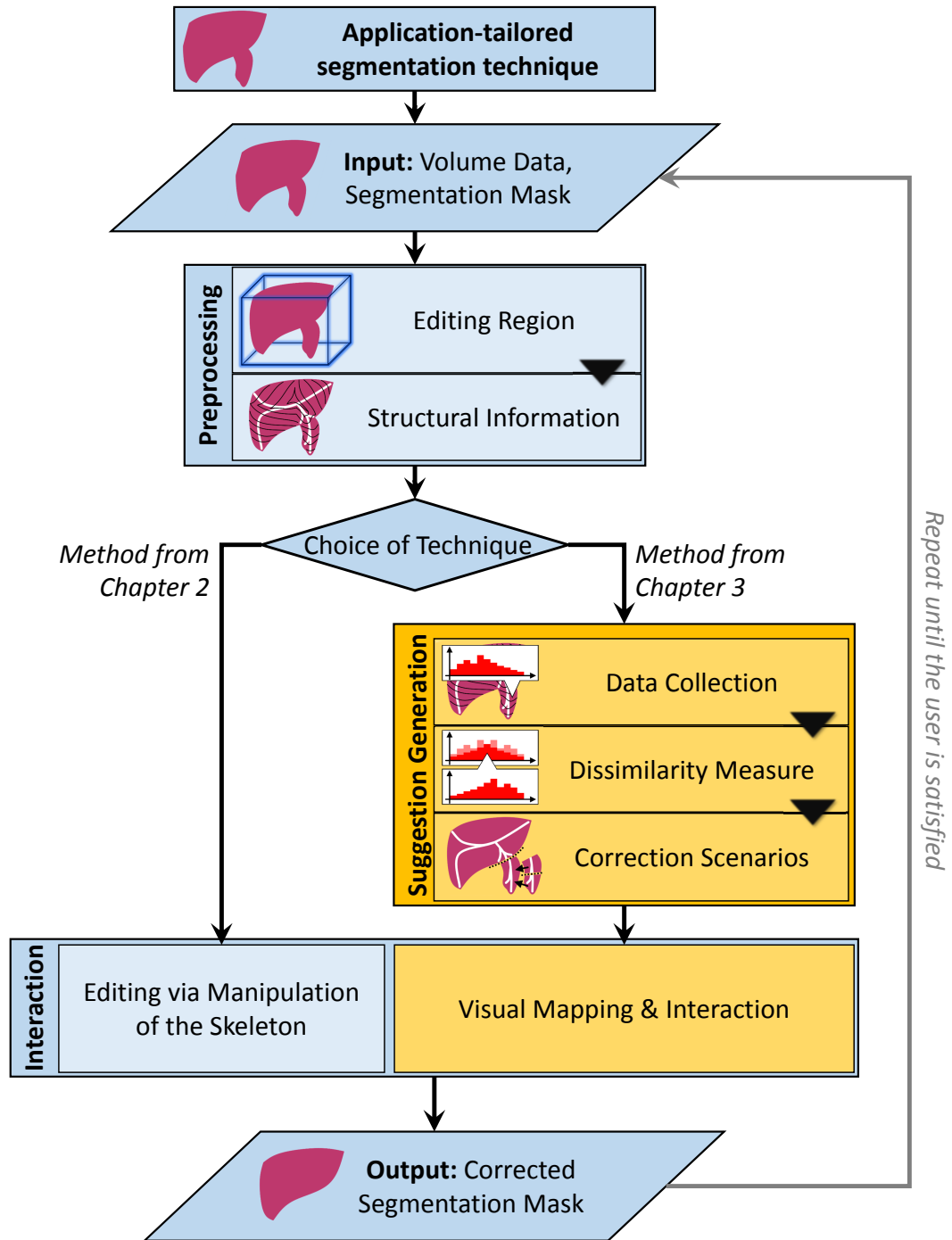
Figure 3.2: Our main contribution is the integration of the suggestion generation and the visual mapping into the workflow of volume editing.

how (*correction operation*) to correct potential segmentation defects. With these suggestions we guide the user towards the desired improved segmentation. An application-tailored segmentation technique, applied prior to the editing process, provides an approximate object segmentation as an input for our technique. We require this segmentation to roughly resemble the correct object in terms of major features. Hence, the basic assumption of our method is that defects in the segmentation are localized (*e.g.*, an over-estimation defect into another organ) and we utilize the fact that the skeleton of the initial segmentation partially corresponds to the shape of the correct object. To simplify the explanation, we restrict our discussion to a binary segmentation, which labels the voxels of a volume as either object or background. Despite this, our approach also supports scenarios where multiple objects are labeled.

Our approach aims to detect inhomogeneities by analyzing data value distributions and incorporates the shape information provided by the initial segmentation. Hence, the method assumes that detectable differences of data values could be the cause of a misclassification. The misclassification can be resolved by incorporating the domain knowledge of the user.

In order to identify them and provide automatic suggestions for their resolution, we analyze changes in the data value distribution along the structure of the initial segmentation using a histogram dissimilarity measure. Based on the nature of these changes, we identify the most likely type of defect and its corresponding correction scenario at multiple levels of detail. Overall, we provide an interactive visual interface that guides the user through the correction process. We now introduce the concepts of our method in detail.

3.2.1 Editing Region

The global *editing region* defines the domain of every subsequent editing operation. For O defects it is the set of all object voxels. To deal with U defects one could be tempted to just choose the whole background as the editing region. However, from a usability perspective this is problematic, as the multitude of structures in the background would generate a large number of possible correction scenarios. Thus, we restrict the editing region to the vicinity of the object by constructing its convex hull and growing the hull away from the object barycenter by a fixed amount of voxels. Removing the object voxels from this set yields the editing region for U defects, which we refer to as the *complement of the object*. Note that very large defects could go beyond this region and would have to be corrected in several editing iterations. We found that a growth by 10% of the maximal dimension of the volume provides a good compromise. It is important that the correction of U defects of the object is conceptually identical to the correction of O defects in its complement; a fact that significantly simplifies the subsequent description of our method. The user chooses the editing region to correct either the object or its complement.

3.2.2 Structural Information

We assume that the object has a certain stereotypical shape (*e.g.*, because the object is a particular organ with known characteristics), so variations in its skeleton can provide valuable information about potential defects. Hence, our approach for determining correction scenarios operates on the skeleton of the editing region. We do not perform any skeleton post-processing, *e.g.*, pruning,

as it may destroy details of the shape, caused by segmentation defects. We adopt the concept of the skeleton distance field and influence zones, introduced in Chapter 2. The *skeleton distance field* provides distances from all non-skeleton voxels to the closest skeleton voxels. For a given skeleton voxel, its associated *influence zone* can be thought of as the set of all non-skeleton voxels that lie closest to this skeleton voxel, similar to a Voronoi region. The editing region is partitioned into influence zones, which emanate orthogonally from the region's skeleton. A phantom dataset is illustrated in Figure 3.3. We denote the influence zone of a skeleton voxel q as $IZ(q)$, the distance of a voxel p from the skeleton as $D_S(p)$, and the largest occurring distance as $D_S^{\max}$.

In our previous technique, described in Chapter 2, the user manipulates the skeleton in order to select defects. The manipulations are converted into selected parts of the object using the influence zones. The correction scenarios alleviate the user from this task, providing a consistent editing experience throughout different objects. Internally, each scenario uses the skeleton distance field and influence zones.

3.2.3   Data Collection

The data values in the *collection regions* are collected into histograms. We implement the collection regions as the influence zones and iso-surfaces of the skeleton distance field (see Chapter 2). This way, the collection regions reflect the structure of the editing region. We make no assumption regarding the underlying distribution of data values and directly compare the histograms to find dissimilarities among them, since they reveal segmentation defects. For all histograms, we choose the same fixed bin size b. As illustrated in Figure 3.3, O defects result in additional structures along the skeleton (O_P) and "orthogonal" to it (O_L). To detect O_P defects, we collect the data values of the influence zone $IZ(p_i)$ of each skeleton voxel p_i into a corresponding histogram $H_1(p_i)$. Additionally, we generate histograms $H_2(p_i, d)$ for each iso-surface with skeleton distance d in $IZ(p_i)$. As shown in Figure 3.3, we expect the H_1 histograms to vary considerably along the skeleton in the presence of O_P defects, while O_L defects would cause an abrupt change of the H_2 histograms at the skeleton distance of the defect. At this point, the histograms are not normalized, as later we apply the summation kernel on them.

3.2.4   Dissimilarity Measure

Using the collected histograms, we now compare data values from different parts of the editing region in order to detect segmentation defects. The comparison is performed by computing dissimilarity measure over histograms. As we are interested in two types of defects O_P and O_L, there are two dissimilarity values that we assign to each skeleton voxel p. In order to detect O_P defects we compute a dissimilarity $\delta_1(p)$ between H_1 histograms. O_L defects are detected by a dissimilarity value $\delta_2(p)$ for H_2 histograms. Each value is given by a *dissimilarity measure* $\Delta(S_H)$, evaluated on a subset S_H of the corresponding histograms. Different types of defects require different subsets S_H in order to be detected. Therefore, we introduce the *dissimilarity detection kernels* for O_P and O_L defects. For O_P defects, the kernel determines influence zones that are compared by their respective H_1 histograms. The kernel for O_L defects defines iso-surfaces of the skeleton distance field for H_2 histogram comparison. Number of compared histograms in each

kernel is controlled by the corresponding kernel *size*. Generation of the subsets S_H by the kernels will be explained in detail later.

We use the histogram intersection metric, a common bin-by-bin histogram comparison metric, equivalent to the L_1 distance. As it requires the histograms to be normalized, we perform this operation on the fly. We denote value of l-th bin of normalized histogram h as $h[l]$. The histogram intersection metric $D_\cap$ between two histograms H_1 and H_2 is defined as follows [76]:

$$D_\cap(H_1, H_2) = 1 - \sum_l \min(H_1[l], H_2[l]). \tag{3.1}$$

Spatial regions that we analyze for dissimilarity of data values usually span several influence zones or iso-surfaces of the skeleton distance field. Each zone and each iso-surface has its own histogram. Therefore, we need to compare more than two histograms at once. In order to do so, we generalize the metric by intersecting all histograms in S_H bin-wise using function $f(S_H, l)$:

$$f(S_H, l) = \min_{h \in S_H} h[l], \tag{3.2}$$

and computing the total intersection area:

$$g(S_H) = \sum_l f(S_H, l). \tag{3.3}$$

The *dissimilarity measure* is defined as follows:

$$\Delta(S_H) = 1 - g(S_H). \tag{3.4}$$

It is normalized, *i.e.*, it yields 0 for a set of identical histograms and 1 for no match between any two compared histograms. Other histogram metrics, such as the Jensen-Shannon divergence, the earth-mover's distance, the Chi-Squared statistic or the Kolmogorov-Smirnov statistic, could be explored. However, we require their generalized versions that can compare more than two histograms at once. A statistical metric, which is adapted to our scenario, requires significant theoretical work and is an interesting direction for future research. Such a metric can be used for the statistical testing of the histogram dissimilarity. Consequently, we could introduce uncertainty information into our pipeline, which is not possible with the current metric Δ.

Sometimes, a single histogram may contain too few values to be representative; hence, larger collection regions are required. For this purpose, we employ an iterative growing process during which we sum up nearby histograms bin-wise, *i.e.*, we apply the summation kernel $\Gamma(H_{in})$:

$$H_{out}[l] \leftarrow \sum_{h \in H_{in}} h[l], \tag{3.5}$$

where H_{in} is the set of input histograms and H_{out} is the resulting histogram. The summation kernel operates with non-normalized histograms only. This does not pose a problem, though, as we perform the normalization on the fly during the dissimilarity computation.

We have the following input parameters for the dissimilarity detection algorithm: the dissimilarity detection kernels *sizes* and the *number* of the growth iterations. Different input data quality, objects, scanning resolutions and modalities complicate the manual specification of these parameters.
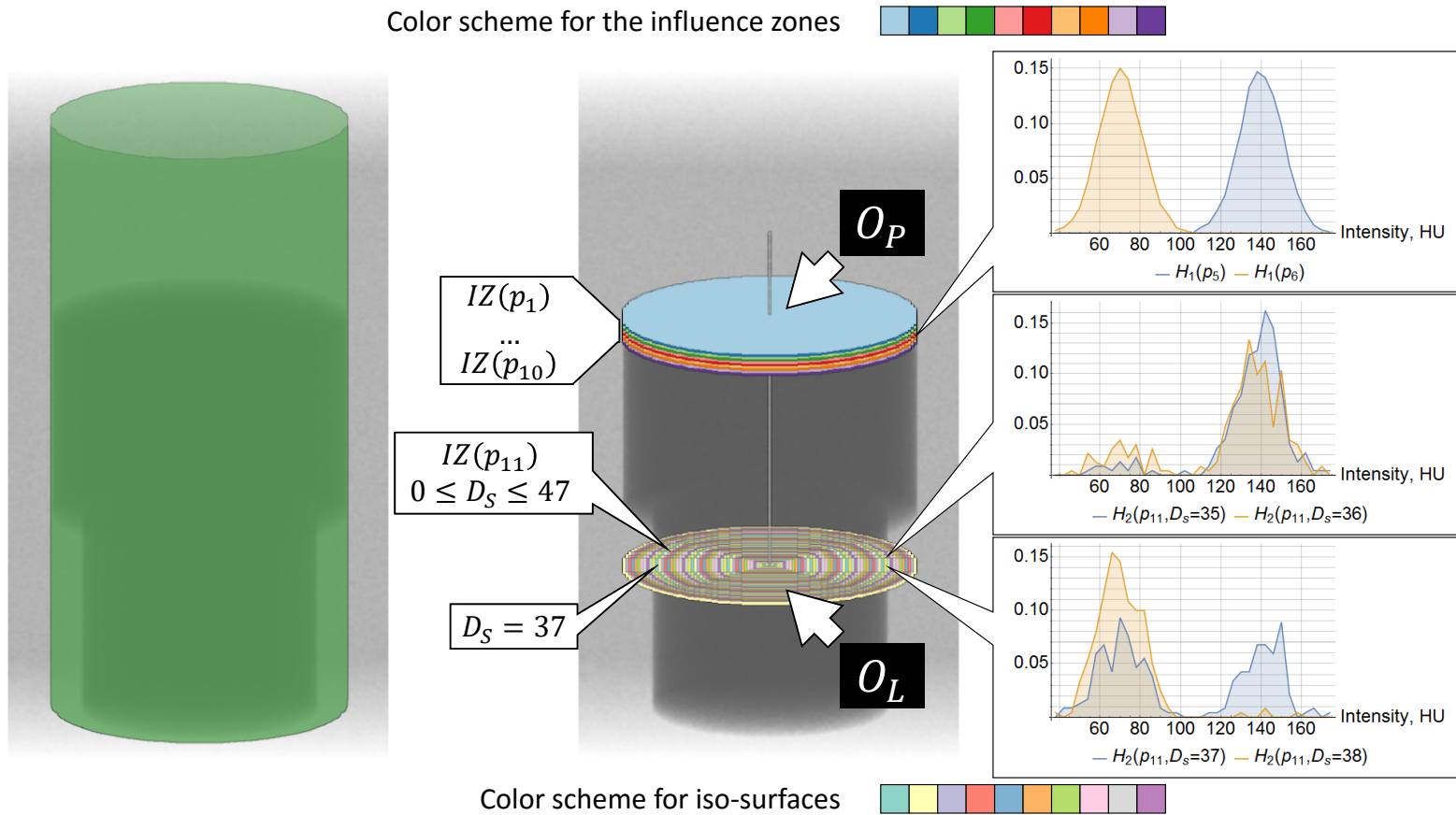
Figure 3.3: Structure-aligned data collection using the skeleton and the influence zones. The histogram $H_1(p_i)$ collects data values in a whole influence zone $IZ(p_i)$ (to identify O_P defects). The histogram $H_2(p_i, j)$ collects data values at a skeleton distance $D_S = j$ in an influence zone $IZ(p_i)$ (to identify O_L defects). The histograms are normalized for illustration purposes. In this figure, the bin size b is set to 4 HU. Two ten-colors schemes from ColorBrewer [77] differentiate the influence zones $IZ(p_1), \ldots, IZ(p_{10})$ and iso-surfaces of the skeleton distance field in $IZ(p_{11})$.

Hence, we propose an automatic tuning procedure. For this, we look at the average $\mathrm{avg}(\delta)$ over all dissimilarity values δ over the whole skeleton, where δ is either δ_1 or δ_2. Serving as a global characterization of the dissimilarities, a value of $\mathrm{avg}(\delta)$ close to 0 indicates a low dissimilarity. Hence, the editing region either has only subtle inhomogeneities and, probably, is free from defects (*case 1*) or the parameter selection suppresses the defect detection (*case 2*). If $\mathrm{avg}(\delta)$ is close to 1, most of the skeleton voxels have a high dissimilarity. This means that either the editing region is very inhomogeneous, inevitably causing false positives (*case 3*), or contribution of noise to the defect detection is too high with the current parameters (*case 4*). The goal of the parameter tuning is to achieve $\mathrm{avg}(\delta) \simeq 0.5$, which discriminates between homogeneous (low dissimilarity, $\delta \rightarrow 0$) and inhomogeneous (high dissimilarity, $\delta \rightarrow 1$) regions with respect to the actual data. We found that a two-phase approach is sufficient for our purposes: in the first phase, the detection kernel size is increased, which increases the number of histograms that are intersected when computing the dissimilarity measures $\Delta(S_H)$. In the second phase, we sum neighboring histograms, enlarging the collection regions. Note that $\mathrm{avg}(\delta)$ monotonically increases in the first phase, whereas it monotonically decreases in the second phase.

For the dissimilarity $\delta_1(p)$ (for O_P defect detection), we form the set S_H by taking histograms H_1. These histograms collect data values in the influence zones of the skeleton voxels. We analyze the influence zones of nearby skeleton voxels with a maximum distance of R_1 to p:

$$\delta_1(p) = \Delta\left(S_H \leftarrow \{H_1(q) \,|\, D(p,q) \leq R_1\}\right). \tag{3.6}$$

In the first phase, we grow R_1 from its starting value 1 in steps of 1 until we hit the limit (10 in our case) or the kernel covers more than 3% of the skeleton voxels or $\mathrm{avg}(\delta_1) \simeq 0.75$, *i.e.*, the midpoint between the target value 0.5 and the maximum 1. The second condition stops the first phase if the kernel becomes too large with respect to the editing region. A threshold of 3% was found during the segmentation editing of phantom and real-world datasets (see Figures 1.1, 1.3). The third condition terminates the first phase if the intermediate value of 0.75 is reached. The reason why the intermediate value is higher than the target value 0.5 lies in the fact that during the second phase the dissimilarity values monotonically decrease. In the second phase, we grow the collection regions using histogram summation between the influence zones:

$$H_1(p) \leftarrow \Gamma\left(H_{in} \leftarrow \{H_1(q) \,|\, D(p,q) \leq 1\}\right), \tag{3.7}$$

until $\mathrm{avg}(\delta_1) \simeq 0.5$, or the number of growth iterations N_G equals to R_1. This limit is due to the fact that the dissimilarity detection kernel with size R_1 should be larger than the grown collection regions of size N_G. From the signal-processing domain perspective, feature detection kernel should be larger than filtering kernel.

The dissimilarity $\delta_2(p)$ is calculated by putting H_2 histograms into the set S_H. These histograms contain data values from iso-surfaces of the skeleton distance field in the influence zone $IZ(p)$. Therefore, the parameter tuning for the dissimilarity $\delta_2(p)$ is slightly different from the case of $\delta_1(p)$. First, the histograms $H_2(\cdot, 0)$, $H_2(\cdot, 1)$, $H_2(\cdot, 2)$ are removed from S_H due to their

insufficient information content (too few collected data values). The first phase is a single step in which the maximal kernel size is chosen by computing $\delta_2(p)$ over *all* skeleton distances:

$$\delta_2(p) = \Delta\left(S_H \leftarrow \{H_2(p,i)\,|\,3 \leq i \leq D_S^{\max}\}\right). \tag{3.8}$$

By summing histograms from neighboring iso-surfaces:

$$H_2(p,j) \leftarrow \Gamma\left(H_{in} \leftarrow \{H_2(p,i)\,|\,|j-i| \leq 1\}\right), \tag{3.9}$$

the second phase is continued until $\mathrm{avg}(\delta_2) \simeq 0.5$ or the iteration limit (again 10) is reached. Finally, Laplacian sharpening of δ_2 is performed along the skeleton to locate the borders of O_L defects between the influence zones.

Choosing a limit of 10 iterations, *i.e.*, a maximal kernel size of 11 for δ_1 and up to 10 growth iterations proved sufficient for all real-world datasets presented in our results, although the actual tuning process typically uses fewer iterations. Note that these limits guarantee termination of the tuning procedure for *cases 1* and *3*, whereas the correct behavior in *cases 2* and *4* is ensured by the monotonicity of the two phases with respect to the increase and decrease of $\mathrm{avg}(\delta)$.

3.2.5 Correction Scenarios

Having calculated the dissimilarity values $\delta_1(p)$ and $\delta_2(p)$ for each skeleton voxel p, we finally generate the *correction regions*. Each such region constitutes a set of voxels on which a correction operation will be applied. One could be tempted to just use thresholding on the dissimilarity values, but this would lead to a binary decision. We employ a hierarchical watershed transformation, following the work of Hahn and Peitgen [27]. The transformation is applied on the dissimilarity values δ_1 and δ_2. This way, we represent the editing region at hands as a hierarchy of spatial regions with the varying level of detail. As a result, we support users with a variety of suggestions and smooth transitions between the levels of detail, thus offering more possibilities to fine-tune the results. Our approach has similarities to the work of Schultz *et al.* [78], which uses the watershed transformation over an anisotropy measure to delineate object features in Diffusion Tensor Magnetic Resonance Imaging (MRI) data. However, we apply the watershed transformation on the dissimilarity values, derived from scalar spatial data.

The watershed transformation splits the spatial data with *watersheds*, placed at the ridges of the data values. Connected components of the data that lie between the watersheds are called *basins*. Applied on the skeleton voxels with δ_1 dissimilarities, the transformation places the watersheds at the ridges of δ_1 values and splits the skeleton into the basins. The basins are connected sets of the skeleton voxels. The correction regions are created by propagating the basins towards the editing region by means of the influence zones, *i.e.*, a basin B is expanded into a correction region R as follows:

$$R = \bigcup_{p \in B} IZ(p), \tag{3.10}$$

where $IZ(p)$ denotes the influence zone of the skeleton voxel p. These correction regions are the O_P and U_P defects and the correctly classified parts of the editing region. The basins are illustrated

in Figure 3.4b. The watershed transformation applied on the skeleton voxels with δ_2 values yields the basins that expand to the correction regions for the O_L and U_L defects. Figure 3.4d shows the corresponding basins.

Having generated a usually large number of correction regions for various defects, we proceed with the construction of a hierarchy. The main idea is that adjacent correction regions, which exhibit the same dissimilarities and correct the same type of defects, can be grouped together. This is the case in the presence of severe defects, which span several influence zones and lead to a multitude of correction scenarios. The resulting hierarchy of correction regions represents the editing region at hands with different levels of detail. As the most detailed level we use the correction regions that are computed from the initial basins before the hierarchy construction.

To simplify the adjacency check of correction regions, we add each watershed to every basin it delineates. Successively two basins are merged in each iteration, so that a new basin becomes a new correction region, which replaces two parental regions from the previous level of detail. To find the merged basins we use the following distance metric D_W:

$$D_W(B_1, B_2) = L_W(B_1, B_2) - \min(L_B(B_1), L_B(B_2)), \tag{3.11}$$

$$L_W(B_1, B_2) = \min_{r \in B_1 \cap B_2} \delta(r), \tag{3.12}$$

$$L_B(B_i) = \min_{p \in B_i} \delta(p), \tag{3.13}$$

$$\delta(p) = \begin{cases} \delta_1(p), & \text{for } O_P \text{ and } U_P \text{ defects,} \\ \delta_2(p), & \text{for } O_L \text{ and } U_L \text{ defects,} \end{cases} \tag{3.14}$$

where B_1, B_2 are two basins, $L_W(B_1, B_2)$ is the watershed level between them, and $L_B(B_i)$ is the bottom level of B_i. Non-adjacent basins ($B_1 \cap B_2 = \emptyset$) cannot be merged. We always merge the pair of basins with minimal distance D_W. The merging stops if there are no more adjacent basins. The metric D_W guarantees that we merge basins with the smallest relative watershed level first, *i.e.*, dissimilarity increment between merged basins is the smallest among all pairs. This leads to merging basins with similar ranges of dissimilarity first. The hierarchy of the correction scenarios is illustrated in the central and the right skeleton views of Figures 3.4b,d.

For each correction region we select a suitable operation to correct the potential defect at this region. For O_P defects, it is sufficient to simply remove the correction region from the editing region, as shown in Figure 3.4c. For O_L defects, the surface layer of the correction region is peeled away from the editing region, as illustrated in Figure 3.4e. The thickness of the peeled layer is fixed at two voxels. We obtained this thickness value empirically.

Each level of the hierarchy represents the editing region at a certain level of detail, determined as the ratio of the number of the correction scenarios at this level to the number of the initially created correction scenarios. This level of detail approach provides the user with a quality control of the correction scenarios. Low levels provide coarse corrections, affecting a large number of voxels, but leaving certain defects. High levels offer detailed correction scenarios, which affect only small local defects, but provide the required fine-tuning to reach an acceptable quality. Two different levels of detail are illustrated in Figure 3.5. Basins that compose these two levels are shown in Figures 3.5b,c. Corresponding correction scenarios are demonstrated in Figures 3.5d,f.

Over-estimation part defect (O_P)

Under-estimation part defect (U_P)

Over-estimation layer defect (O_L)

Under-estimation layer defect (U_L)

Color scheme for correction scenarios
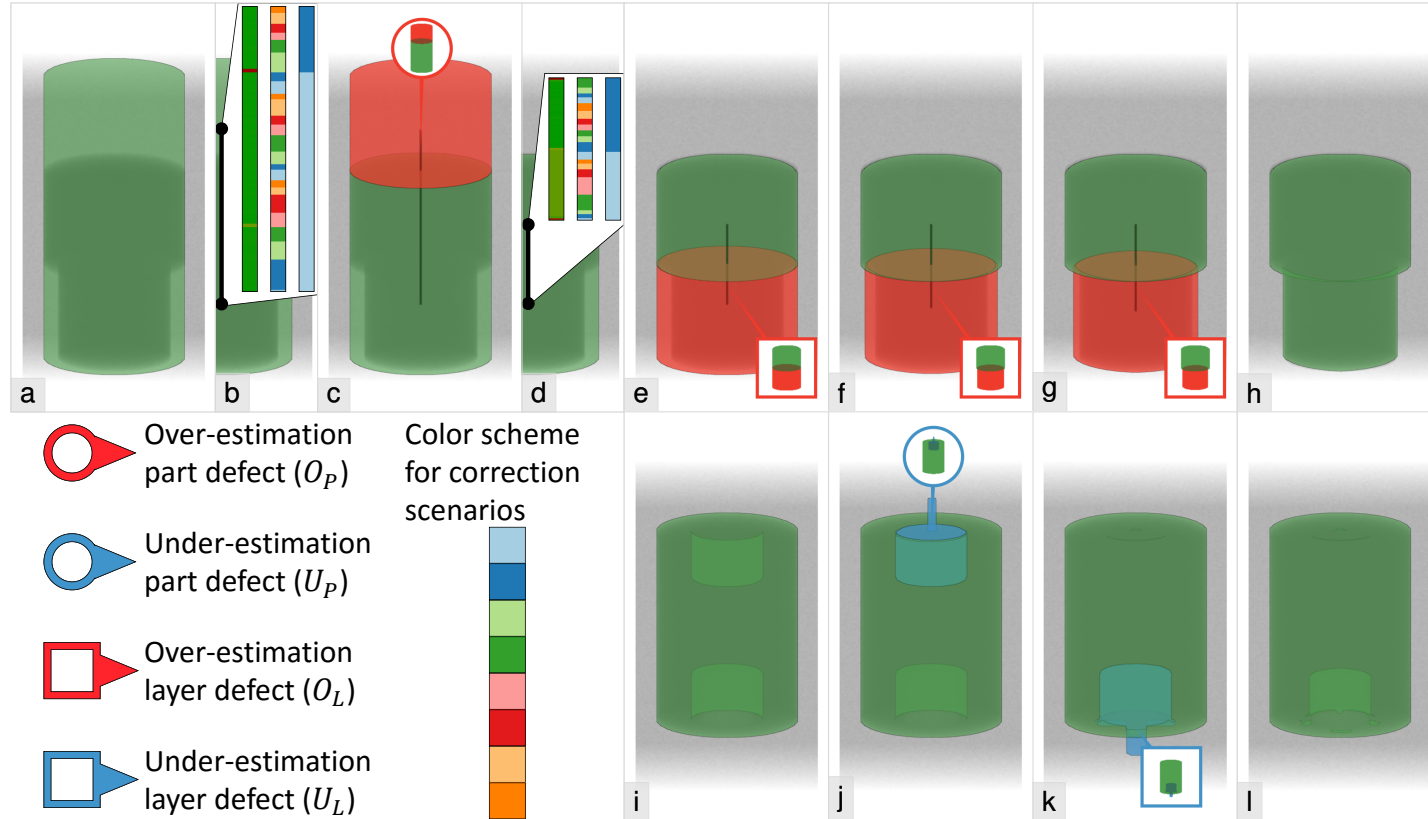
Figure 3.4: Smart volume editing for the phantom datasets of Figure 1.3: a) the initial segmentation of the phantom object #1 with two defects, b) skeleton view: the dissimilarity values δ_1 (green to red transition for $[0;1]$), the correction scenarios at the finest level of detail and at a level of detail, used in (c), c) the correction scenario for the O_P defect, d) skeleton view: the dissimilarity values δ_2, the correction scenarios at the finest level of detail and at a level of detail, used in (e), e-g) the correction scenarios for the O_L defect, h) the correct segmentation of the object #1, i) the initial segmentation of the phantom object #2 with two defects, j) the correction scenario for the U_P defect, k) the correction scenario for the U_L defect, l) the correct segmentation of the object #2. In skeleton views the correction scenarios are encoded with eight-colors scheme from ColorBrewer [77].

3.2.6 Visual Mapping and Interaction

We guide the user through the correction process by visually conveying the correction scenarios with glyphs (Figure 3.4). The segmentation mask of the object-of-interest is displayed as a surface in 3D space, together with silhouettes to enhance shape perception. The surface is generated by the Marching Tetrahedrons algorithm [41]. The context around the object is visualized by Direct Volume Rendering (DVR). We desaturate the colors of the context in order to draw more attention to the object and the correction scenarios.

We only display glyphs for the correction scenarios at the current level of detail. Each glyph connects the projection of the correction region's barycenter (the anchor point) with a miniature that shows a Maximum Intensity Projection (MIP) rendering of the suggested changes. The background, the object, and the changes that the corresponding correction scenario suggests are mapped to data values of 0, 0.5, and 1 respectively. A color map is then applied. In the MIP rendering, the background and the object do not occlude the suggested changes. Similar to Ropinski *et al.* [79], we use a boundary layout to arrange the miniatures around the viewing area, so that the object and the suggested correction region remain unoccluded. We modulate the opacities of the glyphs with the relative depth of their anchor points. We use color-coding to differentiate between editing the object (red) and its complement (blue). To quickly visually convey our two types of defects (P, L), we encode them into the shape of the glyph (Figure 3.4).

We propose the following correction workflow, illustrated in Figures 3.5d-g. First, the user operates on the 3D rendering, finding the appropriate correction scenario. Initially, a coarse level of detail is set. If the user hovers over the editing region, only glyphs representing the corresponding correction scenarios show up. If only a single correction scenario is suggested, it can be selected with a single mouse click. If the user hovers over a glyph, the boundary surface of the respective correction region is highlighted in order to provide a visual feedback. The correction scenario can also be selected by clicking on its glyph. Whenever a correction scenario is selected, a preview is shown to the user. Additionally, an integrated oblique slice view is provided at the anchor point of the selected scenario. The user can quickly analyze the results and, in case of doubt, refine the selection by scrolling with the mouse wheel in order to adjust the level of detail. The glyphs of the requested level of detail are displayed instantly. If the user selects a correction scenario and then adjusts the level of detail, we display only descendants (at finer levels of detail) or ancestors (at coarser levels of detail) of the selected scenario. This offers the user the possibility to find the desired correction at a suitable level of detail. At any time, the user only gets a small number of suggestions (up to 15) to avoid cluttering. To continue, the user applies the selected correction scenario, and the pipeline is repeated until the user is satisfied with quality of the segmentation mask.

While the correction scenarios enable the efficient resolution of defects at multiple levels of detail, some segmentations may still require fine-grained control at the voxel level. This is the case for defects that partially occupy influence zones. Hence, our system provides an integrated oblique slice view as the highest level of detail, which allows users to inspect and edit minor local deficiencies of the selected scenario (Figure 3.6). Even if a suggested correction is not perfect in terms of segmentation quality, the user still has the ability to fine-tune the resulting
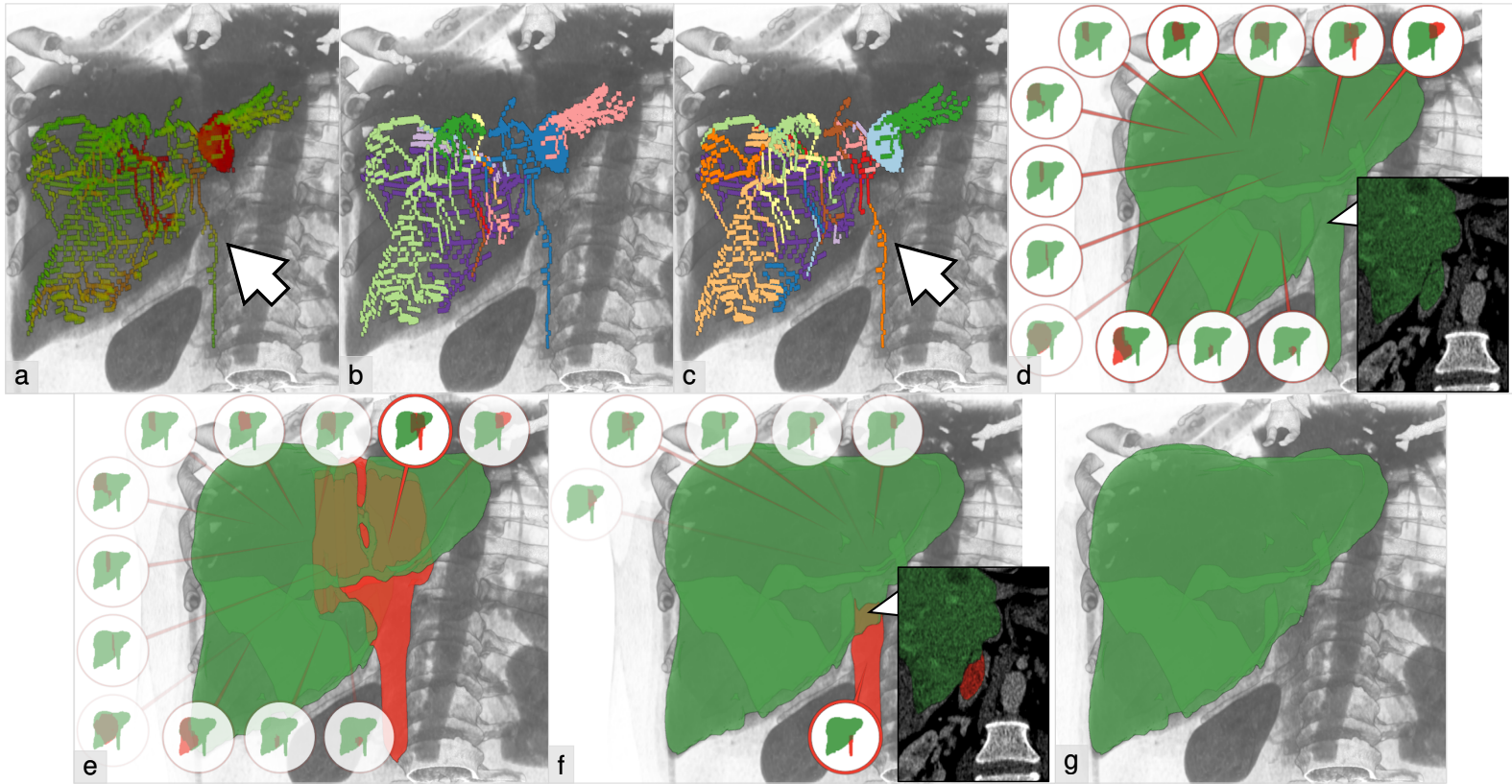
Figure 3.5: a) The dissimilarity values δ_1 (green to red transition for $[0;1]$), b) Correction scenarios at a coarse level of detail, c) A finer level of detail with the correction scenario, which fixes the defect. The user interaction with our proposed technique: d) the initial coarse level of detail, e) the user selects a correction scenario in the coarse level of detail, f) the user explores the finer level of detail and selects the correction scenario, which fixes the defect, g) after applying the selected correction scenario. Each correction scenario is depicted by a single glyph.

segmentation while benefiting from the guidance provided by our approach. To aid the user during the interaction, we allow panning, rotating, and scaling in the 3D view. Miniature 3D renderings are instantly updated in case of rotations. Panning and scaling have no effect on the miniatures, as they are automatically centered and scaled.

3.3 Implementation

Our system is implemented in C#, using Intel TBB (C++, CPU) and DirectX 11 Compute Shaders (HLSL, GPU) for parallelization. For calculating the complement of the object we apply the QuickHull algorithm by Barber *et al.* [80] to construct the convex hull. The implementation is based on the work of Sehnal [81] and runs on the CPU in a single thread. For the skeletonization we employ the algorithm by Lee *et al.* [44]. The skeleton distance field and the influence zones are calculated using the Algorithms 2.1, 2.2. Between individual correction operations all data is recomputed, as all steps of the pipeline depend on the skeleton, which is a global characteristic of the corresponding editing region. The performance was estimated on 20 CTA datasets with a slice resolution of 512×512 pixels and the number of slices ranged from 64 to 512. We used an Intel Core i7-2600K 3.4 GHz CPU with 16 GB of RAM and an NVidia GeForce 680 GPU. The pre-calculation time ranged from 1.8 seconds to 9.1 seconds with an average of 4.8 seconds for correcting the object and from 2.8 seconds to 10.4 seconds with an average of 5.5 seconds for correcting its complement. The dissimilarity analysis took most of the time. Even though we parallelize it on the CPU, a GPU implementation would further improve the performance of our system.

3.4 Results and Discussion

The only free parameter of our pipeline is the **bin size** b of the histograms. We conducted our experiments on an Intel Core i7-2600K 3.4 GHz CPU with 16 GB of RAM and an NVidia GeForce 680 GPU in order to analyze the correction results with different bin sizes. The bin size, which we use for the histograms, was determined during the correction of four abdominal CTA datasets. The objects had data values from -50 HU to 250 HU. The slice resolution was 512×512 pixels, and the number of slices ranged from 128 to 384. With different bin sizes we search for a certain correction scenario in each dataset. The bin sizes tried are 1, 2, 4, 8, 16 HU. We checked corrections of all four types of defects, each on a specific dataset. Our technique detected the defects properly for every bin size and provided appropriate correction scenarios. We suggest $b = 4$ HU to balance computation time, memory usage and a small number of level of detail adjustments (see Figure 3.7).

3.4.1 Robustness against Low Contrast between Object and Defects

To demonstrate the **robustness** of our technique with respect to the **low contrast** between object and defects, we conducted the following two tests on dissimilarity δ_1:
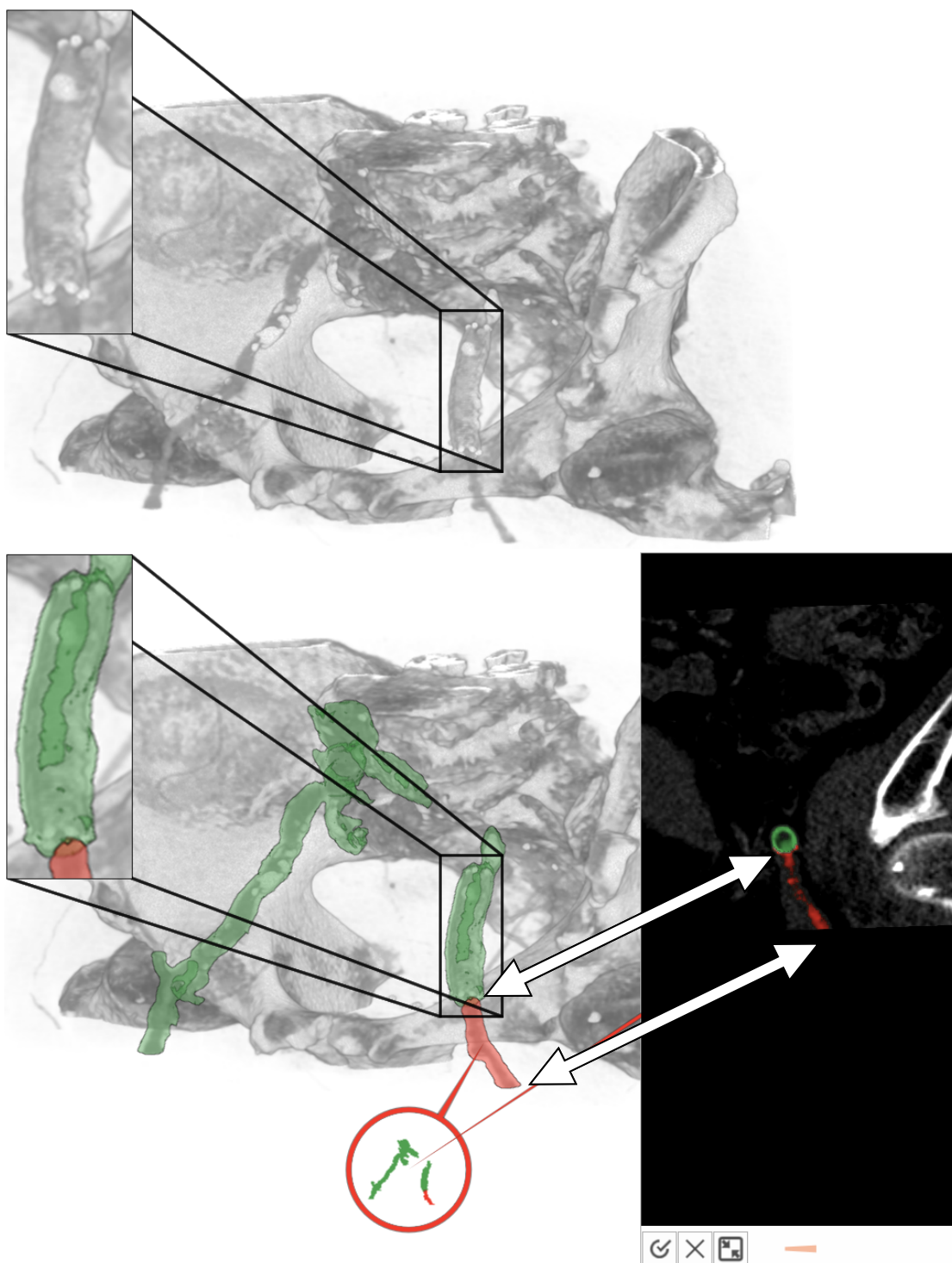
Figure 3.6: Our method correctly detects vascular occlusion in Computed Tomography Angiography (CTA) data.
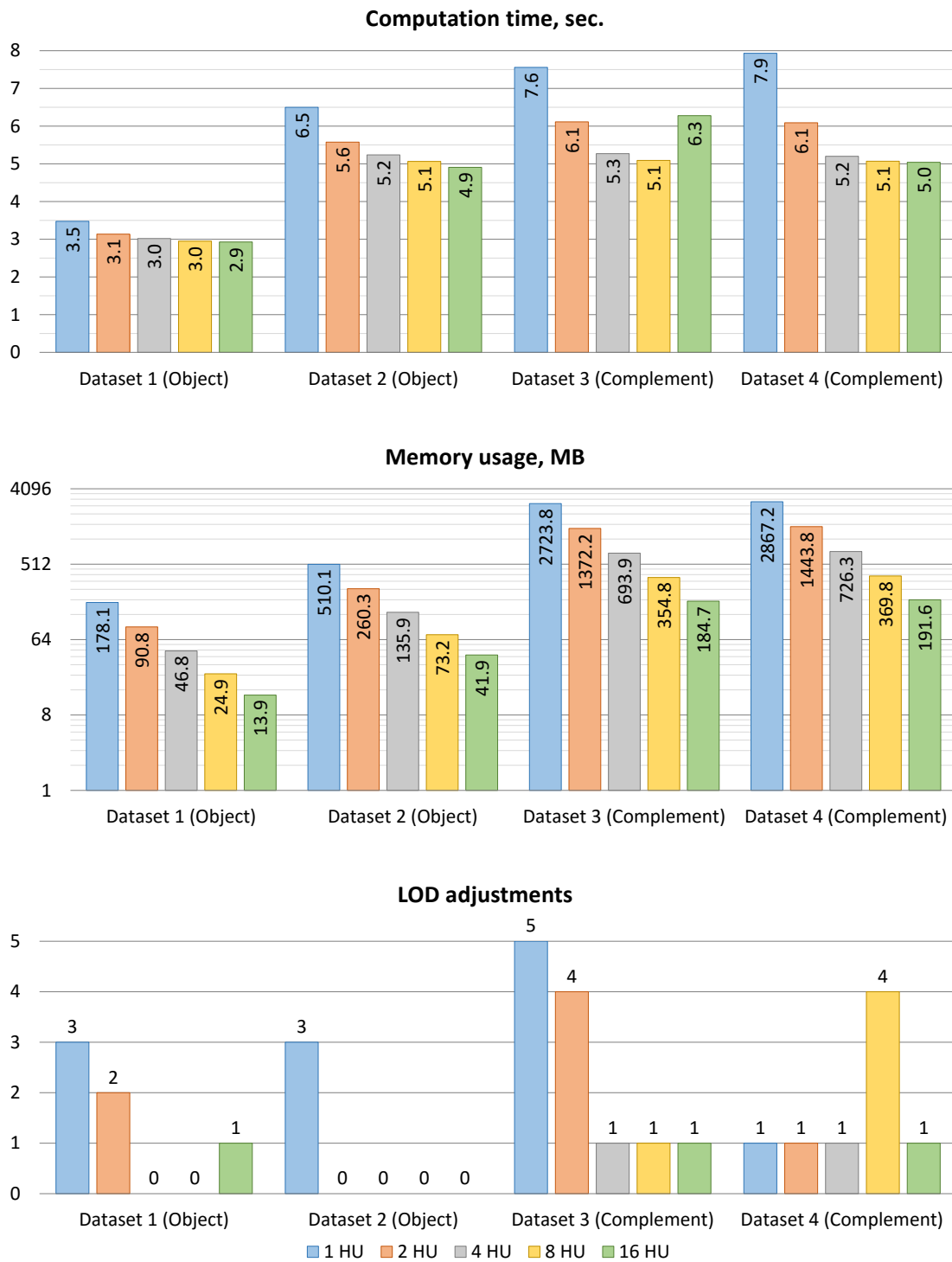
Figure 3.7: Analysis of computation time, memory usage and level of detail adjustments with respect to different bin sizes.

1. two objects with the same data value distribution are split by a one-voxel thick wall between them, which has a different data value distribution

2. two objects with two different data value distributions are located next to each other

The idea behind the tests is to measure the dissimilarity along the skeleton with respect to the varying data value distributions. In the first test, we vary the difference between the means of the objects' distributions and the wall's distribution. In the second test, we vary the difference between the means of two objects' distributions. Each distribution is a Gaussian distribution with standard deviation $\sigma = \dfrac{32}{3}$ HU. We require that the data value distributions differ, as otherwise the delineation of the objects cannot be achieved by data values alone.

Test 1. The test objects are depicted in Figure 3.8. The parameters are $\mu_1 = 70$ HU, $\mu_2 = \mu_1 + \Delta d_1$. The parameter Δd_1 is varied as follows: 1, 2, 4, 8, 16, and 32 HU. With $\Delta d_1 \geq 2$ HU, our technique can clearly separate the two objects, as demonstrated in Figure 3.8. For $\Delta d_1 = 1$ HU, the peak dissimilarity value is not large enough to provide a correct delineation, as demonstrated in Figure 3.9.

Test 2. The test objects are depicted in Figure 3.10. The parameters are $\mu_4 = 70$ HU, $\mu_3 = \mu_4 + \Delta d_2$. The parameter Δd_2 is varied as follows:: 1, 2, 4, 8, 16, and 32 HU. With $\Delta d_2 \geq 2$ HU, our technique can clearly separate the two objects, as demonstrated in Figure 3.10. For $\Delta d_2 = 1$ HU, the peak dissimilarity value is not large enough to provide a correct delineation, as demonstrated in Figure 3.11.

3.4.2 Robustness again Noise

Robustness against **noise** is investigated with the following test. One CTA dataset of abdominal region is segmented by region-growing algorithm with watershed transformation by intensity and gradient criteria. The dataset has 145 slices; each slice has resolution of 512×512 pixels. The segmentation mask contains liver with over-estimation defect at vascular structures. Artificial noise with a Gaussian distribution that has a mean $\mu = 0$ HU and varying standard deviation σ from Table 3.1 is added to the original volume data. Although the model of the statistical error in the detected photons count is Poisson noise, there are more random processes involved in the scanning process [82]. Moreover, the noise occurs in the projected data, which is then back-projected to the actual volume space. Therefore, an accurate noise simulation would require a significant effort, which is out of the scope of this chapter, and a Gaussian distribution is a reasonable approximation for Computed Tomography (CT) data [83].

The segmentation mask is generated anew for each σ. The test aims at correcting the particular over-estimation defect at the vascular structures. The defect is depicted in Figure 3.12. Our technique should detect the defect and provide the possibility to correct it. The Peak Signal-to-Noise Ratio (PSNR) was used to evaluate the degree of artificial noise. We used 2048 HU as a maximal value (upper range for 12-bit integer values). For the slice views, we use the following windowing function: WL/WW=100/300 HU. The segmentation method failed to operate for
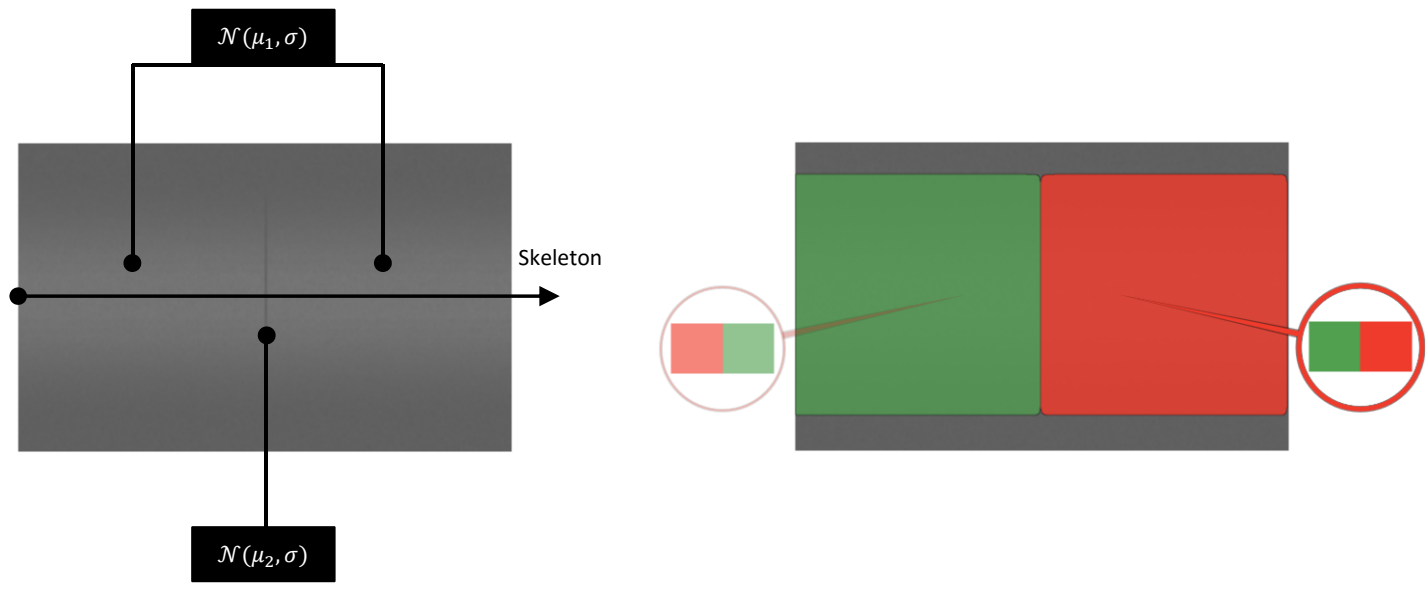
Figure 3.8: Test 1: structure of the test objects and the editing result for the case of $\Delta d_1 = 2$ HU.
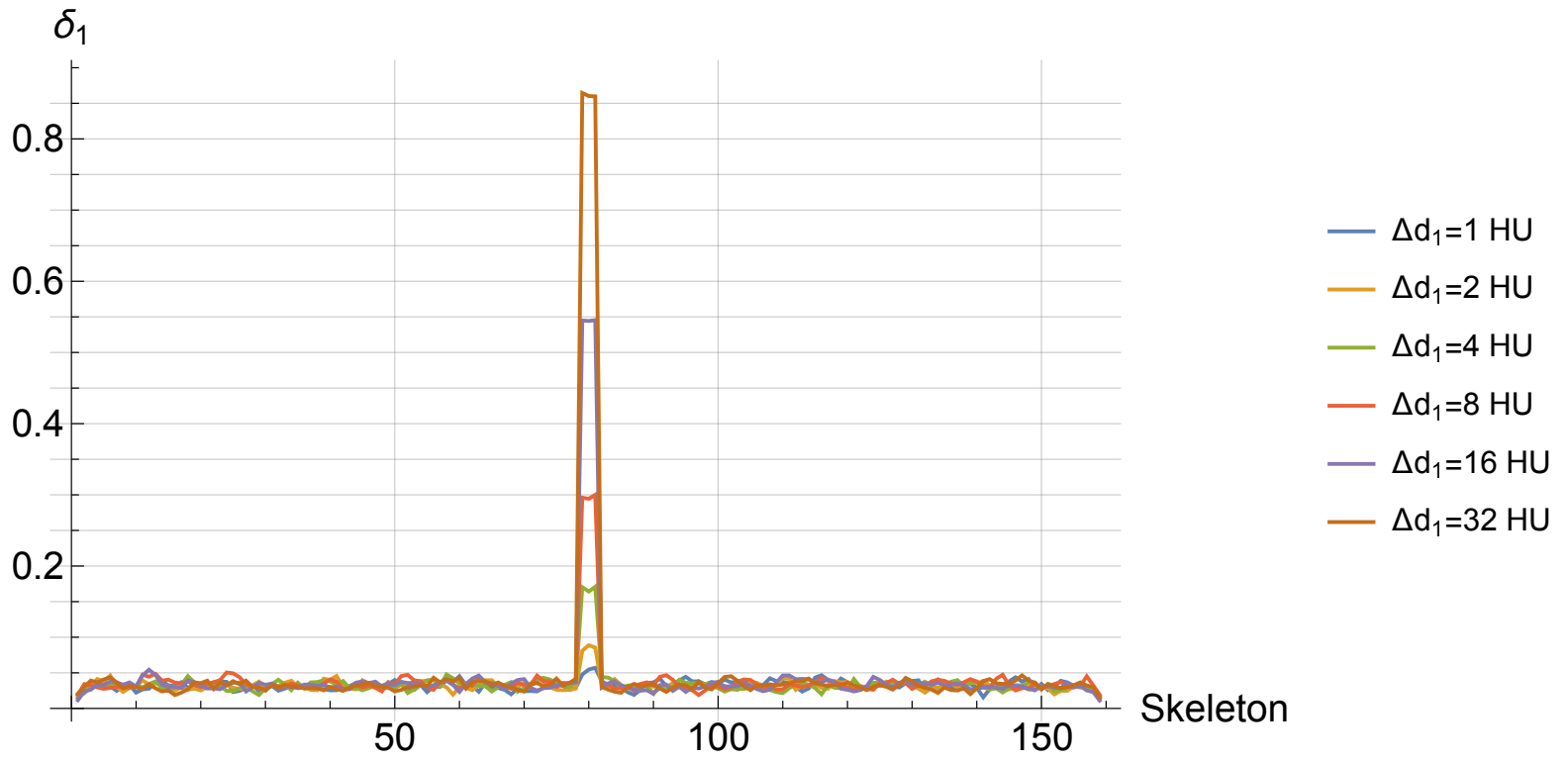
Figure 3.9: Test 1: dissimilarity values δ_1 along the skeleton with respect to varying $\Delta d_1 = |\mu_1 - \mu_2|$.
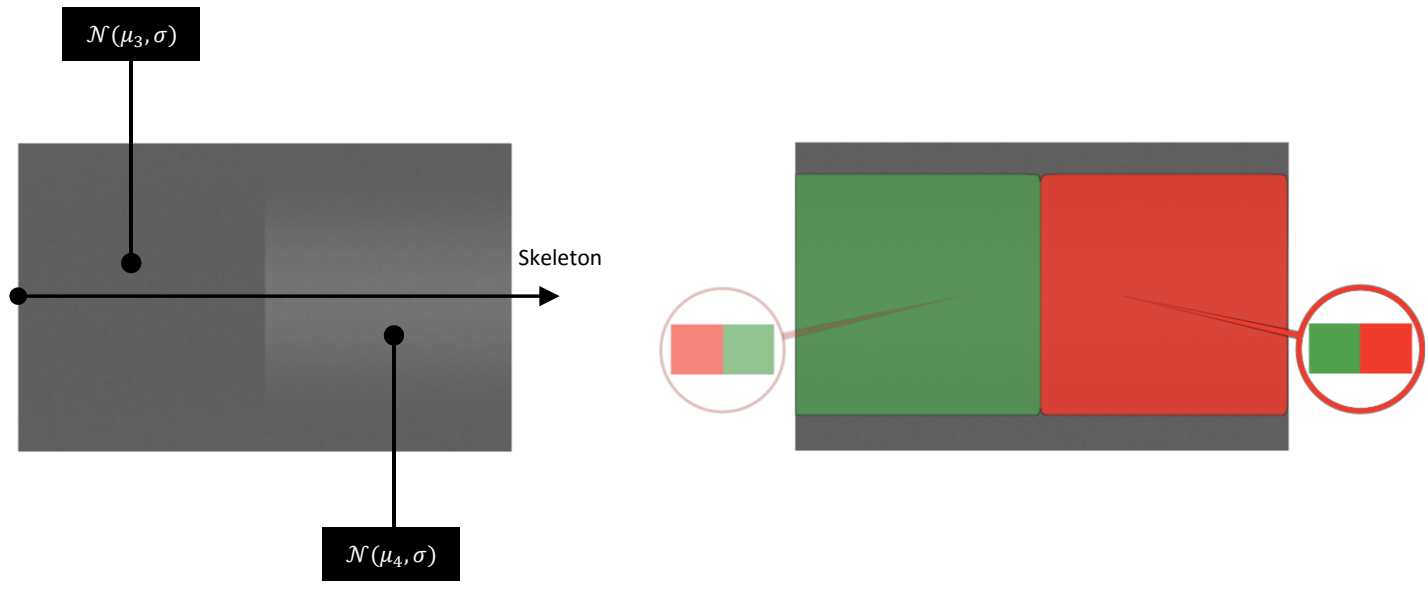
Figure 3.10: Test 2: structure of the test objects and the editing result for the case of $\Delta d_2 = 2$ HU.
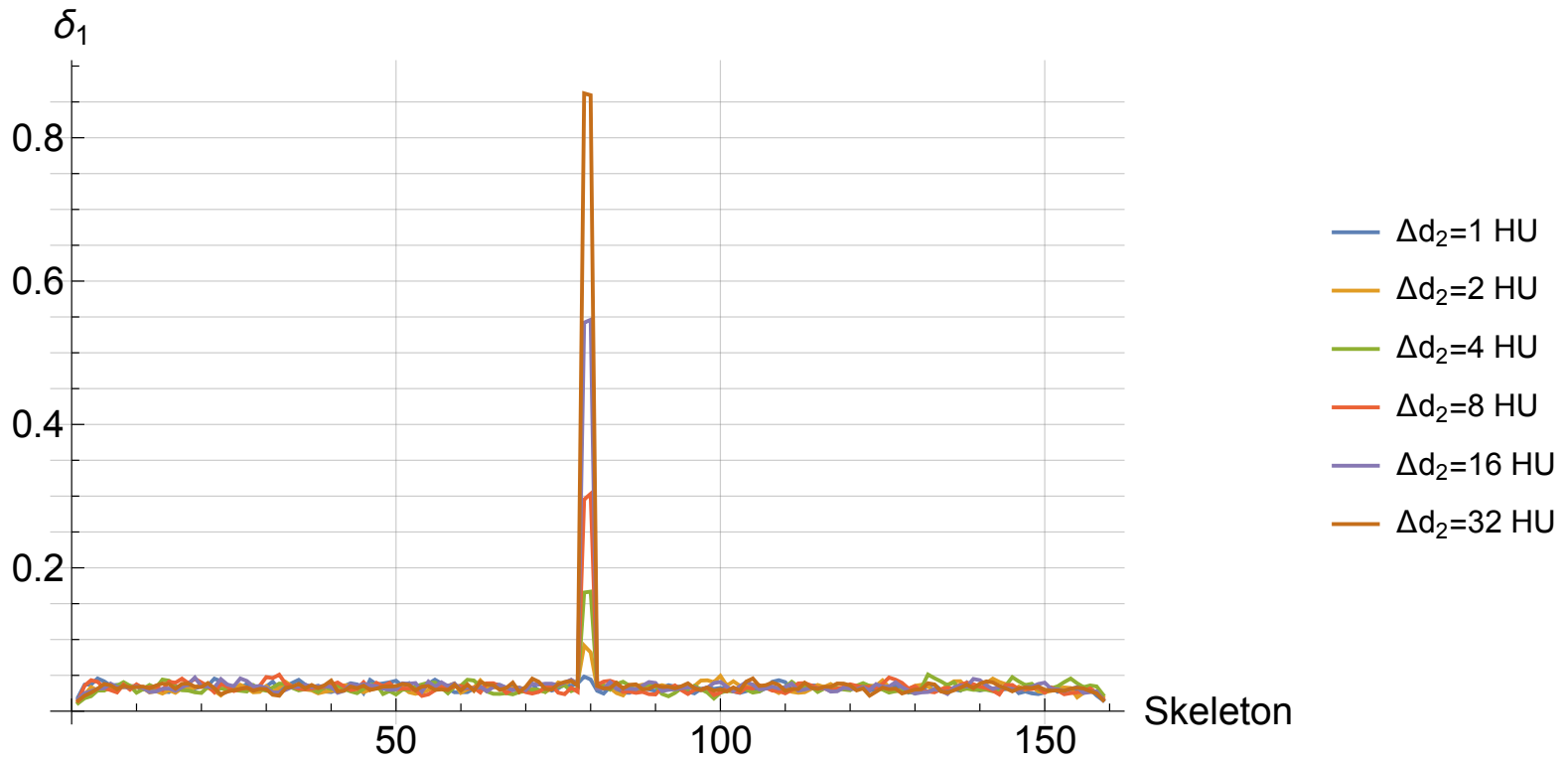
Figure 3.11: Test 2: dissimilarity values δ_1 along the skeleton with respect to varying $\Delta d_2 = |\mu_3 - \mu_4|$.

Table 3.1: Noise characteristics for different test passes.

Pass	#1	#2	#3	#4	#5	#6	#7	#8	#9
Standard deviation σ (HU)	0	20	40	60	80	100	150	200	300
Estimated PSNR (dB)	∞	41	35	32	29	27	24	22	19

PSNRs ≤ 24 dB. In order to check our method at passes #7, #8, and #9, the segmentation mask from pass #6 (the last available mask) was used as an input to our technique.

Based on our experiments, we draw the following conclusions:

- The correction scenario for the defect was detected at all tested noise levels (Figures 3.13-3.22)

- The correction of the defect was possible in all cases

- Our technique for segmentation defect detection and correction operates even at noise levels where the segmentation method fails (Figures 3.20-3.22)

3.4.3 Robustness with Respect to Different Initial Segmentations

The following experiment shows that our technique offers a consistent editing process even in case of **different initial segmentations**. The goal of this experiment is to assess the robustness of our approach with respect to the initial segmentation mask. We generated nine different initial segmentation masks of a certain object (liver in a CTA dataset) by adding noise to the data values and running the segmentation technique anew. The procedure of the noise generation is the same as described previously, and consists of nine passes as well. The quality is measured using the Jaccard coefficient J of the analyzed segmentation mask with respect to the ground truth segmentation mask, provided by domain experts. The segmentation masks differ by the number and the severity of defects. We define the severity of a defect as the number of wrongly classified voxels.

Some defects are shared by two or more masks. In this test we analyze whether the user (the author) can correct such defects in the same way in different masks. One possibility to perform such an analysis is to cluster the correction steps by location of the underlying defects. In total, we have four clusters of the defects. We show the entire correction process for pass #4 in Figure 3.23. The user starts with the initial segmentation mask in Figure 3.23a. First, the user corrects cluster #1 in Figure 3.23b. Then, cluster #2 is corrected in Figure 3.23c. Cluster #3 requires two correction steps (Figures 3.23d,e). Finally, cluster #4 is corrected in Figures 3.23f,g. The user achieves the result – the corrected segmentation mask in Figure 3.23h. All six correction steps were found in a rather straightforward manner: the user hovered over regions of interest and adjusted levels of detail accordingly. Another user would probably perform the same procedure.
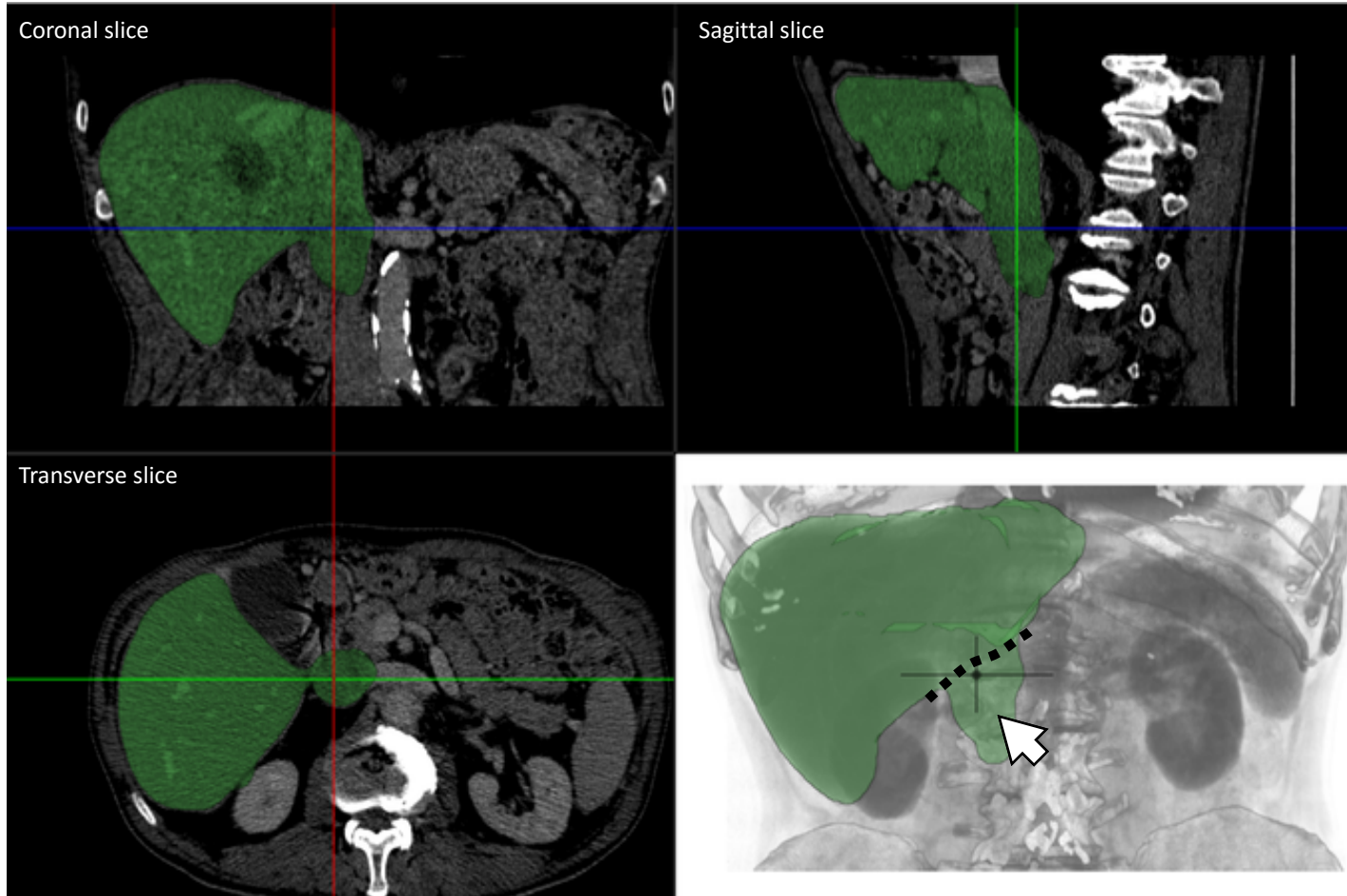
Figure 3.12: The dataset for the robustness test against noise. The object-of-interest is liver. The O_P segmentation defect is marked with an arrow. Coronal, sagittal, and transverse slices as well as DVR image are shown.
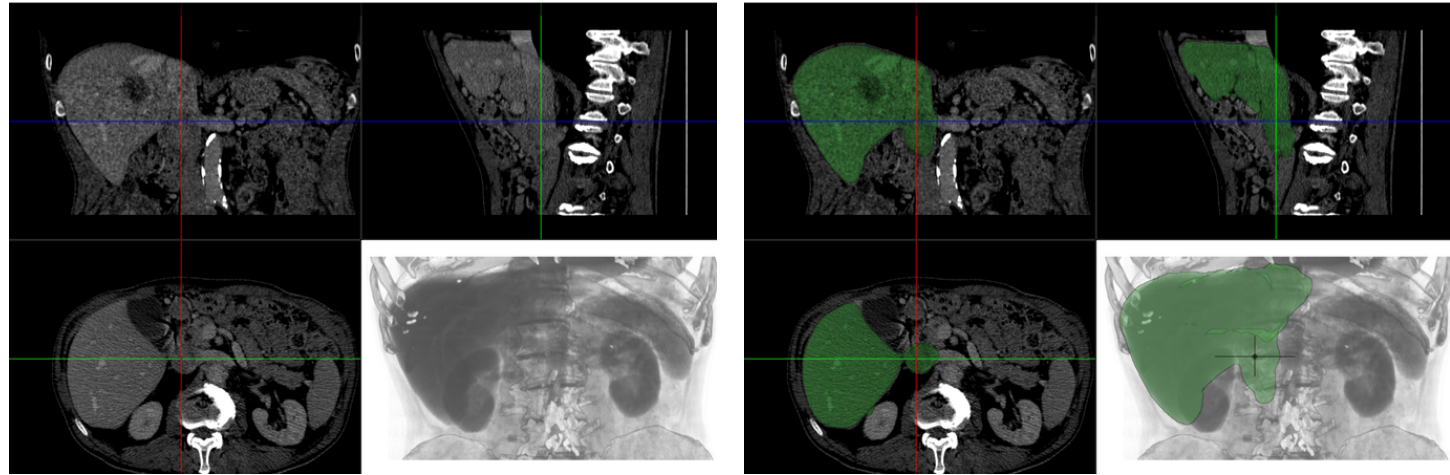
a: Volume data

b: Segmentation mask
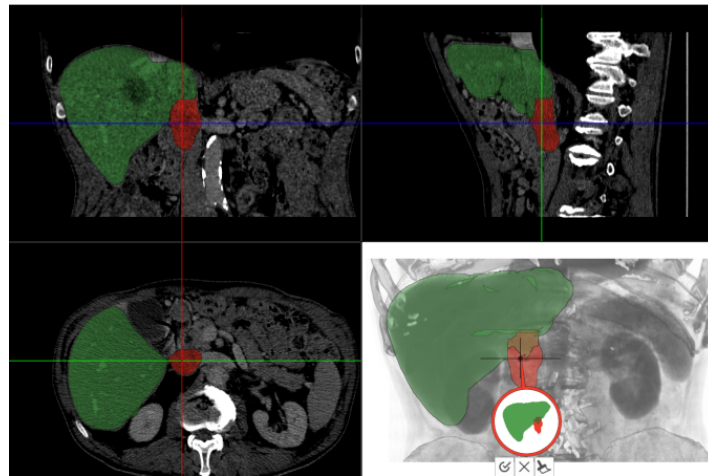
c: Correction scenario for the defect
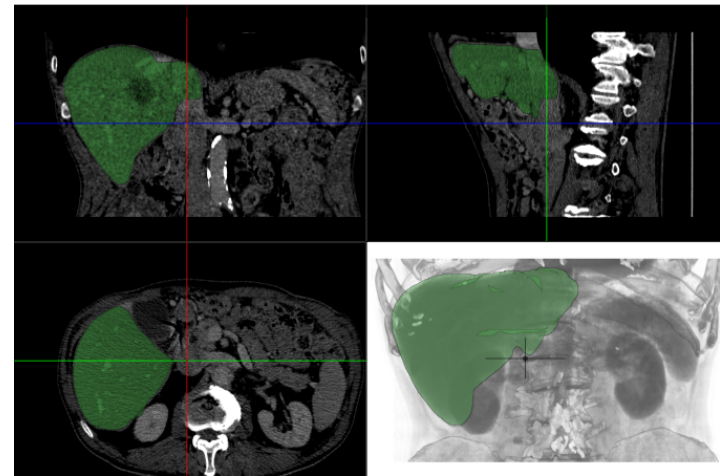
d: Results of the correction

Figure 3.13: Pass #1, PSNR= ∞ dB.

a: Volume data

b: Segmentation mask

c: Correction scenario for the defect

d: Results of the correction
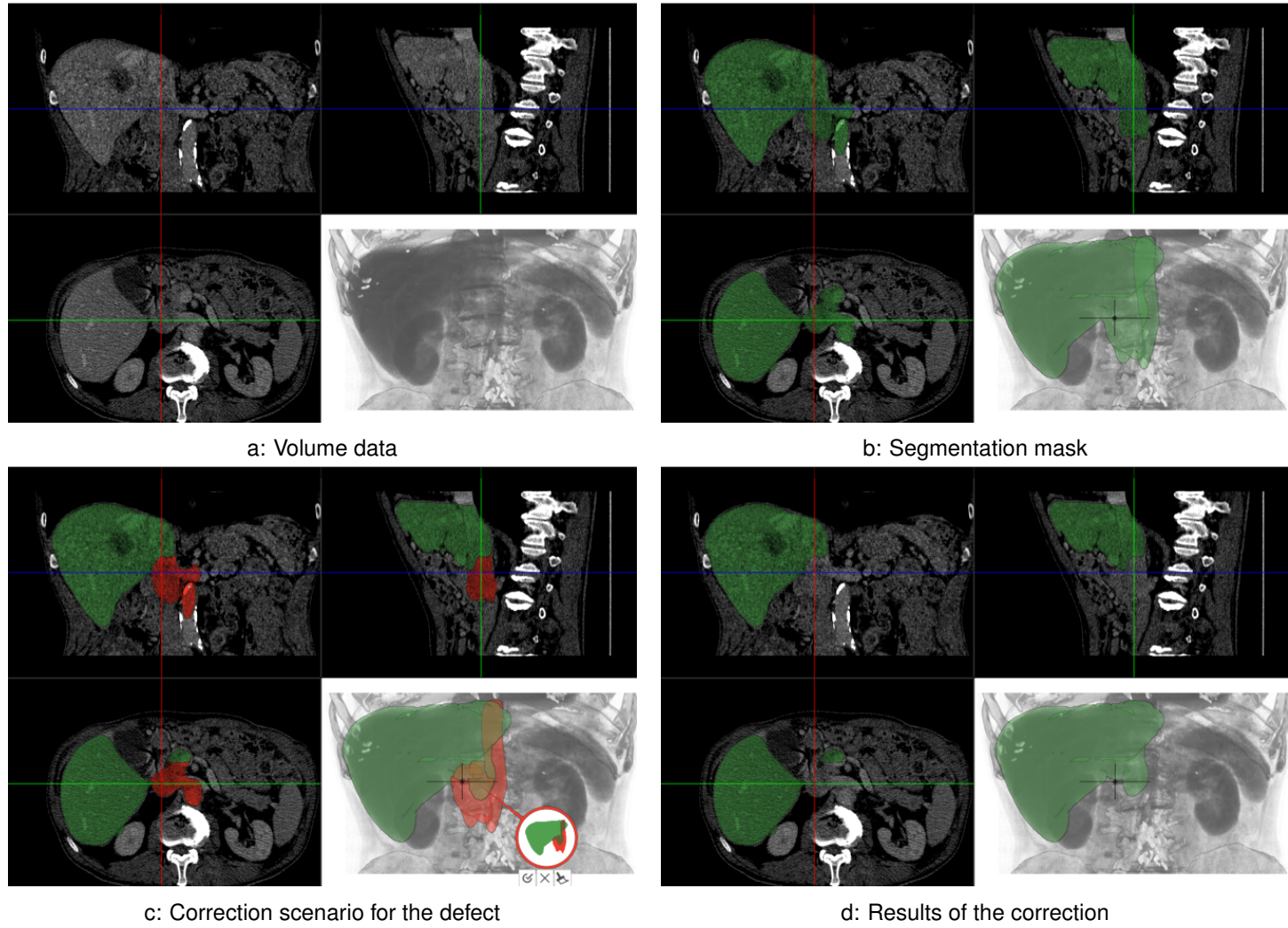
Figure 3.14: Pass #2, PSNR= 41 dB.
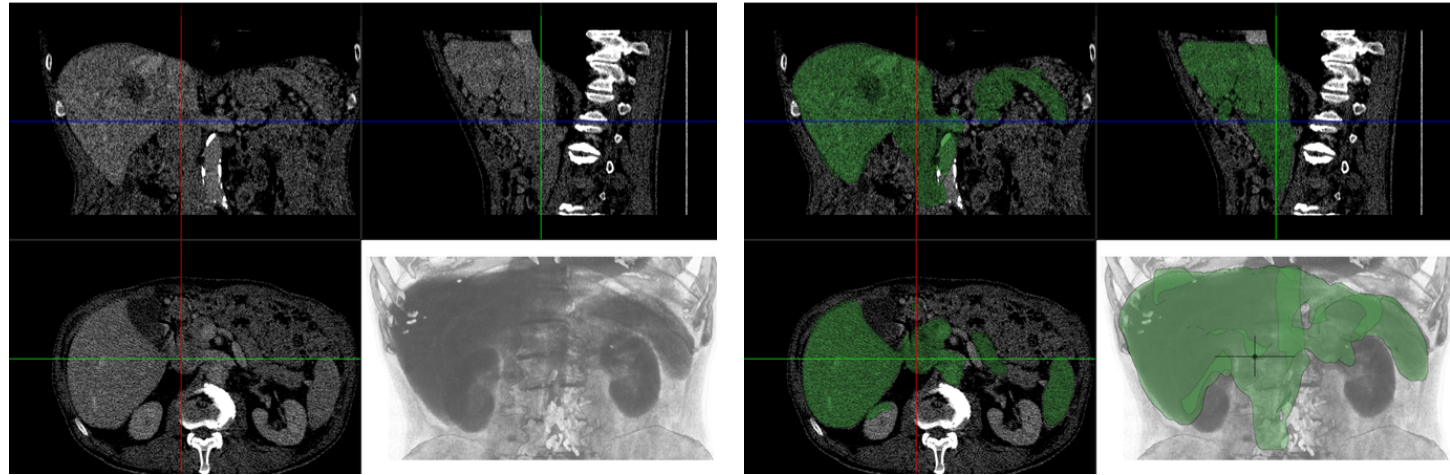
a: Volume data

b: Segmentation mask
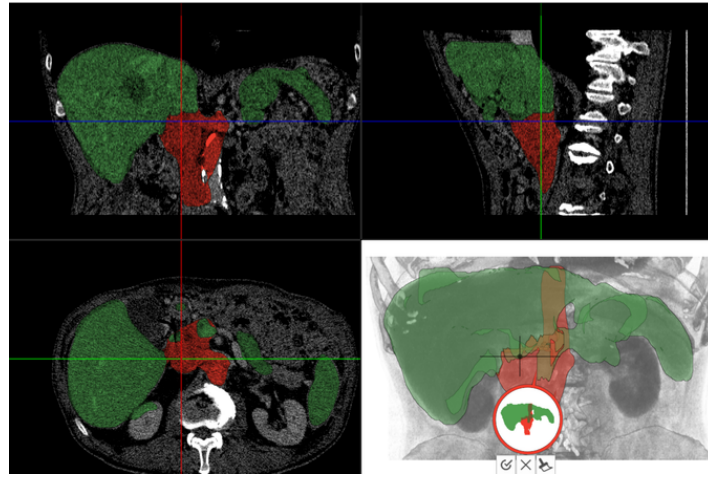
c: Correction scenario for the defect
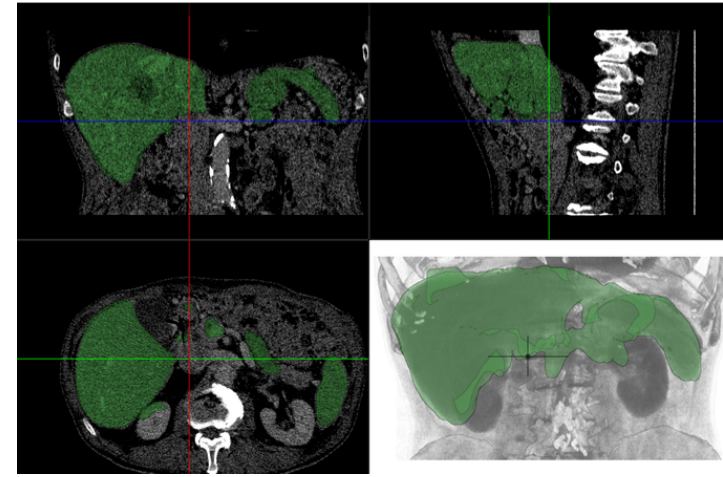
d: Results of the correction

Figure 3.15: Pass #3, PSNR= 35 dB.

a: Volume data

b: Segmentation mask

c: Correction scenario for the defect

d: Results of the correction
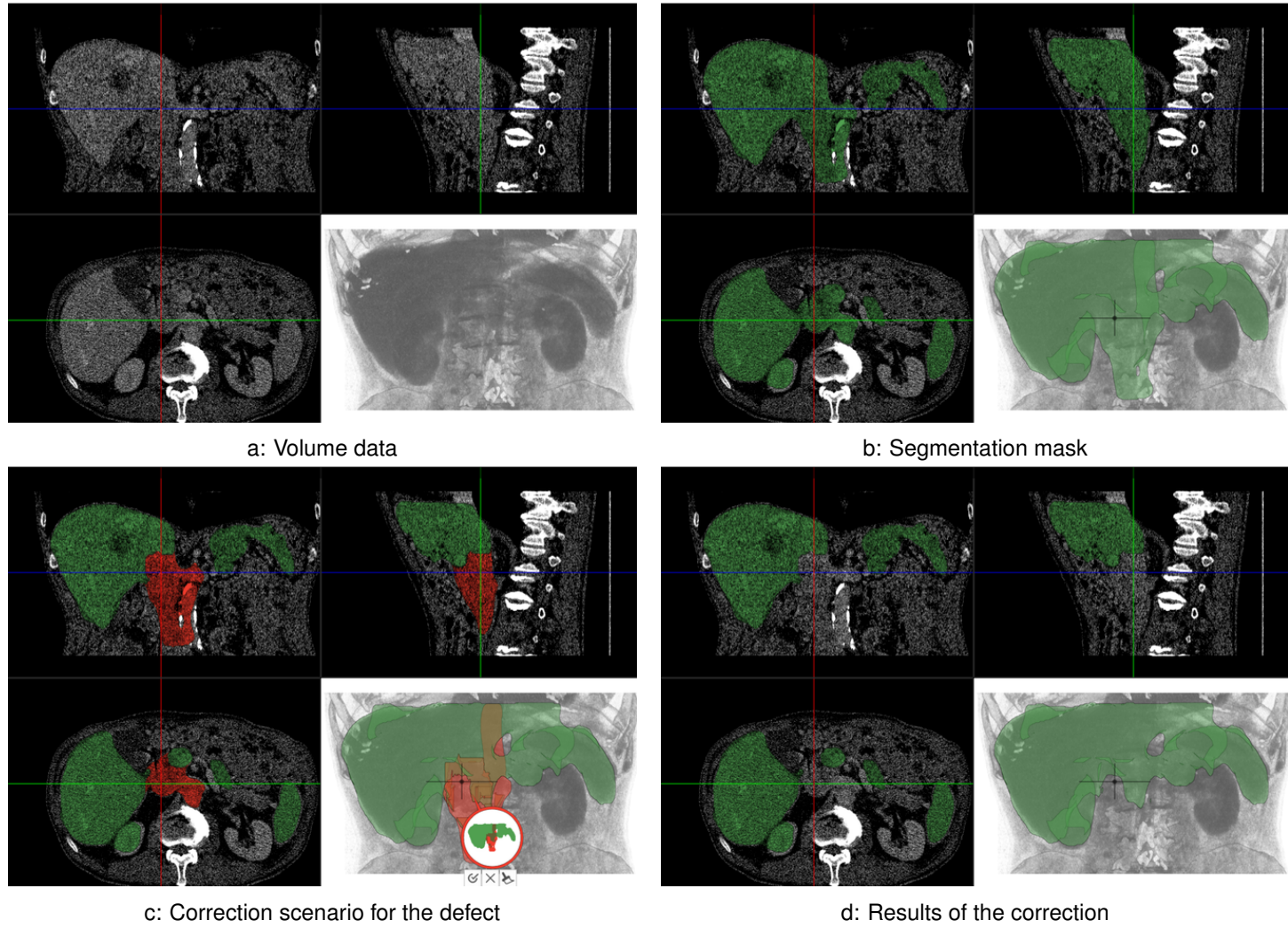
Figure 3.16: Pass #4, PSNR= 32 dB.
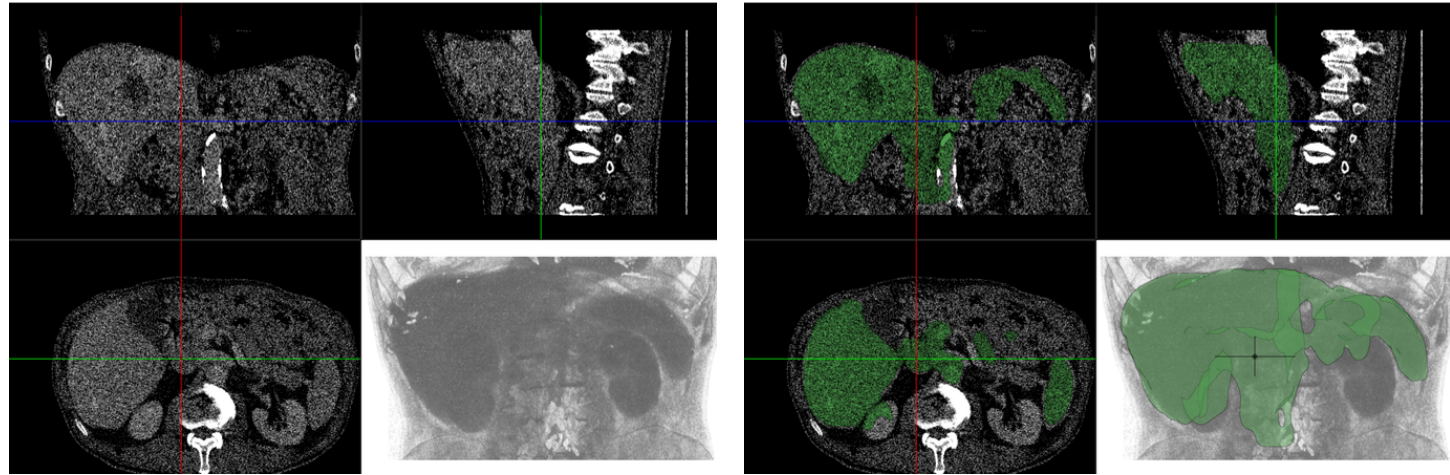
a: Volume data

b: Segmentation mask
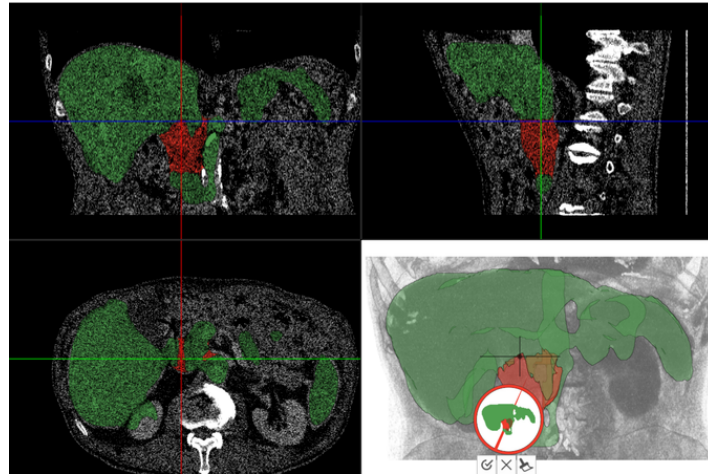
c: Correction scenario for the defect
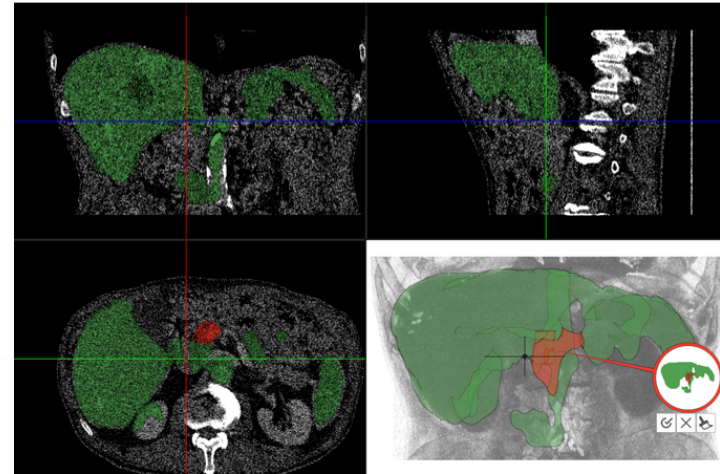
d: Correction scenario for the defect

Figure 3.17: Pass #5, PSNR= 29 dB. The correction process continues in Figure 3.18.

a: Correction scenario for the defect

b: Results of the correction
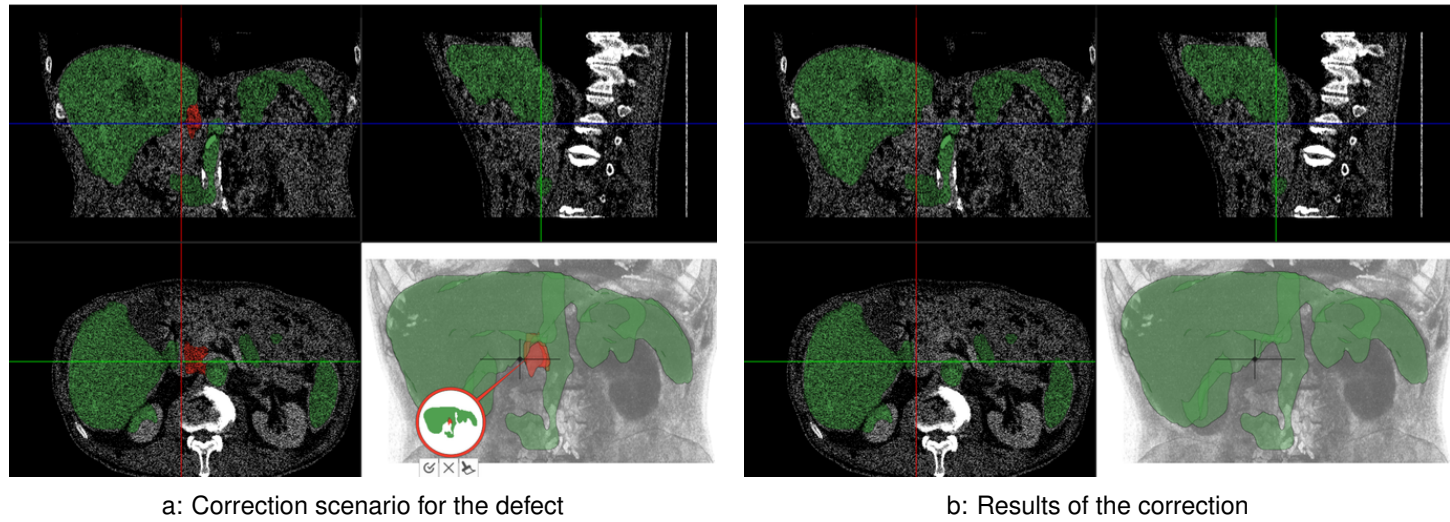
Figure 3.18: Pass #5, PSNR= 29 dB. In this pass, the target O_P defect was merged with another defect, which the segmentation method produced. Therefore, three correction scenarios were used to remove the incorrectly segmented structures. Two correction scenarios are shown in Figure 3.17. The last correction scenario is shown here.
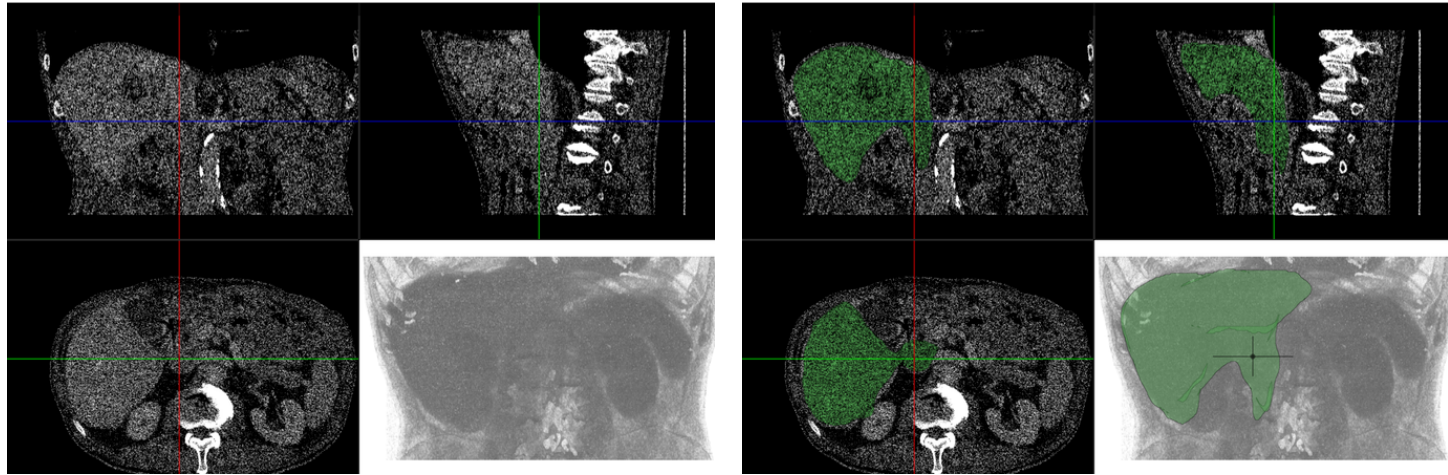
a: Volume data

b: Segmentation mask
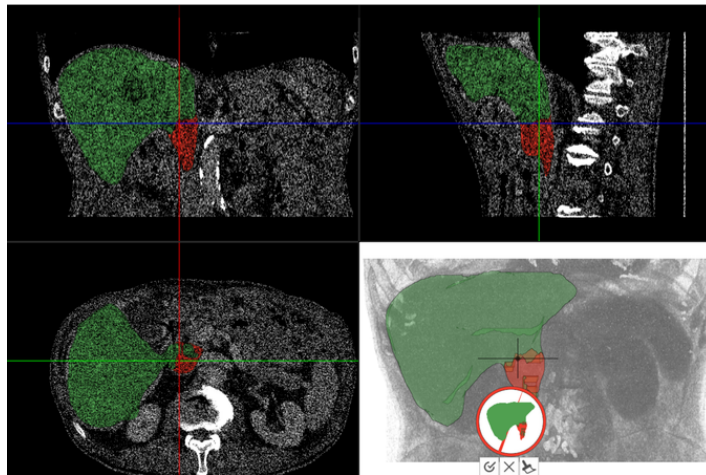
c: Correction scenario for the defect
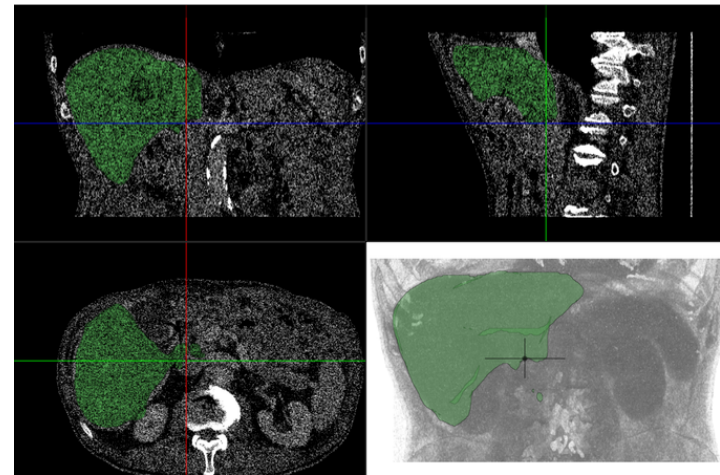
d: Results of the correction

Figure 3.19: Pass #6, PSNR= 27 dB.

a: Volume data

b: Segmentation mask

c: Correction scenario for the defect

d: Results of the correction
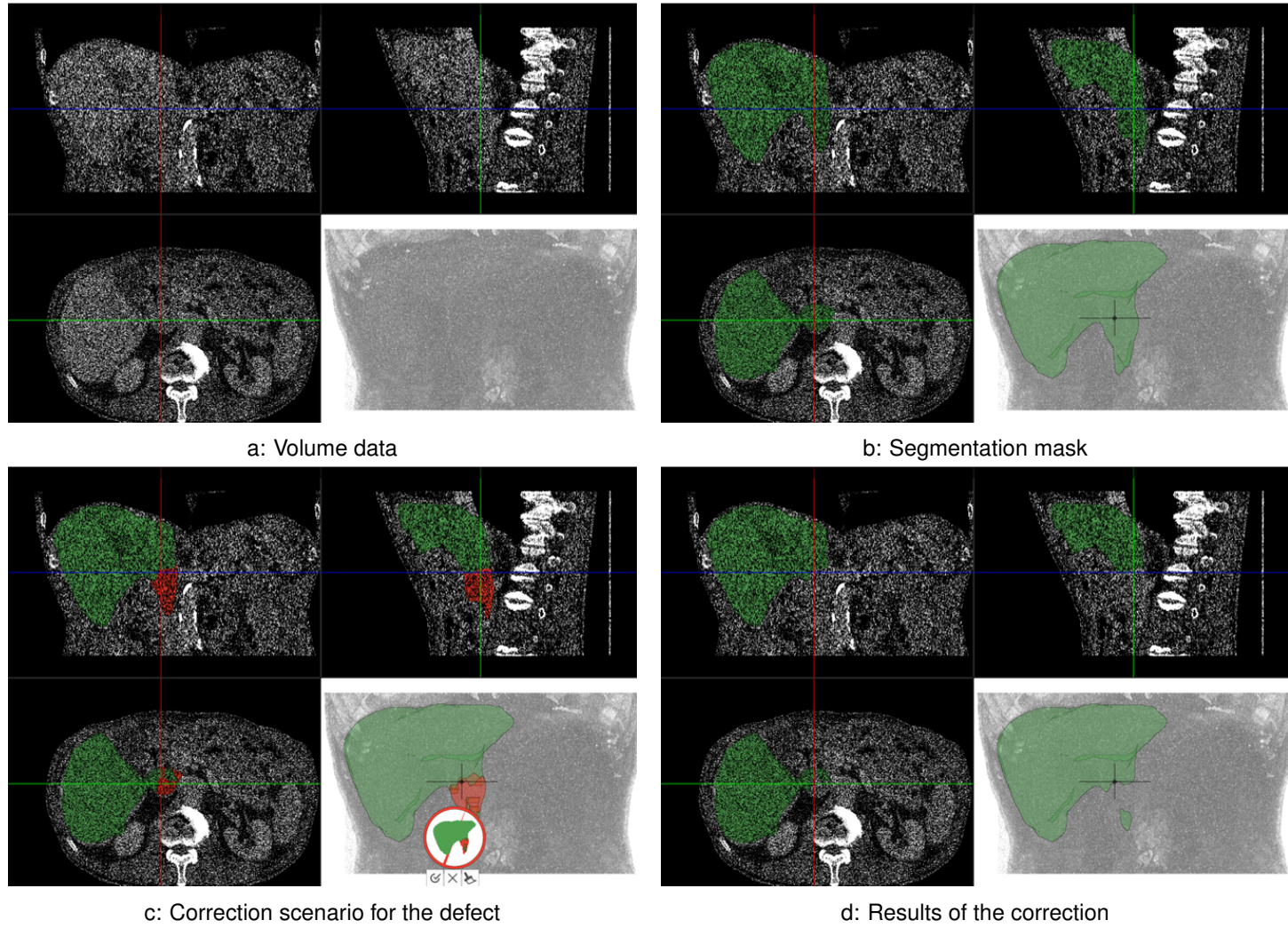
Figure 3.20: Pass #7, PSNR= 24 dB.
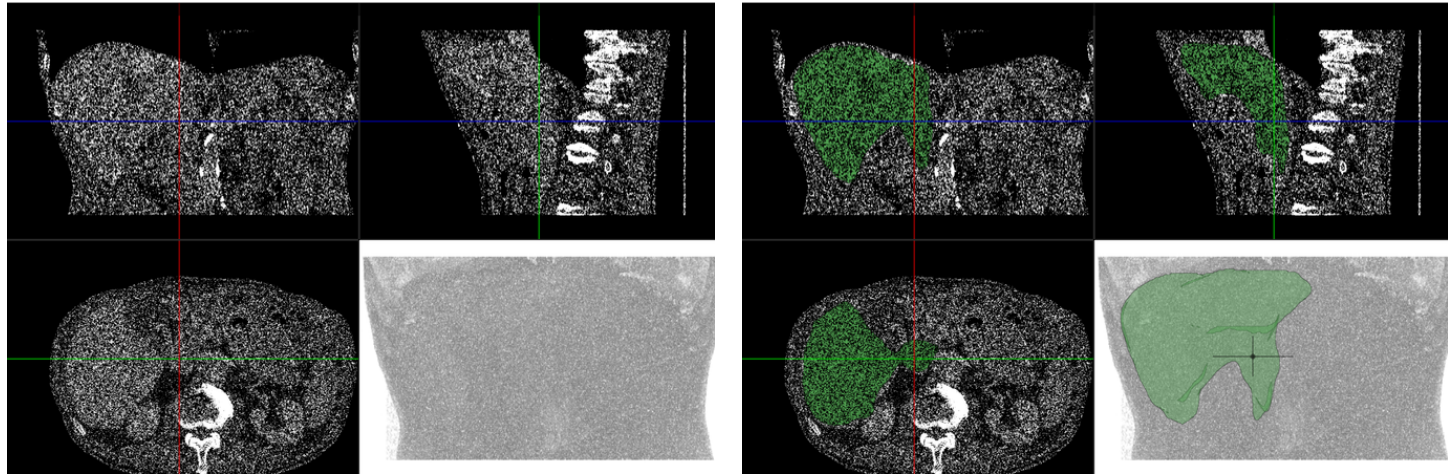
a: Volume data

b: Segmentation mask
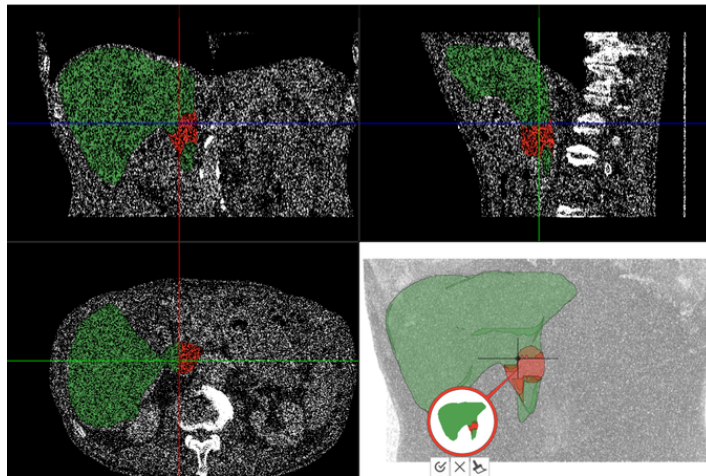
c: Correction scenario for the defect
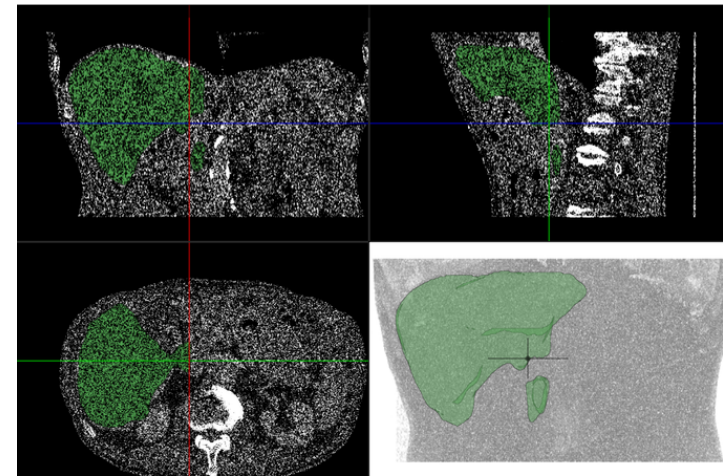
d: Results of the correction

Figure 3.21: Pass #8, PSNR$= 22$ dB.

a: Volume data

b: Segmentation mask

c: Correction scenario for the defect

d: Results of the correction
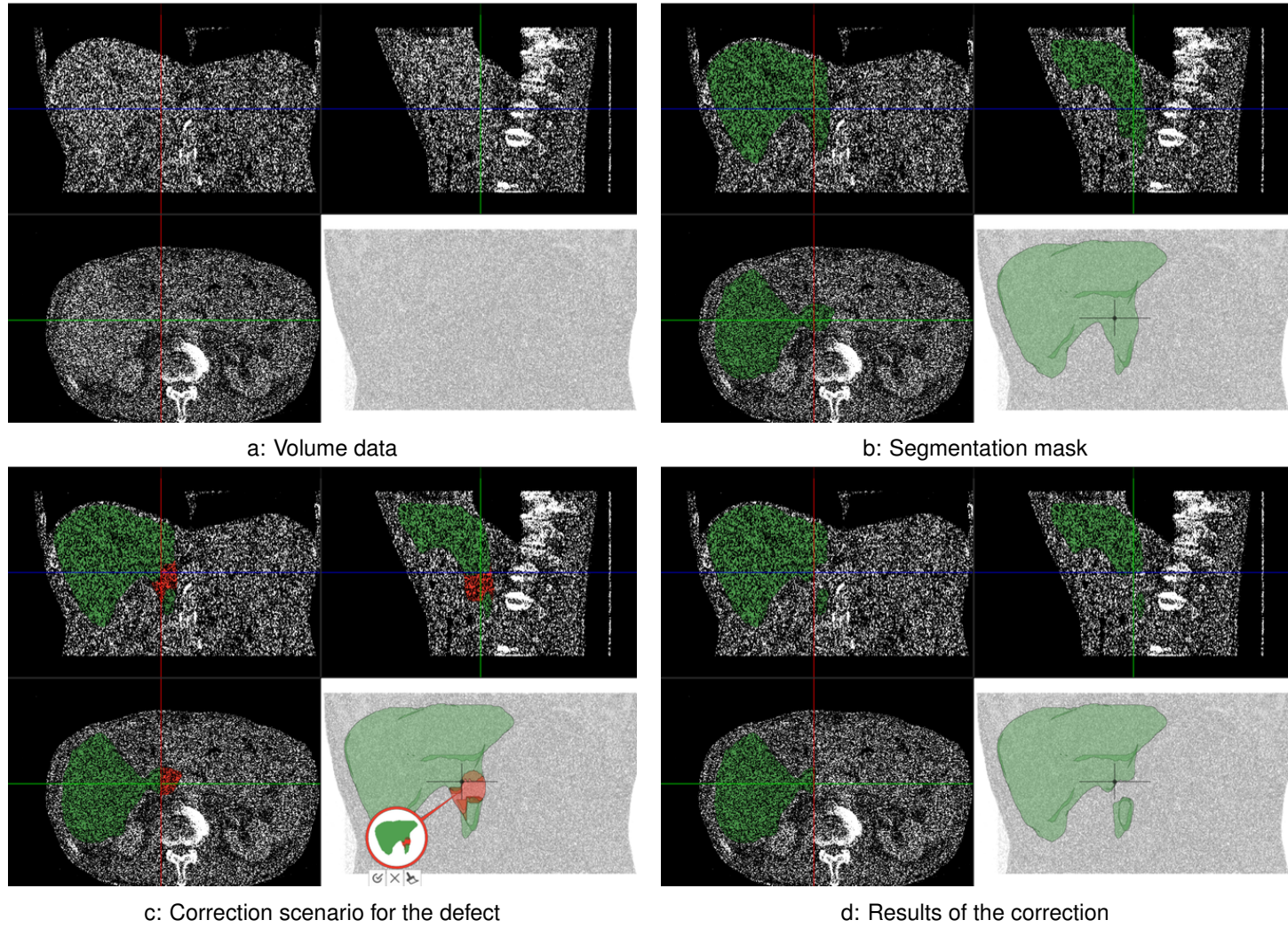
Figure 3.22: Pass #9, PSNR= 19 dB.

a: Input   b: Correcting cluster #1   c: Correcting cluster #2   d: Correcting cluster #3   e: Correcting cluster #3

f: Correcting cluster #4   g: Correcting cluster #4   h: Result
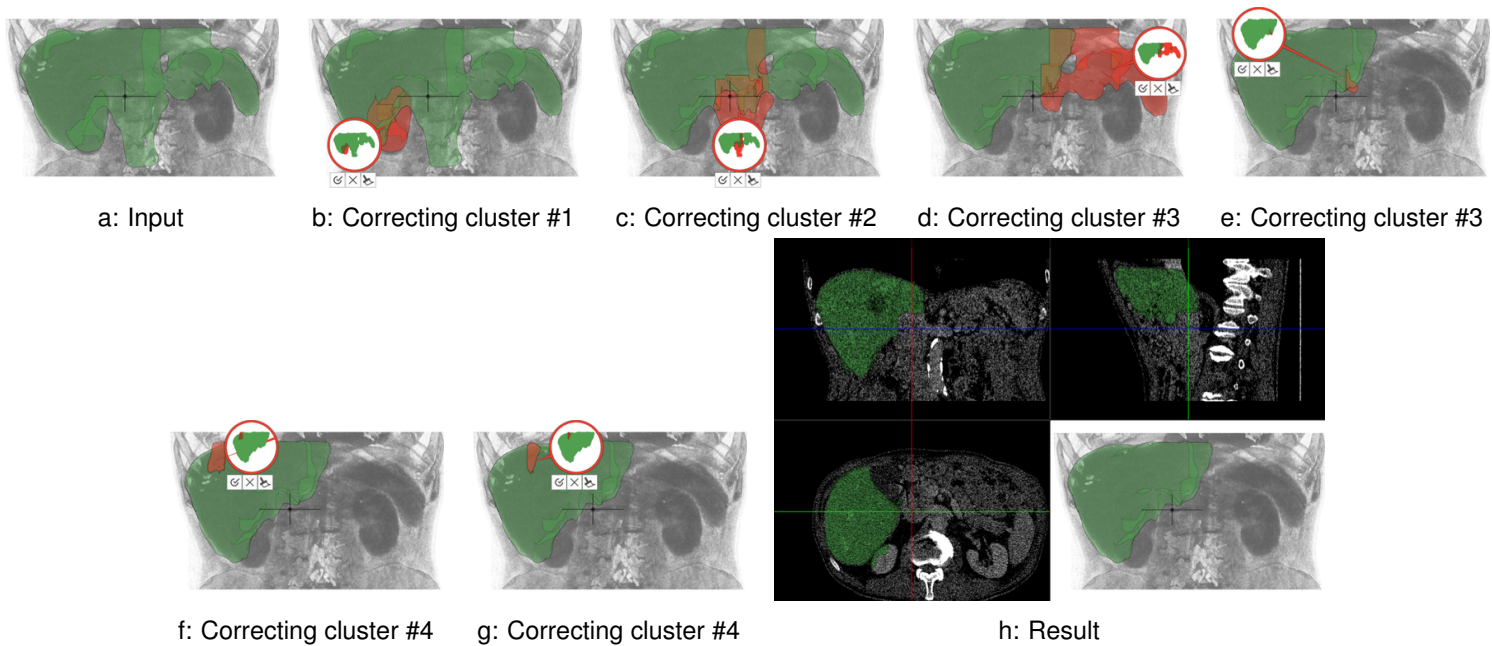
Figure 3.23: Six correction steps, applied to the clusters of the defects in initial segmentation mask #4.

Table 3.2: Comparison of the quality before and after the correction for different initial segmentation masks. The quality is measured by the Jaccard coefficient J (Equation 2.1) of the analyzed segmentation mask with respect to the ground truth segmentation mask, provided by domain experts.

Segmentation mask	Quality J before correction	Correction steps	Quality J after correction
#1	0.9	1	0.92
#2	0.84	3	0.91
#3	0.64	10	0.87
#4	0.61	6	0.87
#5	0.65	7	0.88
#6	0.85	8	0.88
#7	0.85	8	0.88
#8	0.85	4	0.88
#9	0.85	2	0.87

However, the clusters can be corrected in a different order. The chosen levels of detail may slightly vary. The correction processes for passes #3 and #5 are illustrated in Figures 3.24 and 3.25 respectively. By comparing the correction processes for passes #3, #4 and #5, we conclude that the user corrected similar defects in the similar way in different masks.

The correction of the four clusters of the defects is depicted in Figures 3.26, 3.27, 3.28. Apart from variations in the numbers of the correction steps, our technique allows the user to edit the object until a sufficient quality is reached independently of the initial segmentation mask. Even though the defects differ significantly by their severity, the user had edited them in a consistent manner in all nine passes. As our technique abstracts from a direct manipulation of the skeleton, the severity of the defects only affected the number of the correction steps. The user did not have to adjust the individual correction steps in order to accommodate for varying severity. The progress of the quality improvement during the correction is illustrated in Figure 3.29 for each initial segmentation mask.

The quantitative test results are given in Table 3.2. As it is impossible to achieve the ground-truth segmentation having significant amount of noise added, the user stops the editing process after correcting the four clusters of the defects. Minor variations in the resulting quality (from 0.87 until 0.92) are caused by significant differences in the quality of the initial segmentation masks (from 0.61 until 0.90) and the presence of added noise. For each initial segmentation mask, the user achieved approximately the same results, which suggests the stability of our technique with respect to the initial segmentation mask. Due to added noise, the segmentation masks varied abruptly and significantly. As each segmentation mask was corrected in a single try, we can claim that the results were achieved without prior knowledge of the data.

a: Input   b: Correcting cluster #2   c: Correcting cluster #2   d: Correcting cluster #3   e: Correcting cluster #3

f: Correcting cluster #3   g: Correcting cluster #3   h: Correcting cluster #4   i: Correcting cluster #4   j: Correcting cluster #4

k: Correcting cluster #4   l: Result
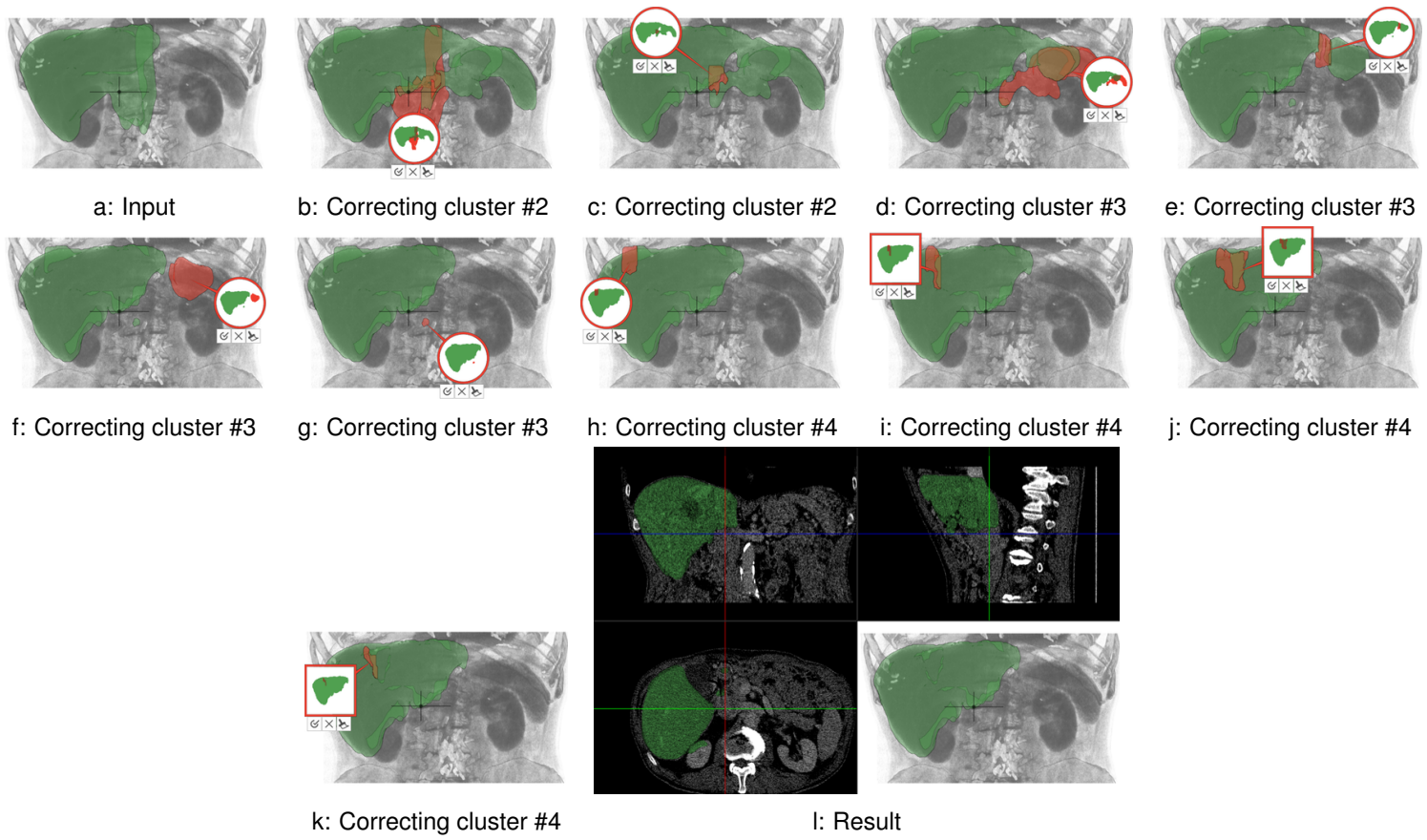
Figure 3.24: Ten correction steps, applied to the clusters of the defects in initial segmentation mask #3.

a: Input   b: Correcting cluster #1   c: Correcting cluster #2   d: Correcting cluster #2   e: Correcting cluster #2

f: Correcting cluster #2   g: Correcting cluster #2   h: Correcting cluster #3   i: Result
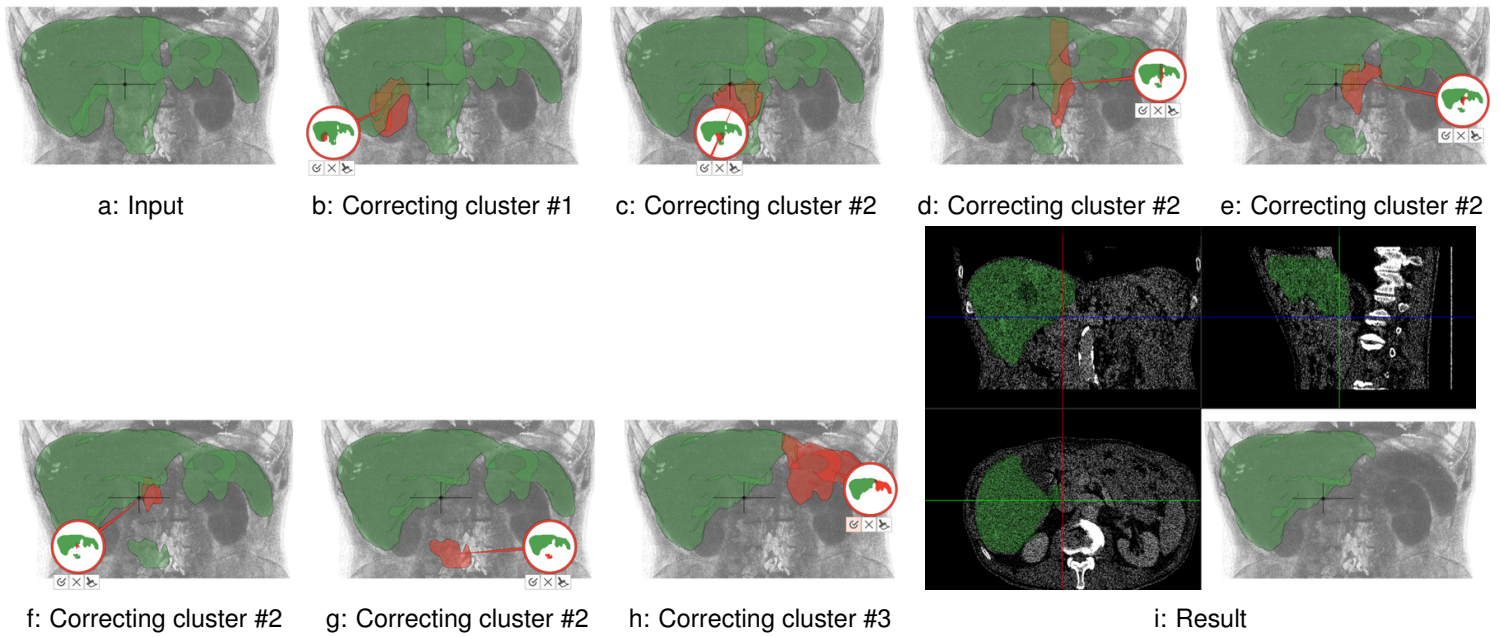
Figure 3.25: Seven correction steps, applied to the clusters of the defects in initial segmentation mask #5.
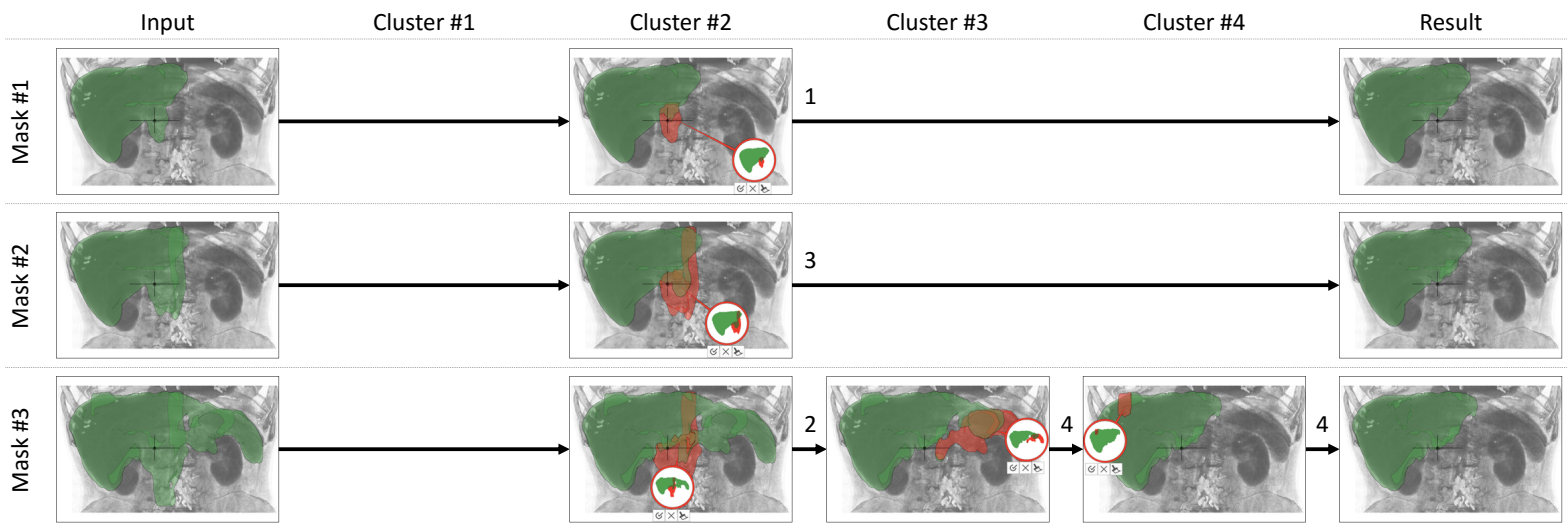
Figure 3.26: Correction steps for the clusters of the defects in initial segmentation masks #1, #2, and #3. The integers above arrows indicate a number of steps, required to correct a certain cluster.
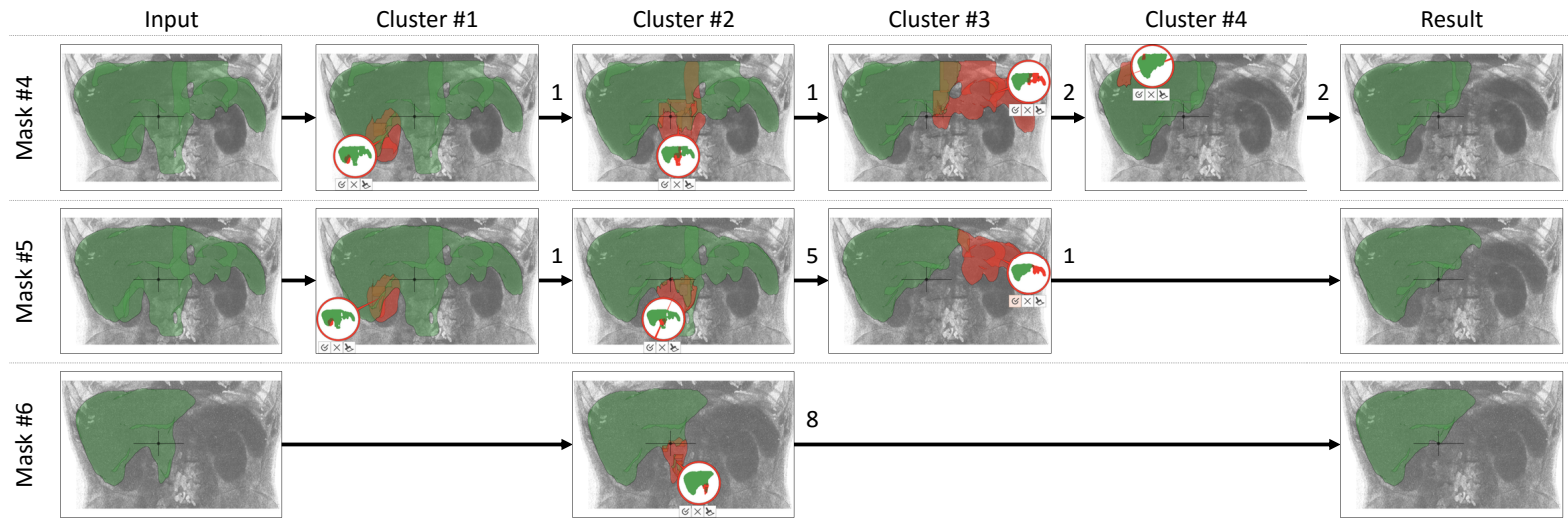
Figure 3.27: Correction steps for the clusters of the defects in initial segmentation masks #4, #5, and #6. The integers above arrows indicate a number of steps, required to correct a certain cluster.
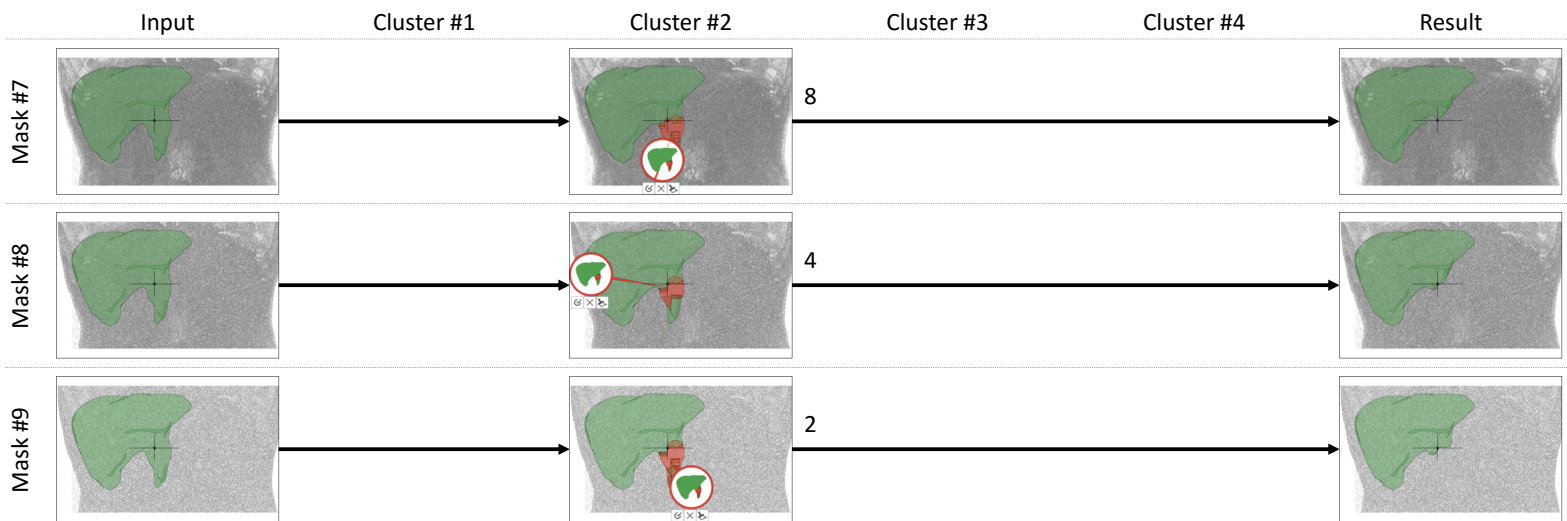
Figure 3.28: Correction steps for the clusters of the defects in initial segmentation masks #7, #8, and #9. The integers above arrows indicate a number of steps, required to correct a certain cluster.
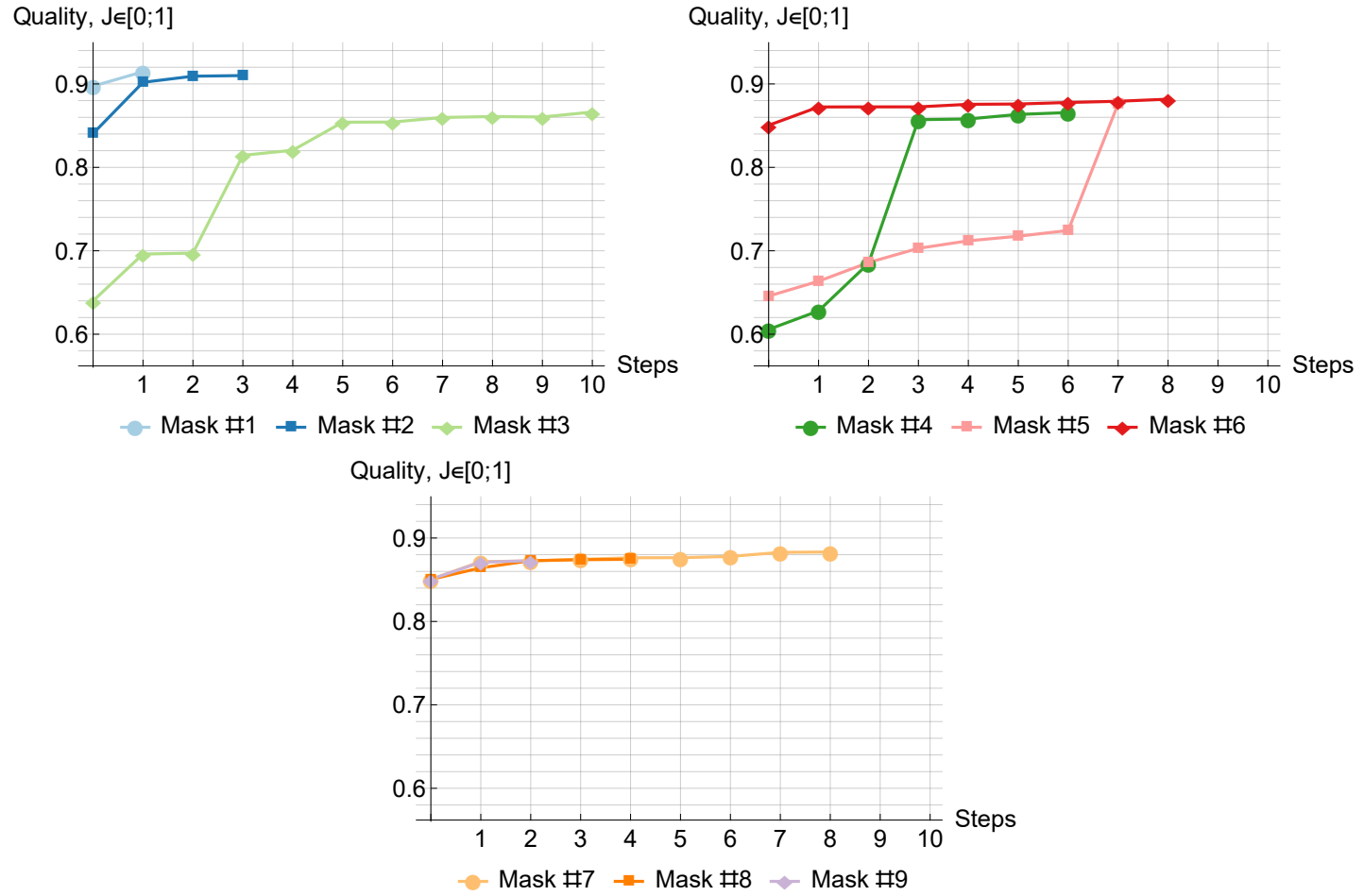
Figure 3.29: Improvement of quality with respect to correction steps for different input segmentation masks. The Jaccard coefficient J (Equation 2.1) is used as the quality measure.

3.4.4 Application Scenarios

We demonstrate the **generality** of our technique by editing objects in medical, biological, and industrial data. In the **medical** scenario, we applied our technique to liver segmentation (Figure 3.1) and vessel segmentation (Figures 3.6 and 3.30). Both cases, acquired with CTA, are clinically relevant and benefit from our tool. One common task in the latter case is the separation of vessels from bone if they are touching each other and automatic segmentations regularly fail. The usual procedure involves manually specifying separation objects – a tedious and time-consuming process. With our technique, this scenario was corrected with only twelve simple editing operations (Figure 3.30f) in about two minutes. The second example shows a vascular occlusion and, surprisingly to us, our technique provides the insight that the occlusion is even extending inside the stent, which hints to its potential usefulness even in segmentation-unrelated applications (Figure 3.6). The **industrial** scenario is represented by X-Ray Computed Tomography (XCT) data (Figures 3.31 and 3.32), featuring significant metal artifacts, rendering the results of common segmentation techniques impractical. Our method allows us to easily correct the segmentations of the metal rods (about 100 seconds, Figure 3.31) and the plastic box (about four minutes, Figure 3.32) – the latter being significantly more challenging. As for the **biological** scenario, we apply our technique to Electron Microscopy (EM) data of a mouse cortex, provided by the ISBI 2013 challenge "SNEMI3D: 3D Segmentation of Neurites in EM images" (Figure 3.33). The automatic technique described by Kaynig *et al.* [84] provided the segmentation of the neurons, which then requires proofreading (*i.e.*, manual inspection and correction), as explained by Haehn *et al.* [85]. The data suffers from high noise levels and the slicing nature of the acquisition technique, leading to difficulties in identifying cell boundaries. Correcting the segmentation of a particular neuron, our technique allows the user to isolate over-estimated regions as well as to recover under-estimated ones in only three simple operations in about 84 seconds – an otherwise cumbersome and time-consuming manual segmentation editing task. We show the result (Figure 3.33e) before the supported slice-based refinement.

3.4.5 Evaluation

We evaluate our method with respect to usability, guidance, interaction, and quality of the results. We conducted an evaluation with three radiologists, who have experience in segmenting various organs and correcting their segmentation masks, since the results of their currently employed methods do not meet the required quality standards. Two experts analyze data for surgical intervention planning, whereas the third assesses treatment results comparing data before and after treatment. The task was to correct the segmentation of the liver in abdominal CTA data. We used 26 datasets, including the material of the SLiver 2007 contest [68] and datasets from the routine practice of our domain experts. For the initial segmentation, we employed a technique based on the work of Šrámek and Dimitrov [86]. This technique merges watershed segments using data values and gradients. Before the evaluation, we introduced the experts to our system with eight datasets. The author presented all types of correction scenarios to provide a basic understanding and overview. Then, the experts were introduced to the workflow of the system, as well as to the method for suggestion generation. Interaction examples were presented. Finally, the experts used the system themselves on the training datasets.
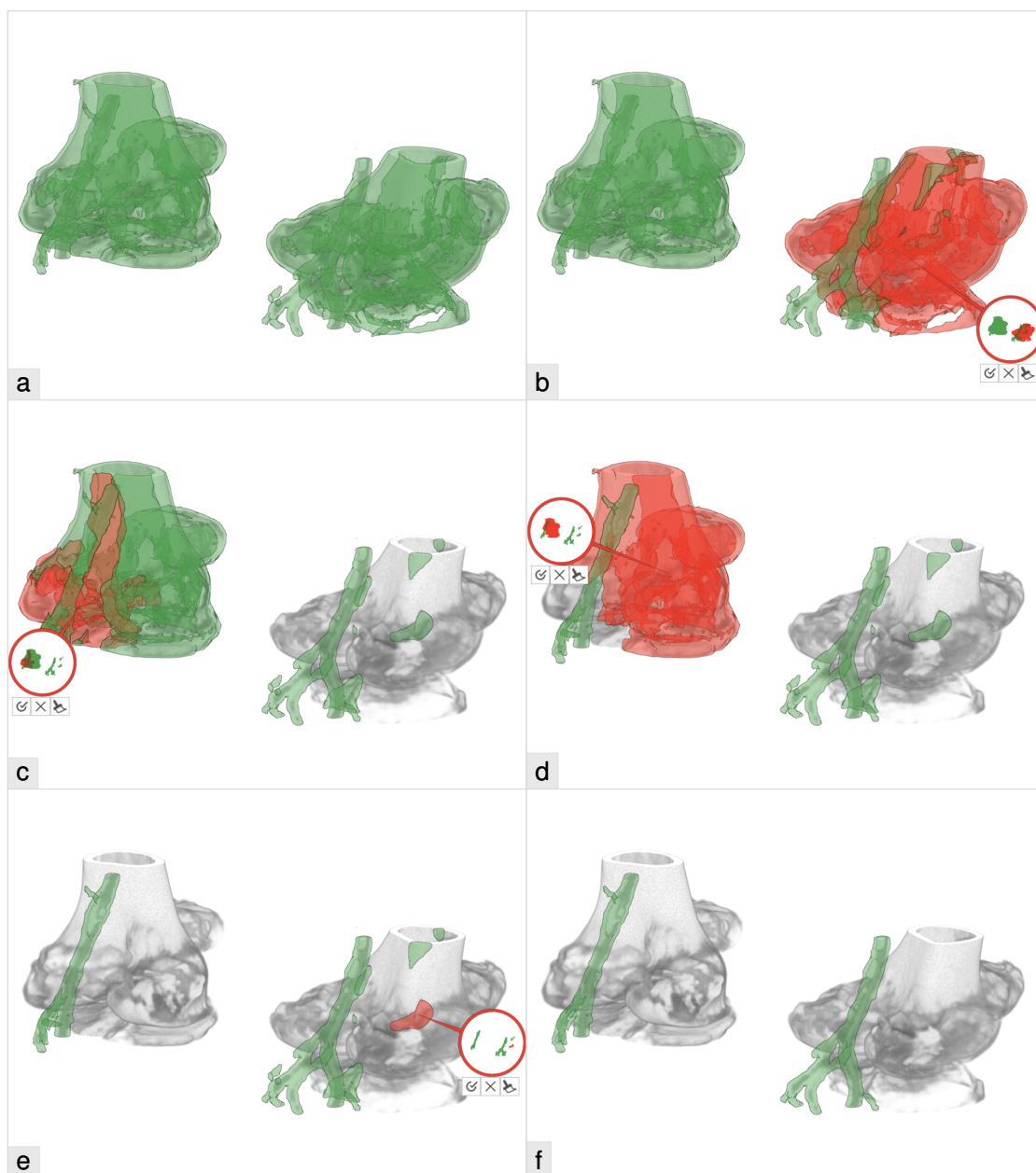
Figure 3.30: Guided volume editing for CTA data: a) the automatic vessel segmentation exhibiting defects (vessels and bones are touching), b-e) the first four correction operations (removal of red parts), f) the corrected vessel segmentation after only twelve operations.

Figure 3.31: Guided volume editing in industrial XCT data: a) the dataset, b) the initial segmentation of the two metal rods, c-f) the first four of seven correction operations, g) the corrected segmentation of the metal rods.
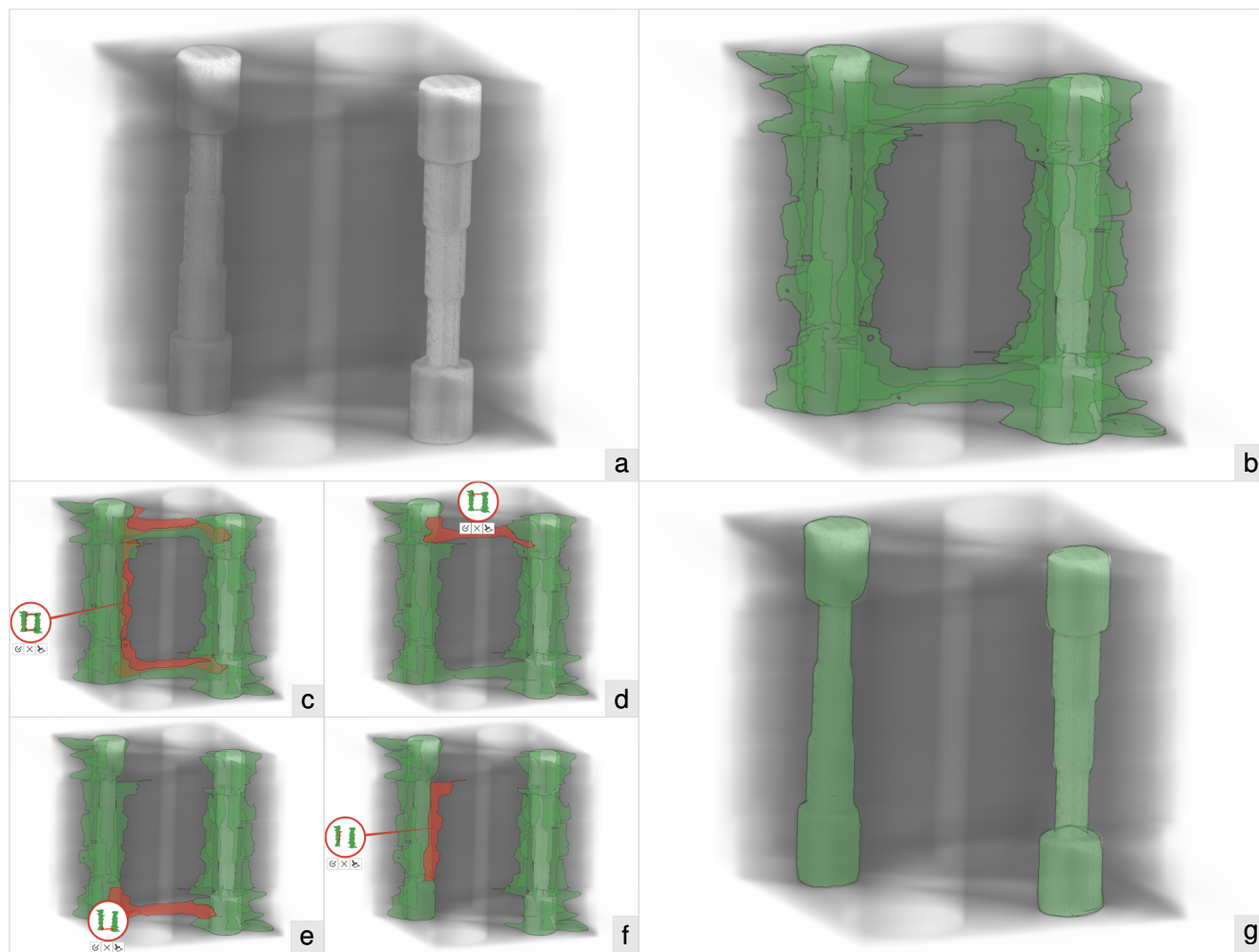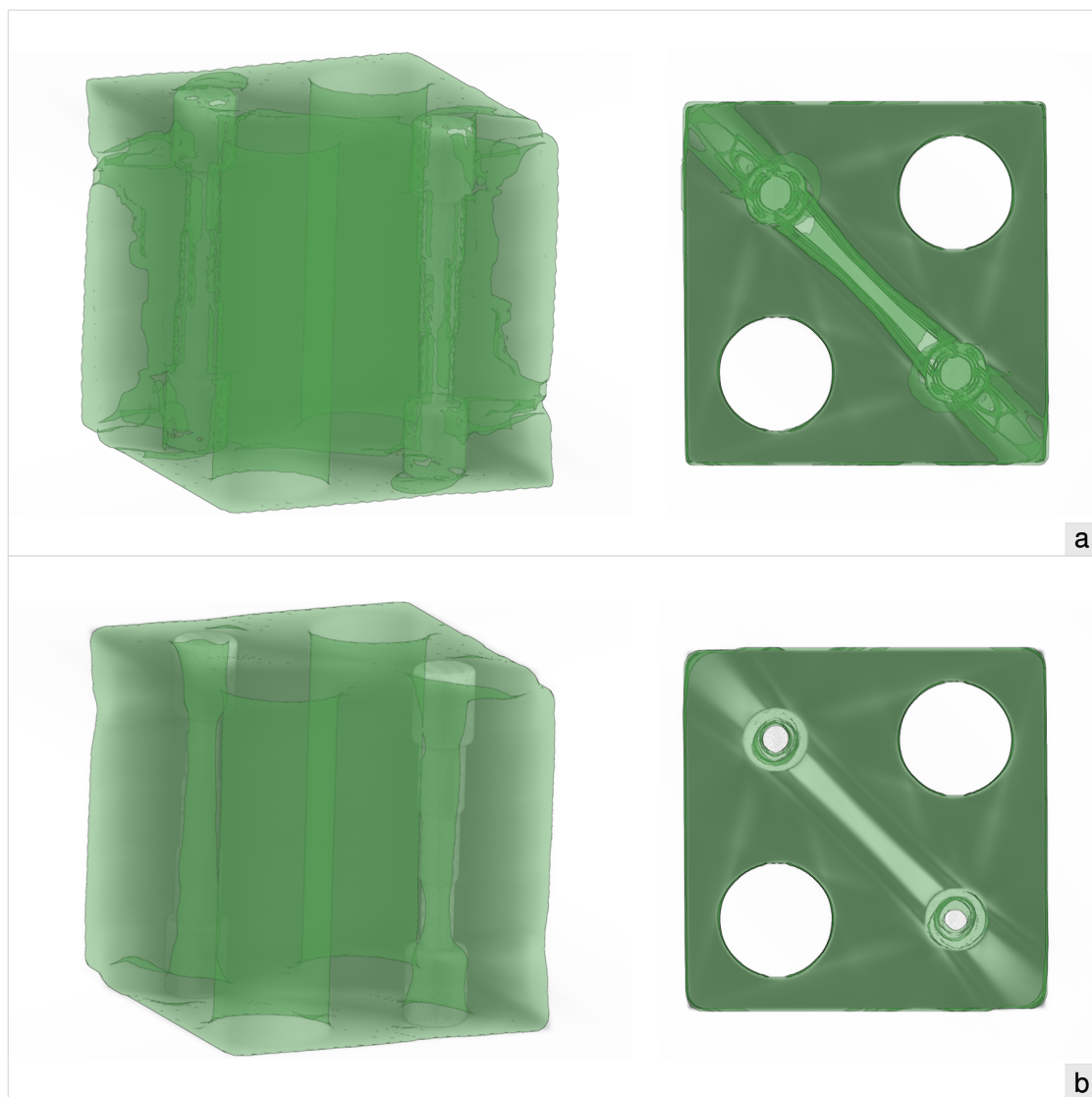
Figure 3.32: Guided volume editing in industrial XCT data: a) the initial segmentation of the plastic box, b) the corrected segmentation of the box, after only nine operations.
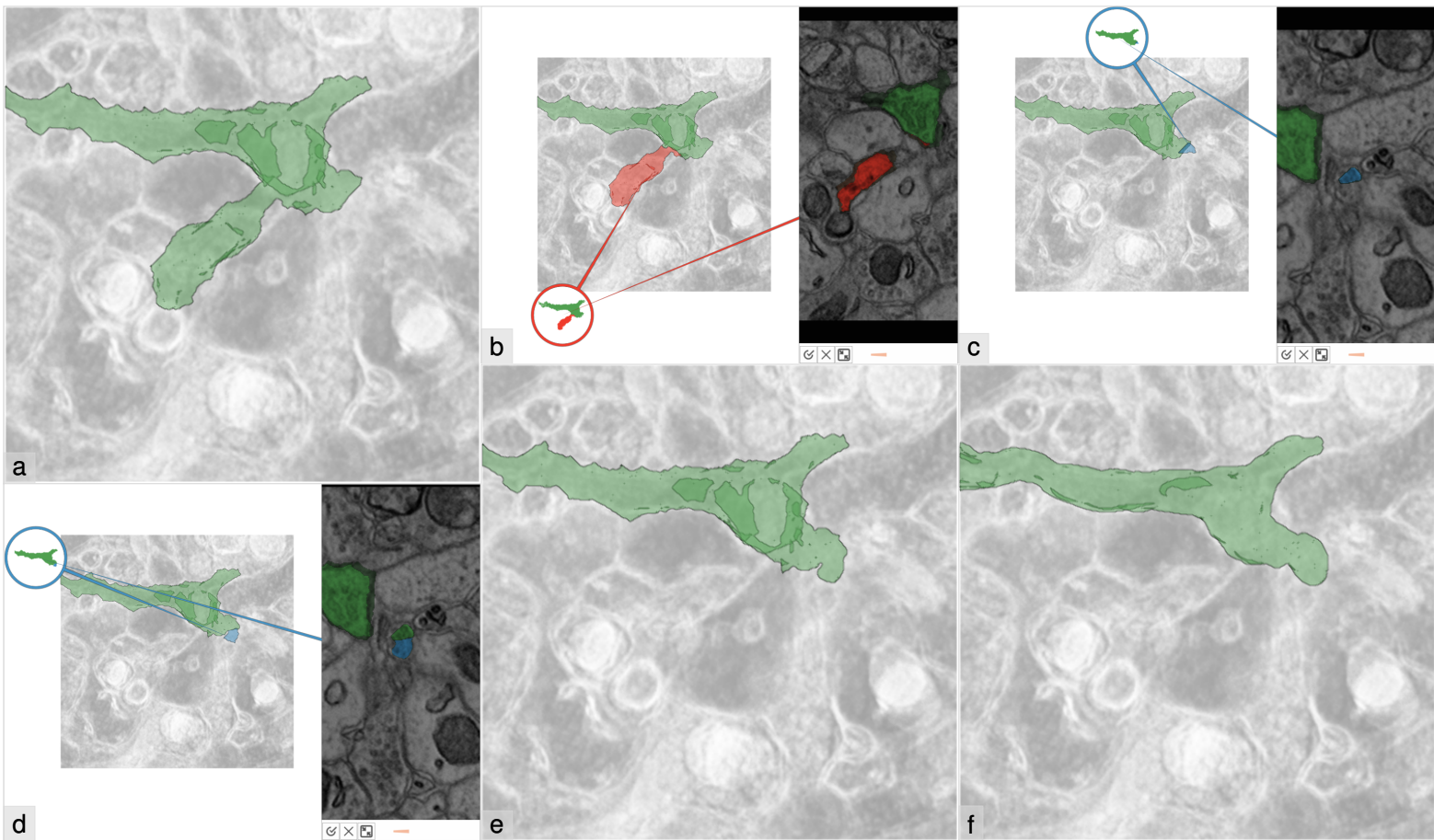
Figure 3.33: Guided volume editing in EM data: a) the neuron, segmented automatically, b-d) three correction steps, e) the segmentation of the neuron, corrected without slice editing, f) the segmentation of the neuron, created manually by the domain expert.

Afterwards, each expert corrected datasets with our technique (T1) and compared it to our previous technique (T2) from Chapter 2, and base-line segmentation techniques of their choice: interactive editing with the *Philips Brilliance* software (T3) and the *Toshiba Vitrea* software (T4), manual editing with the *General Electric LightSpeed* software (T5). As T3, T4, and T5 do not allow importing custom segmentation masks, the experts had to re-create masks beforehand. We excluded the timings and efforts of such operations to conduct a fair comparison. Each domain expert was assigned 18 unique pairs, consisting of one dataset (each assigned only once) and one technique (each used 6 times). After the experts finished their corrections, they answered our questionnaire (see Appendix A). Even after only a short accommodation period of a few hours, the domain experts rated our system favorably against well-established clinical tools of their workflow.

Technique Evaluation

We evaluated the usability and the guidance provided by our technique, as well as the following features: the adjustable levels of detail for the correction, the suggestions, and the miniatures (Figure 3.34a). Our system's average score on the System Usability Scale by Brooke [87] was 78 out of 100, which corresponds to the 82nd percentile of the perceived **usability**. Considering the relatively short accommodation period, this constitutes an overall good result and demonstrates the potential of our approach. The domain experts considered the overall user **guidance** through the correction process as useful. The experts specifically asked whether such guidance could be implemented in other editing methods as well. Since our system describes a fairly general concept of segmentation editing, it can be adapted to other methods, such as random walk segmentation algorithms. In this case, the hierarchy of the correction regions can be constructed using the hierarchical watershed transformation on probability values, computed by random walk algorithm. The outcome shows that there is room for improving the **overview** provided by our technique. The domain experts mentioned that it could be interesting to display the suggestions for an object together with the complement at the same time. A hierarchy visualization of the correction scenario space would be a useful extension, which may help the user in finding the required correction scenarios faster. The domain experts rated the possibility to adjust the **level of detail** for the correction as very useful. With this feature and its immediate visual feedback in the system's interface, the experts were able to quickly select the correction scenario at the desired level of detail. The experts specifically mentioned that the **suggestions** of our system allow them to focus more on the fine-tuning, analysis, and verification of the corrections. The **miniatures** were well accepted by the participants, as they provide an overview and simplify the exploration of the correction scenarios. The experts confirmed that our technique, combined with the automatically shown slice views, provides sufficient information to make **decisions** in segmentation correction.

Comparative Evaluation

We compared the volume editing techniques in terms of interaction, time spent, ease of use, tolerance to imprecise user input, and user satisfaction (Figure 3.34b). Our technique T1 requires the least **interaction**, as it suggests suitable correction scenarios, resulting in a reduction of data
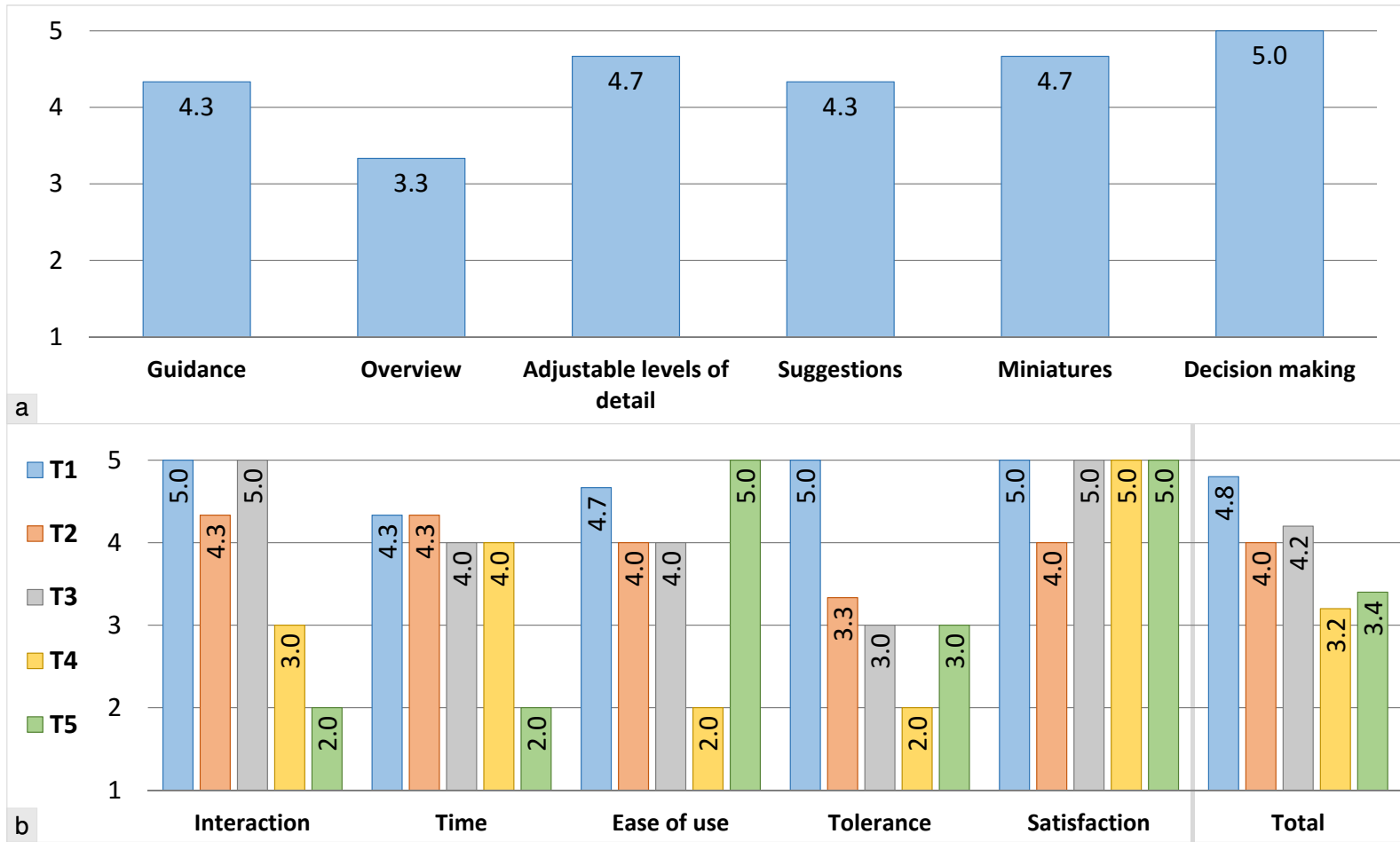
Figure 3.34: a) Evaluation of our technique. b) Comparative evaluation. Grades for (a) and (b) range from 1 (worst) to 5 (best).

exploration time and a reduced need for manual correction. The necessary interaction **time** is shortest with our T1 method, since the user explores the correction scenarios, then fine-tunes and verifies the corrections instead of manually specifying the correction regions. The experts noted, that slice-based editing, like in T5, is always **easier to use**, than 3D-space-based editing. Nevertheless, our technique is rated second, being just slightly more complex than T5. As we present correction scenarios to the user, voxel-based interactions are avoided, leading to an increased **tolerance** to imprecise user input in comparison to the other evaluated techniques. Since the domain experts still favor the possibility to edit on the voxel level (even if only in rare cases), they have rated our technique T1 and their clinical methods T3, T4, and T5 as **completely satisfying** regarding the achievable quality of the results. With no voxel-level editing available, the experts achieved less satisfying results with technique T2.

Quantitative Evaluation

We analyzed the defect **detection and correction rates** to quantify the accuracy of our technique. The first domain expert had no problems with correcting O defects (12 O_P and 8 O_L), but left 1 out of 5 U_P and 1 out of 2 U_L defects uncorrected. The second expert successfully corrected O defects (1 O_P and 9 O_L), but did not correct 1 out of 11 U_P and 3 out of 5 U_L defects. While correcting two of these U_L defects, the expert had an issue with the lowest level of detail. The third domain expert corrected all but one U_P defect out of 8 O_P, 6 O_L, 5 U_P and 3 U_L defects. As each expert left one U_P defect uncorrected, we investigated the issue. We observed that these three uncorrected defects are large under-estimation defects which are caused by tumors. Also, these defects were not surrounded by the corresponding segmentation masks. As a result, they were only partially captured by the complements of the segmentation masks. Correction of these defects requires multiple steps, which were not straightforward to our experts. An example of such a defect is the large tumor in the liver in Figure 3.35. This issue is not specific to our method, since highly competent domain experts had difficulties in correcting these defects with other techniques too. One way to resolve the issue is to modulate the growth of the complement locally by the dissimilarities between parts of the background and the object. In case of low dissimilarity, one can increase the growth value, capturing large under-estimation defects entirely. If the dissimilarities are high, one can use the default growth value, providing the user a possibility to correct defects if necessary. As we support manual local editing, all remaining defects could be fixed for complete satisfaction. Nevertheless, the results, presented in this chapter, were obtained without such low-level editing. To evaluate the **efficiency** of the techniques, we recorded the interaction times. As for the **accuracy**, we evaluated the quality with the Jaccard coefficient J (Equation 2.1) of the resulting segmentation with respect to the ground truth, created manually by domain experts. A perfect match yields one. The coefficient decreases, as differences appear. Before the correction, the average quality was at 0.83. With our T1 technique, the experts achieved an average quality of 0.92 in 40 seconds on average. The same average quality was reached with T2, but it took one minute on average. The quality and the interaction time is comparable to results of the interactive method by Beichel *et al.* [12] (before the MBR stage). Beichel *et al.* [12] state that their interactive segmentation technique is not limited to the liver, but this was not evaluated in their work. Our T1 approach, however, was applied on different types of data: the liver CTA, CTA of the human lower extremities, industrial XCT and biological EM

data. With T3 and T4 the experts achieved an average quality of 0.91 in 100 seconds and of 0.88 in 150 seconds respectively. The technique T5 achieved the best average quality (0.94), but it took much longer (13 minutes on average). We can conclude that the **quality** of the suggestions is comparable to the quality of manual corrections, fine-tuned by the domain experts with T2. Figure 3.1 depicts the correction of each of the four defect types.

3.5 Summary

In this chapter, we proposed a novel approach for guided volume editing of segmentation results based on the analysis of histogram dissimilarities. Considering the shape of the object by means of the skeleton, we allow users to correct parts of the object that contain segmentation defects. Our system makes suggestions, which guide users through the entire correction process. The presented results show that our system provides an efficient segmentation correction, while offering the possibility to reach the desired quality with a small number of simple interactions in a short amount of time.

In order to improve the exploration of the suggested correction scenarios, we next develop a method of statistically analyzing different corrections for dissimilarity against reference parts of the object. We statistically quantify the dissimilarities and order the suggestions accordingly, so that users can easily find the required corrections without exploring all of them. For displaying the suggestions, we use the radial-tree view with links, which offers the necessary functionality for exploring vast hierarchies of elements.
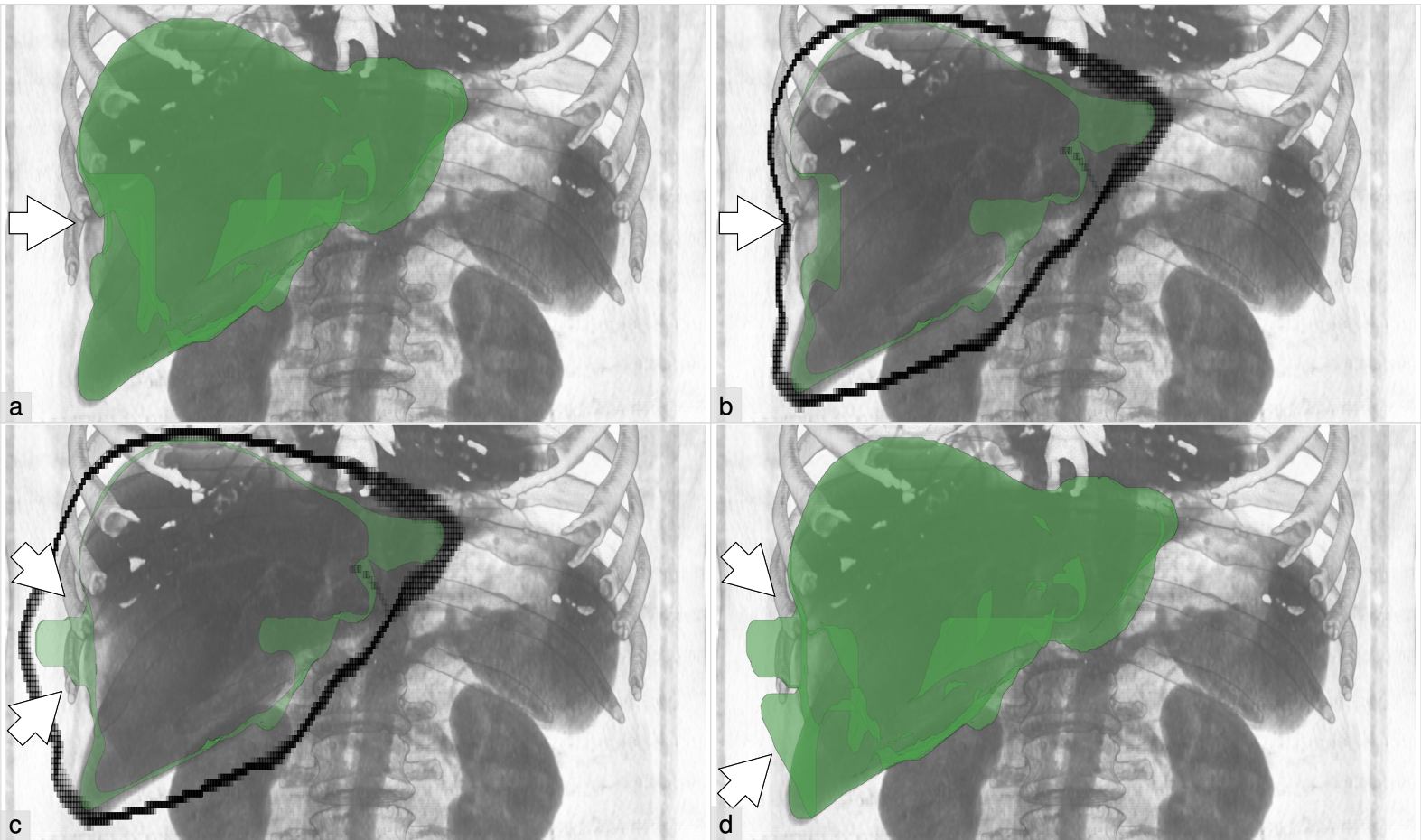
Figure 3.35: Example of a U_P defect, which is not properly captured by the skeleton of the complement. a) The segmentation mask with the U_P defect. b) Slab of the segmentation mask and the skeleton of the object's complement. c) The same slab as in (b) after applying the suggested correction operations; severe over-estimation defects appear as a result. d) The result of editing the U_P defect is unsatisfactory: the U_P defect is solved, new defects are introduced though.

CHAPTER 4

Statistics-Driven Localization of Dissimilarities in Data

*Do not put your faith in what statistics say until you
have carefully considered what they do not say.*

— William W. Watt

This chapter is based on the following technical report:
Alexey Karimov, Gabriel Mistelbauer, Thomas Auzinger and Eduard Gröller. Statistics-Driven Localization of Dissimilarities in Data. *Technical Report TR-186-2-16-1. Institute of Computer Graphics and Algorithms, Technische Universität Wien. April 2016* [88].

The identification of dissimilar regions in spatial and temporal data is a fundamental part of data exploration. This process takes place in applications, such as biomedical image processing as well as climatic data analysis. We propose a general solution for this task by employing well-founded statistical tools. From a large set of candidate regions, we derive an empirical distribution of the data and perform statistical hypothesis testing to obtain p-values as measures of dissimilarity. Having p-values, we quantify differences and rank regions on a global scale according to their dissimilarity to user-specified exemplar regions. We demonstrate our approach and its generality with two application scenarios, namely interactive exploration of climatic data and segmentation editing in the medical domain. In both cases our data exploration protocol unifies the interactive data analysis, guiding the user towards regions with the most relevant dissimilarity characteristics. The dissimilarity analysis results are conveyed with a radial tree, which prevents the user from searching exhaustively through all the data.

A FUNDAMENTAL task in data analysis is the evaluation of internal data consistency by identifying similar and dissimilar subregions inside spatial and temporal data. Manual inspection is generally time-consuming or infeasible due to the amount of data usually encountered in application scenarios such as biomedical image processing, climate studies, *etc.* Thus, the user requires (semi-)automatic search tools. A vast array of domain-specific data analysis tools is available from a large body of research.

In this work we propose a general framework for this task based on sound statistical methods. Thus, our focus lies in the proper and efficient use of statistics in data analysis while still providing sufficient generality of the method with respect to application domains. Having constructed empirical data-value distributions from various regions in the input data, we perform a **statistical hypotheses testing** to compute a **p-value** per region that is used as a dissimilarity measure. We discuss related statistical aspects and provide a practical approach for region aggregation to arrive at a hierarchical ordering of the relative dissimilarities in the data.

We present **visualization means** to assist the user in an interactive exploration of the dissimilarities in the input data. We formulate a **data exploration protocol** which takes advantage of available statistical information on the dissimilarities. A **radial tree** of the hierarchical statistical data guides the user towards potential regions of interest. In case of data change, the global data consistency is tracked via a suitable **timeline plot**.

We apply our framework to two qualitatively different scenarios: temporal data exploration and segmentation editing of volumetric data. We provide the realizations of our abstract concepts in the specific domains. An evaluation of our method's efficacy is given.

4.1 Related Work

Several works in the visualization and image processing domains involve statistical comparison of regions in volume data in order to detect features or highlight irregularities. In particular, the task of classifying samples of volume data into certain features or materials usually involves such a comparison. Kniss *et al.* [89] discuss a statistical classification procedure that they later apply to rendering. Using a classifier, the method assigns to each sample a probabilistic likelihood that it exhibits certain features of the data. The classifier may have parameters which are evaluated on a training set. The probabilistic likelihoods are then transformed to the posterior probabilities with the Bayes Rule. Based on the probabilities, each sample is fuzzily classified. Each class exhibits various attributes, *e.g.*, transfer functions. This technique provides an efficient classification of the features, but requires classifiers, parameter estimation, feature models, and prior class probabilities beforehand. Our approach requires only the definition of regions in order to statistically compare them with a set of references.

Tasdizen *et al.* [90] improve tissue classification in Magnetic Resonance Imaging (MRI) by using the data-value distribution in a certain neighborhood around a voxel. They compute the probabilities of observing particular data values in the neighborhood if the voxel belongs to a certain class. The entropy is calculated over the probabilities of the voxel to belong to each possible class. It is iteratively minimized while the classification of voxels changes. This method

requires an initial classification that the authors obtain by co-registering the data with a digital atlas. In our approach we do not need this potentially complex step, as we get the references for the statistical comparison directly from the input data.

Lundström *et al.* [91] use partial range histograms to introduce a classification certainty as the second dimension in 2D transfer functions. These histograms collect data values of a certain range in a cubic neighborhood. The ranges of values are determined by fitting Gaussian distributions to the global histogram of the data. The partial range histograms are successively subtracted from the global histogram. While this method improves the classification in regions where ranges of different materials or tissues overlap, it does not explicitly test statistical hypotheses, *i.e.*, whether certain materials or tissues are present in the region of interest. With our statistical approach one can test such hypotheses and make decisions on test results. Heinzl *et al.* [92] extend this work by calculating probabilities of different materials at each data sample. However, a Gaussian distribution of the data values is still assumed.

Johnson and Huang [93] detect features with distribution queries. The data used for forming a query is sampled in the neighborhood of each voxel. Then, the user specifies intervals of data values that are of interest. The filtered data is organized into a histogram that represents the target distribution. The user composes a query with clauses on histogram bins. It is fuzzily matched with the actual volume data. The queries may involve statistics on the histograms, such as standard deviation, skewness, covariance, *etc.* The user assigns to the queries various attributes, *e.g.*, transfer functions. With the rendered image, the user may check hypotheses on the volume data, formulated in terms of the queries. However, this work does not include any statistical tests, which, in certain scenarios, are required to make decisions based on the volume data.

Saad *et al.* [94] investigate the uncertainty of a segmentation compared to expert segmentations in medical image processing. They first construct an atlas that involves two kinds of histograms: likelihood versus shape and likelihood versus appearance. The atlas is constructed with training data from experts. A multivariate Gaussian distribution then models the appearance of the features in the volume data. Prior to the uncertainty analysis, the input-volume data is registered to the atlas. Using the Bayes theorem, voxels are classified with the first-best-guess logic. Finally, the discrepancies between actual segmentation and atlas data are conveyed to the user. This reveals regions of misclassification and abnormalities in the volume data. We propose to capture shape information via a special definition of regions that will be explained in Section 4.4.2. We collect the data this way instead of using prior shape information stored in an atlas and which is only applicable after co-registration.

Haidacher *et al.* [75] assume that the data values have Gaussian distributions, whose parameters they iteratively evaluate while growing a sphere around each voxel. They use the Jarque-Bera test for normality [95]. Welch's T-test [96] checks, whether the neighborhood sphere and its hull have the same Gaussian distribution. The resulting mean μ and standard deviation σ are used to modulate the opacity of samples and their shading, so the impact of noise is reduced. The user defines 2D transfer functions in the μ versus σ space. This improves the classification of materials or tissues during volume rendering. The drawback here lies in the fixed distribution model that is fitted to the data. In our approach we do not assume any distribution model, relying instead on

empirical distribution functions and statistical tests on them. These functions approximate the underlying data-value distribution.

Praßni *et al.* [30] investigate the usage of shape information in the classification. They represent an object-of-interest in the data with a curve skeleton. The object is split into skeleton regions by corresponding skeleton segments. Tubiness, surfaceness, and blobbiness are used as additional dimensions for the classification of regions. In contrast to computing such spatial properties, we propose to collect and statistically compare data values in regions that are similar to the skeleton regions.

Comparison of regions is also utilized in segmentation(-editing) techniques, which delineate the object-of-interest from the rest of the data. For example, results of the comparison may identify segmentation defects. Our technique from Chapter 3 detects segmentation defects by analyzing dissimilarities in data values. The object, defined with the segmentation mask, is represented as a set of regions aligned with the skeleton of the object. Inside each such region a histogram collects the data values. The histogram is compared to those from adjacent regions in order to reveal data dissimilarities, associated with segmentation defects. However, this approach cannot tell whether a certain region is actually part of the object. It just delineates the object from the defects, having detected certain dissimilarities. We provide this missing functionality with our framework and guide the user towards regions that are relevant for editing. This application scenario will be discussed in Section 4.4.2.

Some methods are designed for temporal data exploration. Among them is the technique of Hochheiser and Shneiderman [97] that enables the user to filter temporal data, composed of numerous time series, with search constraints. The time series are displayed in a two-dimensional plot. Each constraint is defined by a user-drawn rectangle – a timebox. It only matches time series which have data values inside the corresponding rectangle. Having specified several timeboxes, the user gets an uncluttered view of time series of interest. With our approach we may formulate such a filter as a statistical test of similarity to the data in the timebox. However, we leave such extended functionality for future work.

Buono *et al.* [98] proposes a pattern search for temporal data analysis. It is based on the Euclidean distance between corresponding values of two compared time series. To improve the pattern matching, the distance metric includes four additional transformations, which normalize the compared data. The user gets an overview of all regions matched with the specified pattern. Our method performs a statistical test of similarity of data values in sub-regions of a time series and reports the resulting p-values. As we aggregate regions with similar statistical significance properties, the user can explore the data at different levels of detail.

Buono *et al.* [99] focus on the prediction of a time series using patterns found in the temporal data to make extrapolations into the future. Having constructed the empirical distribution functions of the underlying data, one may simulate future data values with our approach. An implementation, however, would require domain-specific knowledge. We leave such extension for future work.

The work of Bögl *et al.* [100] discusses a Visual Analytics approach to the selection of appropriate models for time series data. The user specifies the model by interactively adjusting parameters and selecting characteristic values in a timeline plot. During this specification, the remaining

model parameters are estimated. Then, the method provides the user with an in-depth analysis of the selected model's fitness to the data. Various models are compared by their information content. The user selects the most informative one that later can be used for prediction.

Applied to temporal data, our approach solves a different task. It detects the most similar or the most dissimilar occurrences, compared to the exemplar events. Even though the results of our technique are not directly applicable for prediction, they may be useful for finding causes which lead to certain occurrences with respect to the background information.

4.2 Method

In this section we provide an in-depth description of our approach. We start with *atomic* regions, which are the basis of our method. Inside each atomic region we construct an *empirical distribution function* of data values. We statistically test dissimilarity between data in the atomic regions and values in a set of *markers*, *i.e.*, user-specified exemplars of regions of interest. P-values of the dissimilarity significance are computed. Unions of atomic regions form *composite* regions, which exhibit either significant or non-significant dissimilarities to the markers. The composite regions form a *hierarchy*, varying in size and representing the data with different levels of detail. To support the data exploration task, we provide the user with a *radial tree view* depicting the hierarchy of the composite regions and the derived p-values. Having linked visualizations of the data and the radial tree view, the user follows our *data-exploration protocol*. To support the user in scenarios with dynamically changing data, we display *global statistics on the p-values*. An overview of our method is given in Figure 4.1.

4.2.1 Requirements for Statistical Testing

The major requirement of our approach involves the definition of the *atomic* regions. These regions should reflect domain-specific features of the data. The atomicity of the region means that it should contain samples of only one single feature. In other words, each atomic region belongs to a single feature only, *i.e.*, the data-value distribution is assumed to be identical over the region. The atomicity is usually achieved by making the regions small enough. Yet, each region should contain enough samples to properly reconstruct the underlying data-value distribution. In order to provide exact p-values, we restrict the atomic regions to be non-overlapping. To preserve the adjacency of the data at the level of atomic regions, each region should be connected. An atomic region realization that satisfies all the aforementioned requirements will provide meaningful results in terms of statistics. The composite regions then represent the domain-specific features of the data on different levels of detail. The computed p-values differentiate the features of interest from the rest of the data.

Our work concentrates on the statistical framework that operates on top of the realized atomic regions. We provide the realization of atomic regions for the general case as well as the specialized realization of atomic regions for three-dimensional spatial data. Moreover, we validate these realizations in two application scenarios.
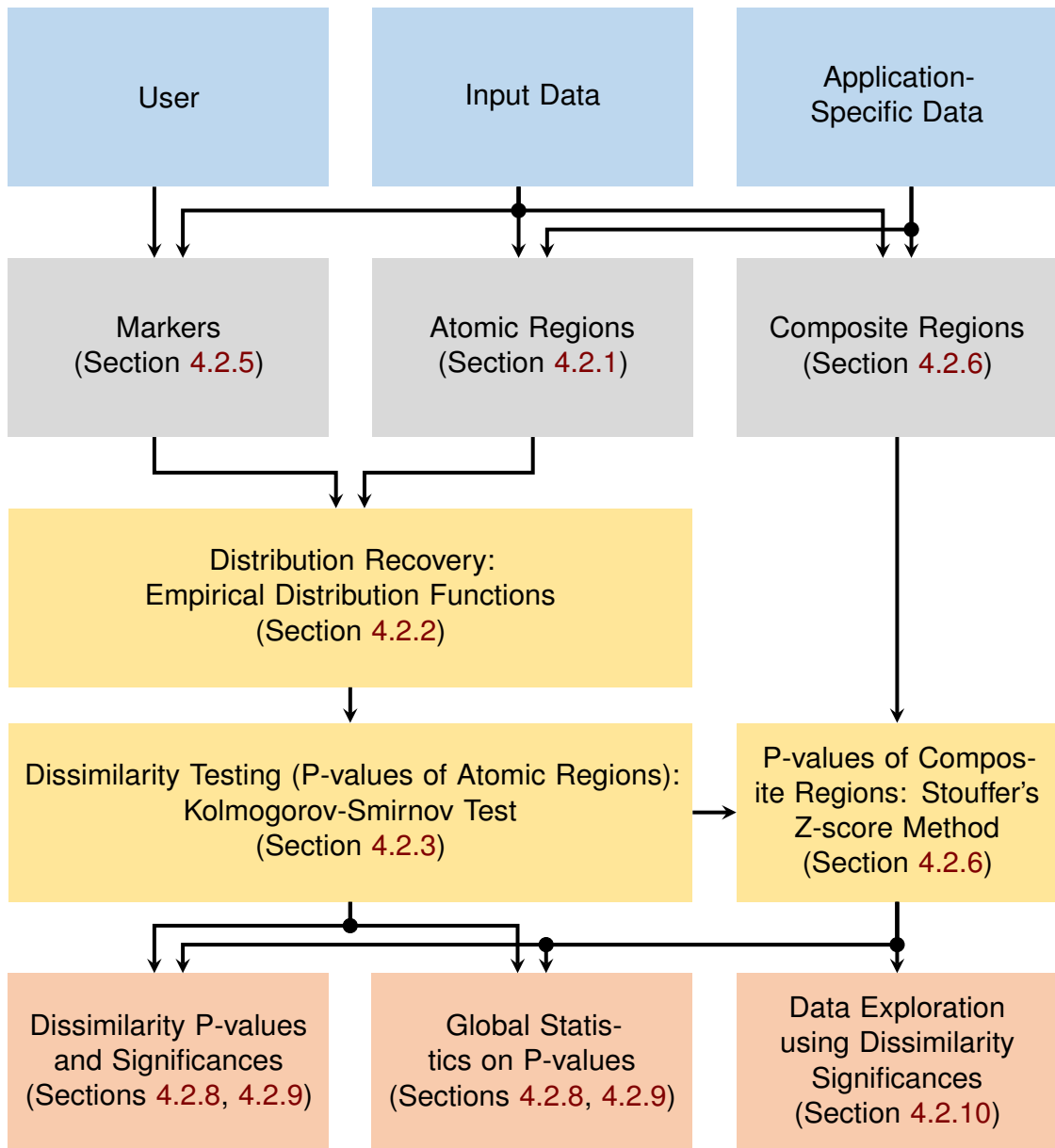
Figure 4.1: Overview of our statistical method. The information from the data sources is transformed into our internal entities (markers, atomic regions, and composite regions). We recover the underlying data-value distributions in the atomic regions and the markers with empirical distribution functions. Then we perform a statistical test on the significance of dissimilarities between the empirical distribution functions of the regions. We present this information as well as global statistics on p-values to the user. Moreover, we propose a data exploration protocol based on the dissimilarity significance information.

The fundamental property of our framework is its ability to handle stochastic data. Our input data can stem from a spatially and temporally varying probability distribution. For real-world data sources this is generally the case. Natural local variations and inhomogeneities in the data values, acquisition noise, discrete and finite measurement domain and time introduce random fluctuations in the data. We do not assume any knowledge of the underlying distribution and operate on samples from it (*e.g.*, image of a Computed Tomography (CT) scan or temperature measurements in climatic data). Even if the data does not exhibit a stochastic nature, our approach is valid, as we operate on data-value distributions that characterize the various data regions and which are tested for dissimilarity.

In this work we focus on scalar data values. An extension to multi-dimensional data is possible, but we consider the corresponding implementation of the statistical concepts to be future work. The dimensionality of the data domain can be arbitrary though. The density values from medical imaging (CT, MRI) are given on a three-dimensional grid. Abstract time series data consists of samples with one temporal dimension. In the following section we reconstruct the underlying data-value distribution from the samples.

4.2.2 Distribution Recovery

Given a one-dimensional random variable V and an associated probability function $F(\mathbf{s})$ that varies over an underlying space $\mathbf{s} \in \mathcal{S}$, we take as input data a single outcome $o(\mathbf{d})$ of V on a discrete subset $\mathbf{d} \subseteq \mathcal{D}$ of $\mathcal{S}$. We do not impose any structural constraints on $\mathcal{D}$. In most applications it is isomorph to a structured grid in an m-dimensional space $\mathcal{S}$. Imaging processes, for example, yield two-dimensional and three-dimensional regular grids of data, whereas temporal data is usually taken at equidistant points in time.

The key idea of our approach is to recover the unknown probability function F by collecting outcome data on certain subsets of $\mathcal{D}$ – the atomic regions. We are interested in atomic regions that are similar or dissimilar to the user-specified markers. The similarity is determined by comparing the *Empirical Distribution Functions (EDFs)* that are the distribution functions of the outcome values of V in the regions. We choose the EDFs as we do not assume that the outcome values were drawn from a certain probability distribution (even with unspecified parameters). Instead, we could have selected a distribution model in advance and fitted it to the data. However, this would limit the generality of our method, as the choice of the appropriate distribution model depends on the data modality. Having estimated the parameters of the distribution model from the data, we could not directly apply classical statistical tests [101]. We could have used kernel density estimation (KDE) methods to model the unknown probability density function. This would require a suitable kernel function. Also, the classical statistical tests would need adaptation (*e.g.*, the two-sample shape test on the KDE by Duong *et al.* [102]). With the EDFs we perform the tests directly.

For an atomic region $\mathcal{R} \subset \mathcal{D}$ we build the corresponding EDF $\hat{F}_{\mathcal{R}}$ from the outcomes in $\mathcal{R}$ by

$$\hat{F}_{\mathcal{R}}(t) = \frac{1}{|\mathcal{R}|} \sum_{i=1}^{|\mathcal{R}|} \mathbf{1}\{o(\mathbf{d}_i) \leq t\}, \quad \mathbf{d}_i \in \mathcal{R}, \tag{4.1}$$

$$\mathbf{1}\{E\} = \begin{cases} 1, & \text{if condition } E \text{ is true} \\ 0, & \text{otherwise} \end{cases}, \tag{4.2}$$

where $|\mathcal{R}|$ is the cardinality of $\mathcal{R}$. Figure 4.2 illustrates this concept. In order to construct the EDF, Equation 4.1 requires that the values $o(\mathbf{d}_i)$ are sampled from identical and mutually independent random variables that correspond to the sampling locations $\mathbf{d} \subseteq \mathcal{D}$. The atomic region definition already guarantees that the random variables have the *identical* distribution function $F_{\mathcal{R}}$ (Section 4.2.1). A set of random variables Ω is *mutually independent* if the probability of any certain outcome of one variable does not depend on outcomes of the remaining variables. Formally, this means that $\forall \{A_1, \ldots, A_M\} \subseteq \Omega$ and $\forall \{a_1, \ldots, a_M\} \in \mathbb{R}^M$ the following statement holds true:

$$Pr\left(\bigcap_{i=1}^{M}\{A_i \leq a_i\}\right) = \prod_{i=1}^{M} Pr\left(\{A_i \leq a_i\}\right). \tag{4.3}$$

This condition is satisfied in the majority of data acquisition settings. For example, in medical imaging data certain physical properties of tissues are measured, such as density or echogenicity. These properties are measured locally at each sampling location independent of the data values from neighboring sampling locations. The independence could be violated due to correlation in the acquisition noise. However, we found this effect to be negligible for our use-cases. As a result, the required EDF is available at each atomic region and can be used for dissimilarity detection.

4.2.3 Dissimilarity Testing

To quantify the data dissimilarities in two non-overlapping regions $\mathcal{R}_1$ and $\mathcal{R}_2$, we formulate the problem as a statistical hypothesis on the similarity of their respective EDFs. The null hypothesis H_0 is that both EDFs $\hat{F}_{\mathcal{R}_1}$ and $\hat{F}_{\mathcal{R}_2}$ follow the same unspecified distribution. In order to detect a possible violation of H_0 (distribution shape), we employ the *two-sample Kolmogorov-Smirnov test (KS)* with *statistic D_n* [103, 104]:

$$D_n = \sup_{t \in \mathbb{R}} |\hat{F}_{\mathcal{R}_1}(t) - \hat{F}_{\mathcal{R}_2}(t)|, \tag{4.4}$$

$$n = \frac{|\mathcal{R}_1||\mathcal{R}_2|}{|\mathcal{R}_1| + |\mathcal{R}_2|}, \tag{4.5}$$

which is a random variable. We test the null hypothesis H_0 by computing the p-value pv as the conditional probability:

$$\text{pv} = Pr\left(D_n \geq d \,|\, H_0\right), \tag{4.6}$$

where d is a value of the statistic D_n, computed with the EDFs $\hat{F}_{\mathcal{R}_1}$ and $\hat{F}_{\mathcal{R}_2}$ of the outcome values $o(\mathbf{d}_i)$. This statistical test is illustrated in Figure 4.2.
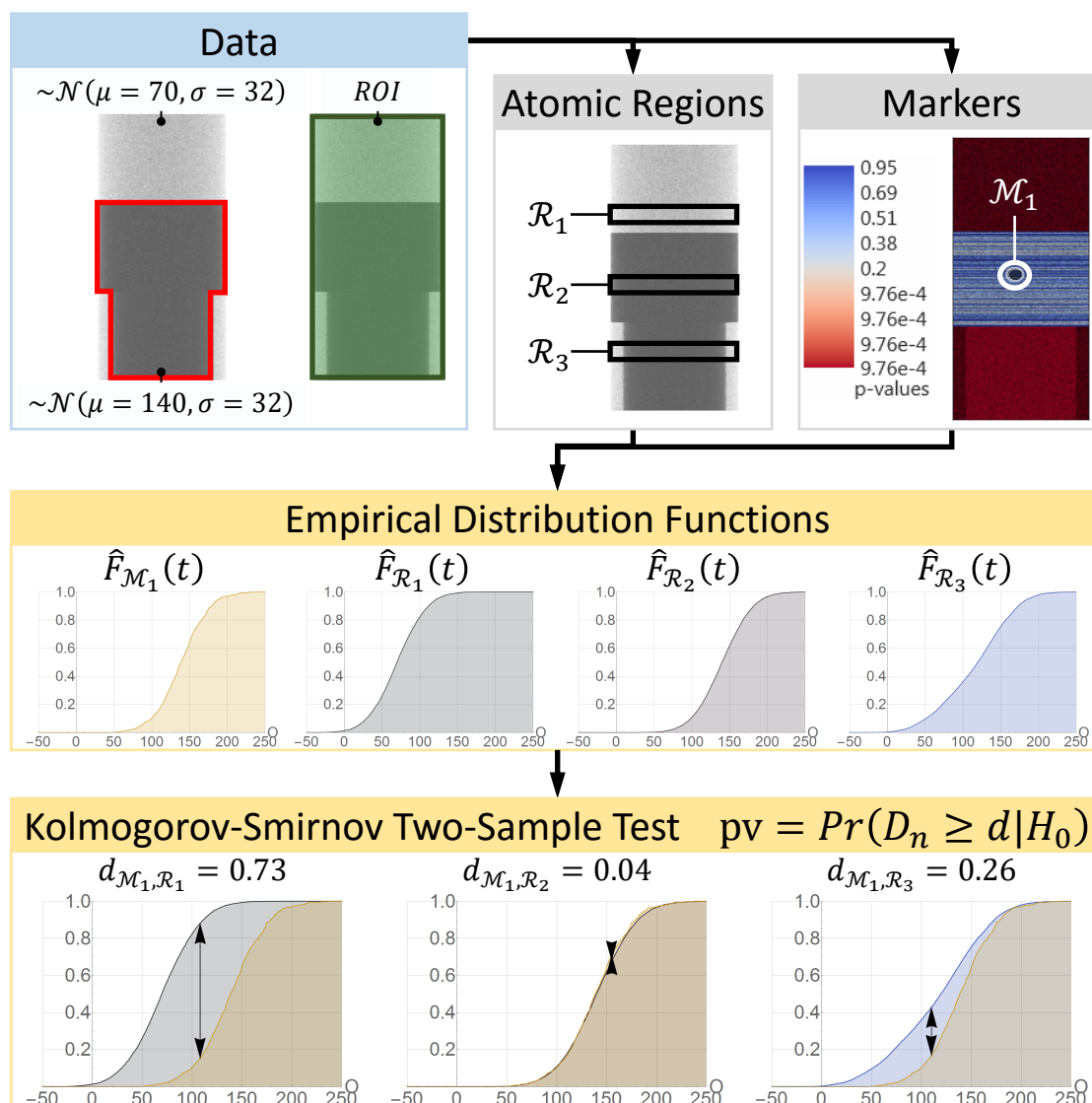
Figure 4.2: Our method applied to a three-dimensional phantom dataset. The data includes an actual object (red) in the region of interest (green), where the user would like to evaluate the dissimilarities. The region of interest is split into the atomic regions $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \ldots$, according to the realization that will be explained in Section 4.4.2. Once the user has specified a single marker $\mathcal{M}_1$, our method builds empirical distribution functions of the data values in the marker and the atomic regions. Our method employs the Kolmogorov-Smirnov test to compute the p-values of the dissimilarity significance between the atomic regions and the marker.

In common scientific practice, a *significance level* α is chosen a-priori. The null hypothesis H_0 is rejected if pv $\leq \alpha$. Here, α determines the probability of falsely rejecting H_0. It is usually set to 0.05, however, it may be different in various scientific domains. In our approach we do not employ a single global significance level, but directly report the p-values to the user. This allows an inspection of the dissimilarity significance of all tested region pairs.

Generally, computing the Kolmogorov distribution D_n is complicated due to a lack of closed-form solutions. Existing approaches often operate with only a one-sample test, where the EDF $\hat{F}_{\mathcal{R}_1}$ is matched against a fully specified distribution. However, we require the more general two-sample test which checks whether two samples were drawn from the same unspecified distribution.

The KS test requires that the hypothetical distribution is continuous, and, therefore, there are no ties in the sampled data (no same values). As we do not assume any restriction on the hypothetical distribution, it can be either continuous or discrete. If the hypothetical distribution is discrete, p-values reported by the KS test are inaccurate [105]. Also, if ties exist in the sampled data, the test yields inaccurate p-values. For generality, we allow the ties.

As stated by Arnold and Emerson [105], in case of a two-sample test with a discrete hypothetical distribution, the distribution of the test statistic depends on this unspecified distribution. In such a situation, a resampling can relax the requirements of continuity and no ties in the data, so the statistical tests can be performed. This was demonstrated by Dufour and Farhat [106] for the KS test.

The resampling of the outcome values $o = \{o_1, \ldots, o_M\}$ is achieved via bootstrap, permutation, or randomized (Monte Carlo) tests. Let us assume the following $\mathcal{E}$ new samples of the outcome values are drawn: $r_1, \ldots, r_{\mathcal{E}}$. Each new sample contains the same number of the outcome values as the original sample o. The permutation test generates all possible samples without replacement out of o. The bootstrap test iterates over all possible samples with replacement from o. The randomized tests simulate $\mathcal{E}$ samples (with or without replacement) from o. Then the test statistic is computed on the original sample (d_o) and the new samples ($d_{r_1}, \ldots, d_{r_{\mathcal{E}}}$). Assuming that D_n is a random variable of the test statistic value, the p-value of the test is calculated as follows:

$$\text{pv} = Pr\left(D_n \geq d_o \,|\, H_0\right) = \frac{1 + \sum_{i=1}^{\mathcal{E}} \mathbf{1}\{d_{r_i} \geq d_o\}}{1 + \mathcal{E}}. \tag{4.7}$$

The permutation and bootstrap tests have the same asymptomatic power for the KS test, according to Praestgaard [107], and report true p-values. However, taking into account all possible samples requires enormous computational efforts. To alleviate this issue, the randomized tests probe only some randomly drawn samples, and converge to true p-values given enough samples. According to Manly [108], the smallest recommended number of samples is 1000 for the significance level $\alpha = 0.05$, however, Jackson and Somers [109] propose a minimum of $\mathcal{E} = 10000$. We choose the randomized permutation KS test with at least 1000 samples. Next, we discuss alternative tests. Then, we continue with the comparison of data from the atomic regions with the reference values in the markers.

4.2.4 Alternative Statistical Tests for Dissimilarity

Alternative two-sample distribution shape tests include the Chi-Squared test, Cramér-von Mises test (CvM), and Anderson-Darling test (AD). The Chi-Squared test uses binned data. We assume that the data values from the tested regions $\mathcal{R}_1$ and $\mathcal{R}_2$ are binned into the following N_B bins $\mathcal{H}[1], \ldots, \mathcal{H}[N_B]$. The l-th bin counts we denote as $\mathcal{H}_{\mathcal{R}_1}[l]$ and $\mathcal{H}_{\mathcal{R}_2}[l]$ respectively. The binning is usually done in such a way that each bin contains enough samples (more than five). The optimal binning depends on the unspecified hypothetical distribution. The test uses the statistic X [110][pp. 616–617] that follows a Chi-Squared distribution with N_B degrees of freedom under the null hypothesis:

$$X = \sum_{l=1}^{N_B} \frac{(K_1 * \mathcal{H}_{\mathcal{R}_1}[l] - K_2 * \mathcal{H}_{\mathcal{R}_2}[l])^2}{\mathcal{H}_{\mathcal{R}_1}[l] + \mathcal{H}_{\mathcal{R}_2}[l]}, \tag{4.8}$$

$$K_1 = \sqrt{\frac{|\mathcal{R}_2|}{|\mathcal{R}_1|}}, \tag{4.9}$$

$$K_2 = \sqrt{\frac{|\mathcal{R}_1|}{|\mathcal{R}_2|}}. \tag{4.10}$$

The test reports the p-value for the null hypothesis H_0. However, the Chi-Squared test does not suit our purpose well due to the necessity of data binning.

The KS, the CvM and the AD tests assume a continuous hypothetical distribution. The CvM test uses the statistic W^2 [111]:

$$W^2 = \frac{|\mathcal{R}_1||\mathcal{R}_2|}{|\mathcal{R}_1| + |\mathcal{R}_2|} \int_{-\infty}^{\infty} [\hat{F}_{\mathcal{R}_1}(t) - \hat{F}_{\mathcal{R}_2}(t)]^2 \, d\hat{F}_{\mathcal{R}_1 + \mathcal{R}_2}(t), \tag{4.11}$$

$$\hat{F}_{\mathcal{R}_1 + \mathcal{R}_2}(t) = \frac{|\mathcal{R}_1|\hat{F}_{\mathcal{R}_1}(t) + |\mathcal{R}_2|\hat{F}_{\mathcal{R}_2}(t)}{|\mathcal{R}_1| + |\mathcal{R}_2|}. \tag{4.12}$$

The AD test computes the statistic A^2 [112]:

$$A^2 = \frac{|\mathcal{R}_1||\mathcal{R}_2|}{|\mathcal{R}_1| + |\mathcal{R}_2|} \int_{-\infty}^{\infty} \omega(t) [\hat{F}_{\mathcal{R}_1}(t) - \hat{F}_{\mathcal{R}_2}(t)]^2 \, d\hat{F}_{\mathcal{R}_1 + \mathcal{R}_2}(t), \tag{4.13}$$

$$\omega(t) = \frac{1}{\hat{F}_{\mathcal{R}_1 + \mathcal{R}_2}(t)[1 - \hat{F}_{\mathcal{R}_1 + \mathcal{R}_2}(t)]}. \tag{4.14}$$

All three statistics D_n, W^2, A^2 can be used in the permutation, bootstrap, and randomized tests in case of a discrete hypothetical distribution or ties in the sampled data. The comparison of the randomized versions of the KS and the CvM tests, conducted by Dufour and Farhat [106], shows little difference in the power of these tests. The authors state that the KS test is more conservative in rejecting the null hypothesis H_0 than the CvM test. In case of a discrete hypothetical distribution, the latter rejects the null hypothesis H_0 more often than the specified significance level. The AD test exhibits a better sensitivity to differences in the tails of the distributions than the KS test, as it is using a weighting factor $\omega(t)$. Feigelson and Babu [113] mention the following

commonly overlooked restrictions of the KS test: the independence of the two tested samples and applicability to one dimension only. The same restrictions also apply to the CvM and the AD tests. To keep computational efforts low, we favor the KS test.

4.2.5 Markers

In most applications, the user is interested in the comparison of the regions with certain reference regions, rather than in the comparison of all possible region pairs. This enables data exploration tasks where reference regions are selected and the most dissimilar (objective O1) or the most similar (objective O2) regions are identified.

As a direct specification of the reference EDF is not possible, we have to estimate it from several reference regions, which we subsequently call *markers*. A marker $\mathcal{M}$ is a connected subset of the sampling space $\mathcal{D}$ of the input data. The EDFs of all the markers approximate the unknown reference distribution function. Given l *disjoint* markers $\mathcal{M}_1, \ldots, \mathcal{M}_l$, we compute the dissimilarity significance for each region $\mathcal{R}_i$ and each marker $\mathcal{M}_j$ by evaluating Equation 4.4. This gives us a p-value $\mathrm{pv}_{i,j}$ for each such pairing and assigns l p-values to each region. Note that due to their independence requirement, the statistical tests are only valid on disjoint regions. Thus, for each of the aforementioned regions, its intersection with the marker support has to be removed prior to the tests. If the number of samples in the region $\mathcal{R}_i$ is too low for calculating statistics, the corresponding p-values cannot be computed exactly. Denoting the region $\mathcal{R}_i$ without the marker support with $\mathcal{R}'_i$, we check the following criteria:

$$|\mathcal{R}'_i| < S_0, \tag{4.15}$$

$$\frac{|\mathcal{R}'_i|}{|\mathcal{R}_i|} < S_1, \tag{4.16}$$

where S_0 is the absolute minimal count and S_1 is the minimal percentage. If any of the criteria is satisfied, we leave the p-values $\mathrm{pv}_{i,1}, \ldots, \mathrm{pv}_{i,l}$ undefined.

Since we are interested in a single p-value pv_i for each region $\mathcal{R}_i$ from all the markers, we combine the p-values $\mathrm{pv}_{i,1}, \ldots, \mathrm{pv}_{i,l}$ of the KS tests. We formulate the task as follows: combine p-values $\mathrm{pv}_1, \ldots, \mathrm{pv}_l$ into a **compound p-value** pv. Nichols *et al.* [114] suggest to combine the individual null hypotheses $H_{0(1)}, \ldots, H_{0(l)}$ with a logical conjunction. The compound null hypothesis $H_0 = H_{0(1)} \wedge \ldots \wedge H_{0(l)}$ is then tested. The corresponding compound p-value pv is computed as follows: $\mathrm{pv} = \max(\mathrm{pv}_1, \ldots, \mathrm{pv}_l)$. This is based on the following logic: the compound p-value is significant only if each individual p-value is significant. The region has significant dissimilarity only if it is significantly dissimilar to all the markers. For each region $\mathcal{R}_i$ we compute its p-value pv_i as

$$\mathrm{pv}_i = \max(\mathrm{pv}_{i,1}, \ldots, \mathrm{pv}_{i,l}). \tag{4.17}$$

If the region $\mathcal{R}_i$ has insufficient cardinality, then none of the p-values $\mathrm{pv}_{i,1}, \ldots, \mathrm{pv}_{i,l}$ is defined. In this case, we do not define the compound p-value pv_i as well. In the following exposition, we introduce the concept of composite regions for the data representation.
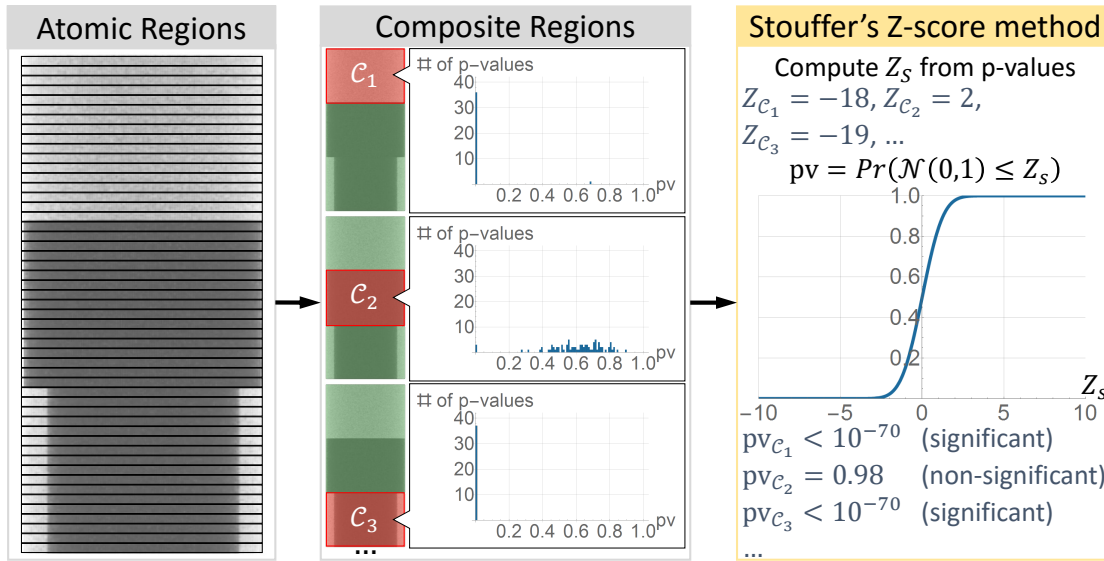
Figure 4.3: Computation of the p-values for the composite regions of the phantom data from Figure 4.2. The realization of the composite regions will be discussed in Section 4.4.2. The composite regions span several atomic regions. We show only three composite regions $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$, but there are more. For these three composite regions we present histograms of the p-values from the underlying atomic regions. For each composite region, we combine these p-values with Stouffer's Z-score method into a single p-value. The resulting p-value refers to the compound null hypothesis H_0 being true for all atomic regions within the composite region.

4.2.6 Composite Regions

In practice, the number of locations $\mathbf{d}_i$ at which the outcome values $o(\mathbf{d}_i)$ are sampled is usually large. Common medical or biological imaging output, for example, is given on regular grids with millions of cells with data values. This results in a large number of regions where we compute dissimilarity significances. Without additional means, the user would have to investigate all of them during the data exploration, which hinders the analysis task.

To alleviate this issue, we combine the p-values of certain regions into a compound p-value of the regions' union. Figure 4.3 illustrates such a combination. Given a set of k *disjoint* regions $\mathcal{R}_1, \ldots, \mathcal{R}_k$, we state the following compound null hypothesis: the respective EDFs $\hat{F}_{\mathcal{R}_1}, \ldots, \hat{F}_{\mathcal{R}_k}$ follow the same distribution as the EDF of the markers $\mathcal{M}_1, \ldots, \mathcal{M}_l$. For each atomic region $\mathcal{R}_i$ we have computed the p-value pv_i (Equation 4.17). We now combine $\mathrm{pv}_1, \ldots, \mathrm{pv}_k$ into the p-value for the compound null hypothesis. For combining the p-values in case of the multiple markers, we use the logical conjunction scheme. However, it cannot be applied here, because the compound region is supposed to have significant dissimilarity if any of its subregions is significantly dissimilar to the markers. Therefore, we use Stouffer's Z-score method [115] with

the statistic Z_S:

$$Z_S = \frac{\sum_{i=1}^{k} Z_i}{\sqrt{k}}, \tag{4.18}$$

$$Z_i = \inf\{t \mid \mathrm{pv}_i \leq Pr(\mathcal{N}(0,1) \leq t)\}, \tag{4.19}$$

which follows a normal distribution $\mathcal{N}(\mu = 0, \sigma = 1)$. Each combined p-value is assumed to be drawn from a normally distributed random variable $Z_i \sim \mathcal{N}(0,1)$ (Equation 4.19). The combination method requires these random variables to be independent. Therefore, the p-values should come from independent tests, otherwise the resulting p-value is inaccurate. We ensure this requirement by testing only disjoint regions in the KS test.

As the sum of independent normal distributions is a normal distribution itself, a p-value of Z_S can be computed, which represents the p-value for the compound null hypothesis:

$$\mathrm{pv} = Pr(\mathcal{N}(0,1) \leq Z_S). \tag{4.20}$$

Later, Liptak, Mosteller and Bush introduced weights $\omega_1, \ldots, \omega_k$ for the combined p-values [116, 117]:

$$Z_W = \frac{\sum_{i=1}^{k} \omega_i Z_i}{\sqrt{\omega_1^2 + \ldots + \omega_k^2}}. \tag{4.21}$$

The test statistic Z_W also follows a normal distribution $\mathcal{N}(0,1)$, so we obtain a p-value for the compound null hypothesis. The choice of the weights ω_i is an open question, *e.g.*, Whitlock suggests to use the degrees of freedom from preceding statistical tests [118]. In our case, the KS test and the Nichols *et al.* [114] combination formula do not have any degrees of freedom, therefore, we cannot assign the weights ($\omega_1 = \ldots = \omega_k = 1$). One minor modification we introduce relates to atomic regions with undefined p-values. If the p-value pv_i is not defined, then we omit it during the test. If all p-values $\mathrm{pv}_1, \ldots, \mathrm{pv}_k$ are not defined, then we do not define the p-value for the compound region as well.

By partitioning the sample locations $\mathcal{D}$ into a set of disjoint *atomic* regions $\mathcal{R}_i$, the p-values of arbitrary regions in $\mathcal{D}$ can be computed as long as such regions are represented as unions of atomic regions. In this sense, the p-values of the atomic regions form a basis from which the p-values of all possible unions can be computed with Stouffer's method. We refer to such unions as *composite* regions $\mathcal{C}_1, \ldots, \mathcal{C}_h$. The p-value of the composite region $\mathcal{C}_j$, spanning over the atomic regions $\mathcal{R}_{j_1}, \ldots, \mathcal{R}_{j_k}$, is computed with Equations 4.19, 4.20. To facilitate the data exploration, we suggest that the composite regions vary in cardinality and represent the data with different levels of detail. Together the composite regions form a *hierarchy*. Next, we discuss alternative combination methods. Then, we continue with ranking the composite regions.

4.2.7 Alternative Combination Methods

There are alternative methods to combine p-values $\mathrm{pv}_1, \ldots, \mathrm{pv}_k$ into a p-value pv for the compound null hypothesis. Winkler *et al.* [119] mention the approach of Nichols *et al.* [114], Stouffer's
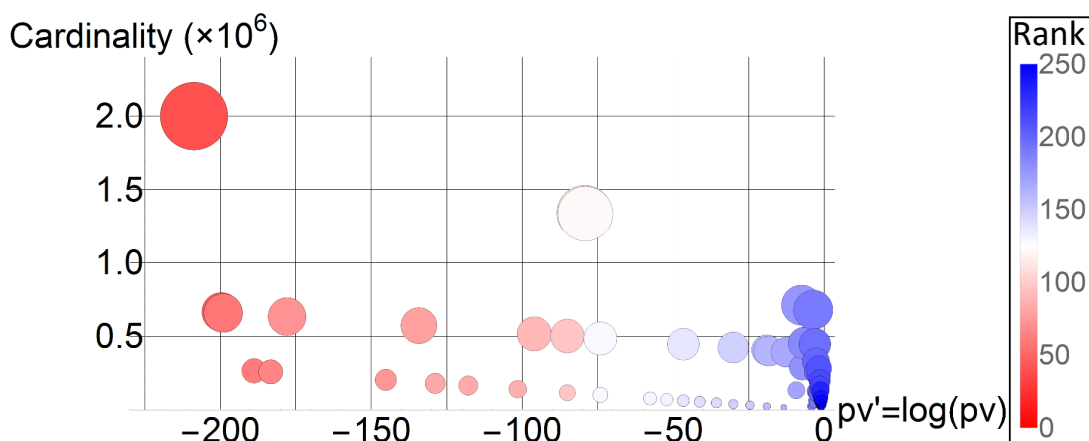
Figure 4.4: The p-values and the ranks of the composite regions of the phantom data from Figure 4.2. Each composite region out of 196 is shown with a disk. The area of a disk reflects the cardinality of the corresponding region. The p-values non-uniformly cover the range of $[0,1]$. The ranks are color-coded. They correspond to the p-values and can be used to compare the regions with respect to the dissimilarity significance. All the ranks are equalized by the sum of the regions' cardinalities (area of the disks) instead of the number of the regions.

Z-score method [115] and some other approaches. Edgington [120] suggests the following combination formula: $pv = pv_1 + \ldots + pv_k$. Fisher's combined probability test [121] computes the statistic $X = -2\sum_{i=1}^{k} \log pv_i$ that follows a Chi-Squared distribution with $2k$ degrees of freedom. Friston *et al.* [122] propose to calculate the p-value as $pv = (\max(pv_1, \ldots, pv_k))^k$. Tippett [123] suggests $pv = \min(pv_1, \ldots, pv_k)$ for the p-values combination. A retrospective discussion by Cousins [124] shows that there is no definite opinion on which test is the best. Different authors recommend Stouffer's Z-score method, Fisher's combined probability test and Tippett's approach. We choose Stouffer's Z-score method.

4.2.8 Global Ranking by Dissimilarity Significance

In our visualizations we encode the p-values of regions with certain visual attributes. Since the p-values generally span a large range of orders of magnitudes, we cannot directly map them to these attributes. Instead, we rank the regions by their p-values and map the ranks to the visual attributes. Our approach assigns ranks to all the regions, taking into account their cardinalities and preserving their relative differences in p-values. Using the ranks, the user can compare regions with respect to their p-values (Figure 4.4).

First, we sort all regions by their p-values in ascending order. Next, we put the regions into N_R bins $\mathcal{B}_1, \ldots, \mathcal{B}_{N_R}$. Since the regions vary greatly in cardinality, we equalize the bins by the sum of the cardinalities of the regions, instead of the number of the regions. During the binning process, we keep the order, defined by the p-values. Each bin has its own range of p-values that

are determined by its initially assigned regions. However, the p-value ranges can overlap at the end-points. In this case, we resolve ambiguities in bin assignments as follows. If a region has a p-value falling into the ranges of several bins $\mathcal{B}_{k_1}, \ldots, \mathcal{B}_{k_s}$ ($k_1 < \ldots < k_s$), we put this region into the bin $\mathcal{B}_{k_1}$. The *rank* of a region is the ordinal number i of the bin $\mathcal{B}_i$ it belongs to. Regions with lower ranks have smaller p-values and are more dissimilar to the markers than regions with higher ranks. Depending on the task, the user explores either low ranks (objective O1) or high ranks (objective O2).

Since the composite regions cover several atomic regions, they may have different range of p-values compared to the atomic regions. As we do not compare the atomic and composite regions, they are ranked separately. While the user specifies the markers, we display the ranks of the atomic regions. By looking at the ranks of regions that contain data values similar to the reference ones, the user can quickly validate the employed markers. If these ranks are rather low, then the reference data-value distribution is not adequately approximated by distributions from the markers. The user can refine the employed markers by adjusting their sizes and locations as well as adding new markers. During the data exploration we display the ranks of the composite regions, aiding the user in efficient localization of the regions with a desired dissimilarity significance. At this point we have established all our abstract concepts. Next, we present the associated visualizations.

4.2.9 Dissimilarity Significance Visualization

Having collected all necessary statistical information, we now visually encode it to facilitate the data exploration. The composite regions form the hierarchy that represents the data with different levels of detail. We visualize the hierarchy with a *radial tree*, conveying p-values of the regions via color (Figures 4.5c, 4.6). We define the level of detail in the hierarchy as the ordinal number of the corresponding level of the radial tree. The levels of detail start from one at the root of the tree and increase. With increasing levels of detail, this visualization occupies successively more screen space. The angular size of a region in the tree is determined by its cardinality. In order to aid the user in reading statistical information, we order the displayed regions with increasing dissimilarity significance in a counterclockwise fashion. Because of this layout, users easily spot relevant regions.

As all the data is represented with different levels of detail, at each such level i there is a region with the most significant dissimilarity $\mathcal{C}^{\max,i}$ and a region with the least significant dissimilarity $\mathcal{C}^{\min,i}$. Such regions are of particular interest to the user, as they fit the selected objective (either O1 or O2) the most. In order to convey the relationships between these regions throughout different levels of detail, we display a path of exploration in the radial tree (Figure 4.5c2). In case of the objective O1, this path connects regions $\mathcal{C}^{\max,i}$ at each level of detail i. For the objective O2, we connect with the path regions $\mathcal{C}^{\min,i}$. This path connects a composite region to its subregion with the desired dissimilarity significance, *i.e.*, $\mathcal{C}^{\max,i+1} \subset \mathcal{C}^{\max,i}$, $\mathcal{C}^{\min,i+1} \subset \mathcal{C}^{\min,i}$. This way, the user can investigate the data, by adjusting the level of detail interactively. To simplify the exploration of the finer (higher) levels of details, the j-th level parent of the currently selected region becomes a new root of the displayed radial tree. We set the value of j to half of the number of visible levels in the radial tree. Sometimes, the automatically suggested path would lead the user to a
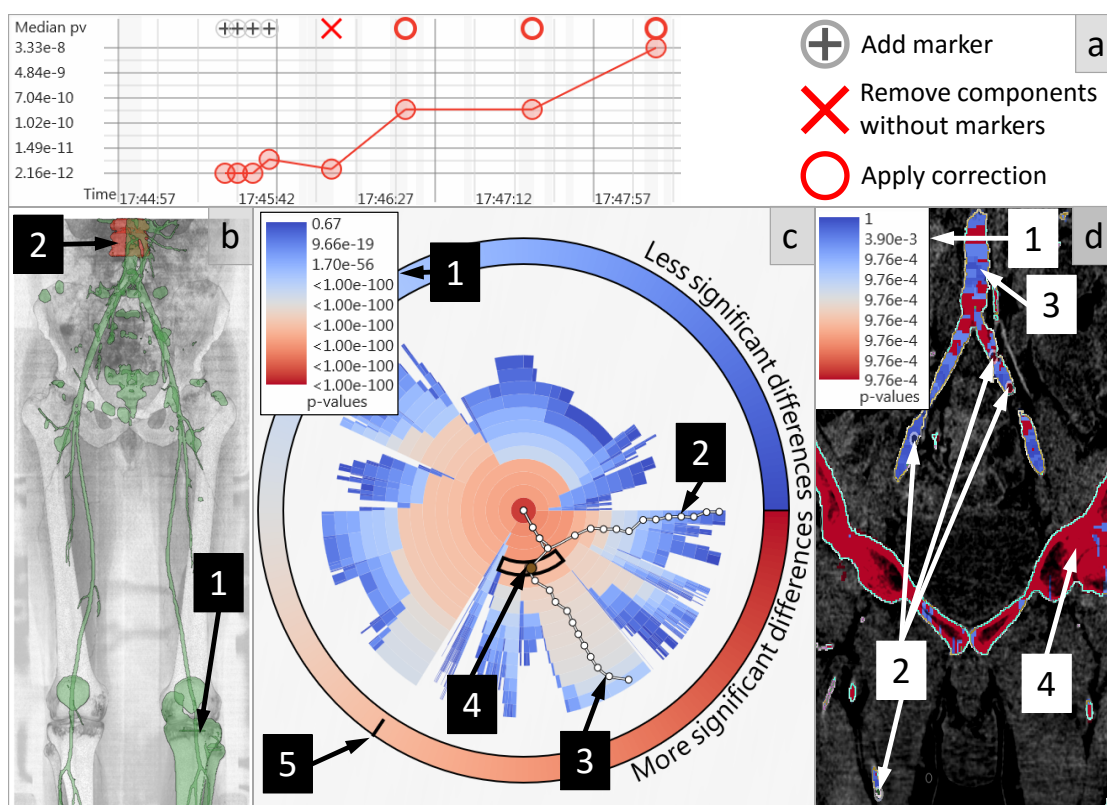
Figure 4.5: The visualization components of our proposed approach: a) the (median) p-values timeline plot, b) the data view, c) the radial tree view, d) a slice view for marker specification. The timeline plot (a) shows the median p-values after the user specified four markers and applied four operations. The data view (b) shows the entire data (b1, green) and the region of data, which is currently selected in the radial tree view (b2, red). Each composite region is displayed as a node of the radial tree (c). The p-values in the tree are color-coded according to the legend (c1). The user explores the data in the radial tree view by following the paths (c2, c3). The automatically-generated path (c2) links the composite regions with the most significant dissimilarities at each level of detail in the tree. Here, the user chose to explore an alternative path (c3) by selecting composite region (c4). The rank of the selected region is communicated at the annulus (c5). In this application scenario the markers are specified via the slice view (d). The p-values in this view are color-coded according to the legend (d1) The user specified four markers in a vascular structure (d2). The vessels are shown in blue tones (relatively low dissimilarity significance, d3), and the bones are highlighted with red tones (relatively high dissimilarity significance, d4).

region that is sub-optimal in terms of the chosen objective. We denote the level of detail that contains this region as k. In order to proceed, the user employs the radial tree. At the level k the user checks regions, alternative to $\mathcal{C}^{\min,k}$ and $\mathcal{C}^{\max,k}$, and alters the path. In the following sections we explain in detail the data exploration with the radial tree and the paths.

Figure 4.6: a) The radial tree and some of its nodes. b) The corresponding composite regions (shown in red) of the data from Figure 4.2. The dissimilarity against the marker $\mathcal{M}_1$ is evaluated. Its significance is color-coded in (a): blue tones depict large p-values (low significance), red tones highlight small p-values (high significance), and white tones are in the middle.

We employ the diverging color map *blue-white-red* by Kenneth [125] for displaying derived statistical information of the regions, as shown in Figures 4.5c1,d1. The higher ranks are displayed with blue color t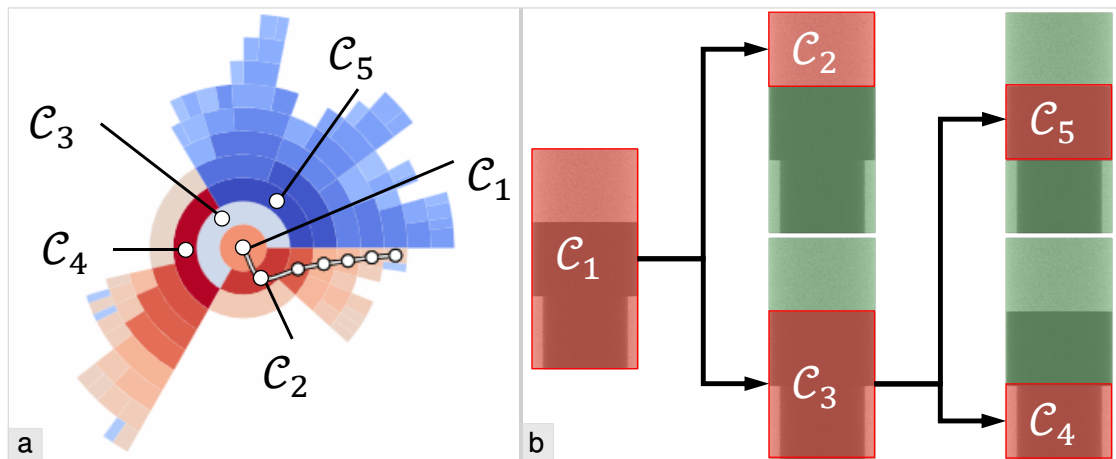ones. This tells the user that these ranks do not carry significant dissimilarities to the markers. White tones indicate an uncertainty related to the middle ranks: their p-values are not small enough to judge if these ranks have a significant dissimilarity. The lower ranks get the user's attention, being highlighted with red tones. The corresponding regions exhibit significant dissimilarities to the markers. The legend is displayed as an annulus around the radial tree, conveying both the diverging color map and the direction of increasing dissimilarity significance. As the data may consist of multiple connected components, at the coarsest level of detail the user can select the preferred component for further exploration. An additional annulus around the tree is divided into sectors that correspond to the connected components, which are given in the data. This annulus is shown only if there are two or more connected components, thus, it is not displayed in Figure 4.5. We link the data view (Figure 4.5b) and the radial tree view. This way, they complement each other. The data view displays the data and its regions in two-dimensional screen space. The radial tree view conveys statistical characteristics of the regions and their hierarchy.

The specification of markers requires a certain precision of the user input and involves the selection of samples from the data. Minor variations in the marker's position are allowed as long as it only contains reference data values. If markers belong to two or more features of the data, all such features would be matched. For data with one or two dimensions only the markers can be specified directly in the corresponding data views. In case of three or more dimensions the occlusion problem hinders the exact specification of the region and, thus, we recommend to show such data using the two-dimensional slice view (Figure 4.5d). We encode the ranks of the atomic

regions with the same *blue-white-red* color scheme, as used in the radial tree. This way we convey the dissimilarity significance of the atomic regions with respect to the currently specified markers. While the user specifies the markers, we compute the p-values and ranks interactively to facilitate an immediate check whether the current set of markers adequately approximates the reference data-value distribution.

We support dynamical situations, when the data, the markers, the atomic, or the composite regions change. Such changes can occur even if the analyzed data is spatial. We give as an example an interactive segmentation-editing task, where the segmentation mask is a part of the data. We record in time the consistency that indicates an agreement of data-value distributions between the markers and the rest of the data. As a measure of the consistency, we use the median p-value across all regions. The consistency is communicated to the user via the *p-values timeline plot* (Figure 4.5a). In this plot we use a logarithmic scale for the vertical axis. Each time a change occurs, we add to the plot a new time step. In order to better depict the trend of the p-values, we connect the median p-values from the individual time steps with lines. We do not use the largest and the smallest p-values, because in real-world data there can always be outlier regions with significant and non-significant dissimilarities to the markers. We do not average the p-values, as they may differ by orders of magnitude. Although the user might be interested in the p-value distribution, the median p-value is sufficient to judge consistency changes in our use-cases.

The p-values timeline plot is useful in three cases. If too few markers are specified, approximation of the reference distribution by their EDFs is inadequate. This situation is indicated by low median p-values. The user should add more markers to improve the low consistency in this case. After each change the user gets a hint whether the consistency actually improves. Such a hint may be useful in several applications, including tuning of filtering as well as reconstruction algorithm parameters, detection of acquisition artifacts, and abnormalities in the data, segmentation editing. The latter application we will investigate in Section 4.4.2. Having stored the p-value range of each rank for each time step, we can re-rank all current regions according to the p-value ranges of the selected time step. This enables a rank-based comparison that is often required in the data exploration tasks.

4.2.10 Data Exploration using Dissimilarity Significance

We propose the following protocol for data exploration, aided by the dissimilarity significance (Figure 4.7). The user starts the exploration in the radial tree view at the coarsest level of detail. Depending on the task, the user chooses the composite region with the most dissimilarity significance (objective O1) or with the least dissimilarity significance (objective O2). Then, the user follows the automatically suggested path in the radial tree, reaching the finer levels of detail. With the linked data view, the user finds the necessary level of detail along the path.

If the current path (Figure 4.5c2) leads to a result, that is sub-optimal in terms of the chosen objective, a new path should be selected. To do so, the user finds along the current path a coarse region, which is optimal in terms of the chosen objective, and selects it by clicking. Then, the radial tree view is locked on the selected region. In the radial tree view, the user explores a few subregions of the selected region and selects one that fits the chosen objective the best. The new
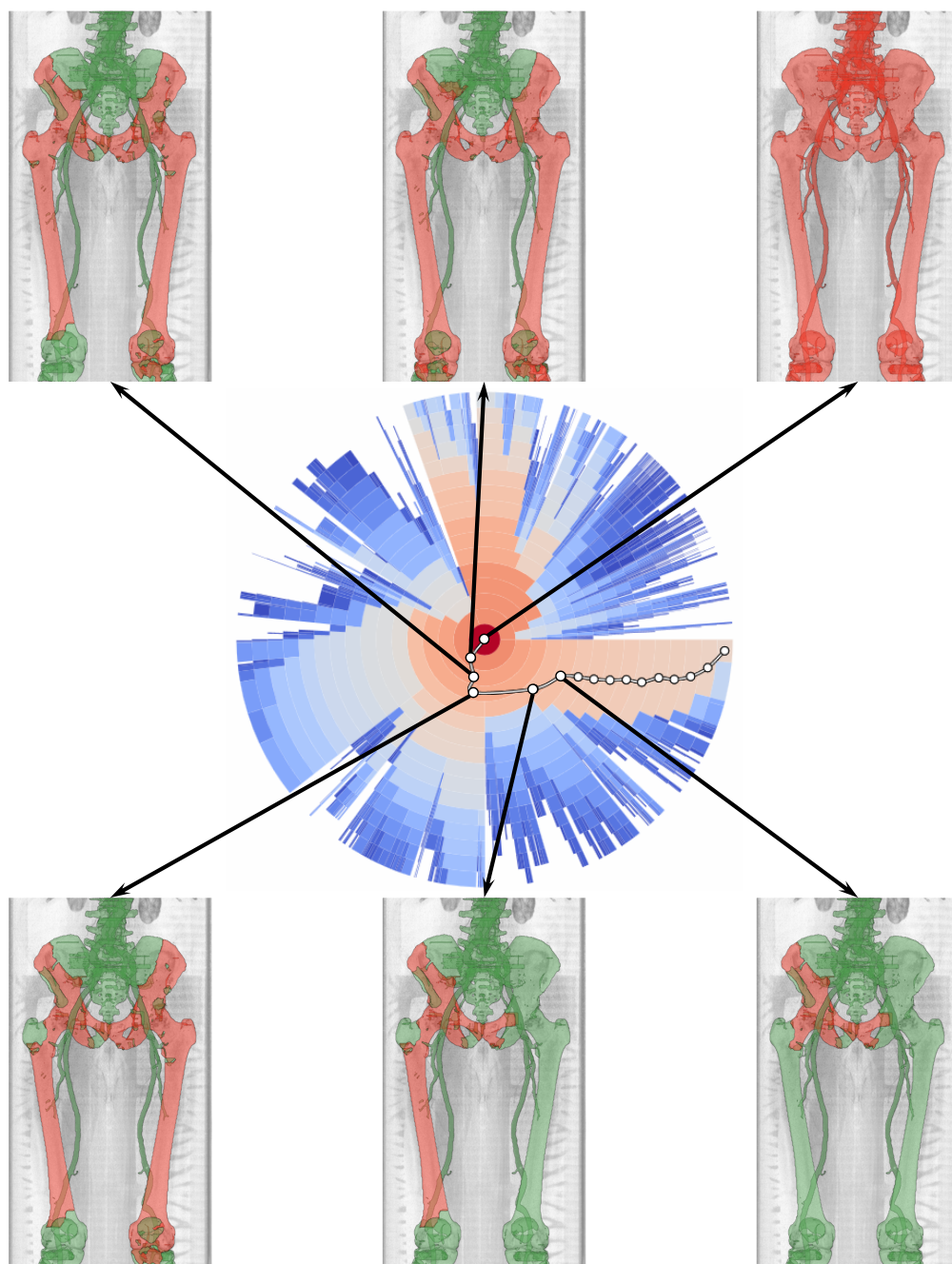
Figure 4.7: The data exploration protocol, using dissimilarity significance information. The radial tree view shows the data with varying levels of detail. At each level of detail, the region with the most significant dissimilarities is marked with a circle. Such regions across different levels of detail are linked with a path. The first six levels of detail are explored, and the corresponding regions are highlighted in red in the linked data view. The rest of the data is shown in green.

path, which goes through the selected subregion (Figure 4.5c4), is automatically generated and displayed (Figure 4.5c3). Finally, the user continues the data exploration, along the new path.

During the exploration, the user gets immediate visual feedback in both the data view and the radial tree view. This way, the user arrives at the desired region of the data, which has a proper level of detail and fits the chosen objective the best, without inspecting a possibly vast number of composite regions.

4.3 Implementation

Our implementation uses C# with Intel TBB (C++, on the CPU) and DirectX 11 Compute Shaders (HLSL, on the GPU) for parallelization. We implemented the two-sample randomized permutation KS test on the GPU. The number of permutations has to be in the interval $[1024, 1024 \cdot 16]$. The KS test reports p-values with single-precision floating-point numbers. We run Stouffer's Z-score method on the CPU in parallel threads. The computation of the normal distribution function is based on works of Cody [126] and Wichura [127]. In order to improve the numerical stability of our statistical framework against overflows and underflows, we compare $pv' = \log(pv)$ rather than the p-values pv themselves. The logarithm preserves the p-value comparison logic.

We store the samples from the continuous distributions in sorted lists. As for the discrete distributions, the data is organized into histograms, where each bin counts only a single discrete data value.

The value of the parameter $S_0 = 15$ in Equation 4.15 was determined in the application scenario with small objects (vessels in the human lower extremities), which will be described in Section 4.4.2. Smaller values of S_0 allow statistical tests on fewer data samples. However, if the data samples are too small, a statistical approach is not appropriate. With $S_1 = 0.5$ in Equation 4.16, the related criterion is only satisfied in regions that have large overlaps with markers. As data values in the overlaps are excluded from the statistical tests, the remaining parts of such regions are insufficient to perform the statistical tests. However, the users already explored these regions during marker specification, so we can safely exclude them from the further analysis. We use $N_R = 256$ as the number of ranks for implementation convenience, mapping the ranks to the *blue-white-red* color scheme.

4.4 Results

Our statistical framework supports a general scenario of comparing regions of the data with the user-specified markers. In this section we provide the following components: the basic test for a pair of an atomic region and a marker, the mechanism for combining p-values from multiple markers, and the combination method for composite regions. For each concrete application, one first adapts our framework to the application domain. The adaptation requires a realization of the following abstract concepts: the atomic regions $\mathcal{R}_1, \ldots, \mathcal{R}_k$, the composite regions $\mathcal{C}_1, \ldots, \mathcal{C}_h$, the markers $\mathcal{M}_1, \ldots, \mathcal{M}_l$. Moreover, our null hypothesis H_0 should have a proper interpretation in the application domain.

Atomic regions. We suggest a general atomic-region realization that should be used in case of a lack of a domain-specific realization. All atomic regions have the same cardinality, do not overlap, and each of them represents an m-dimensional hyper-cube in $\mathcal{D}$. The cardinality is application-dependent. It may use natural time intervals of temporal data, *e.g.*, hours, days, months, *etc.* For spatial data it may reflect the expected feature size or units of the volume.

Markers. The user specifies the placements and sizes of the markers. To keep the markers easy to localize, we realize them as m-dimensional hyper-cubes or m-spheres. The markers are usually specified in the analyzed data. However, they can also come from preselected reference data values in external data sources.

Composite regions. The composite regions can be realized according to the domain-specific logic. If such a realization is missing, we suggest the following general realization. First, we compare the p-values of the atomic regions to the user-specified significance level α. This will effectively remove many atomic regions that are not important for the users. Depending on the objective, we remove the regions with either non-significant (objective O1) or significant (objective O2) dissimilarities. From the remaining atomic regions we construct the composite regions by means of a watershed transformation. For the objective O1 we use the p-value of each atomic region as the height, whereas for the objective O2 we take one minus the p-value. The watershed transformation creates basins and watersheds. The watersheds are the ridges of the height field. They correspond to atomic regions that maximally disagree with the chosen objective compared to their neighborhood excluding the ridges. Each basin contains a local minimum of the height field. Each such minimum corresponds to an atomic region that fits the chosen objective best compared to its neighborhood. The remaining basin elements are separated from other basins by the watersheds. These elements correspond to atomic regions, which fit the chosen objective better than the watersheds. Finally, each basin represents a composite region of the data that fits the chosen objective better than its surrounding. We preserve connectedness of the data between the basins by adding the watersheds to each basin that they separate.

The number of composite regions may be too large for the user to inspect directly. We recommend constructing a hierarchy of the composite regions, if it is not accomplished by the composite region realization. We continue with the hierarchical watershed approach by Hahn *et al.* [27]. For combining the basins, we use the original metric, suggested by the authors. It is equal to the watershed height. Using the metric, we choose two adjacent basins with the lowest watershed between them among all available basin pairs. With the objective O1, we select two basins with the most significant dissimilarities. Under the objective O2, two basins with the least significant dissimilarities are found. We merge these two basins together, creating a new basin (a new composite region). The merging is repeated iteratively until there are no more adjacent basins. The constructed hierarchy represents the data with adjustable levels of detail. Alternatively, one may cluster the composite regions. However, clustering algorithms often require additional input in order to produce sensible results. We favor the hierarchical watershed transformation, as it does not introduce any additional parameters and respects the connectedness of the data. We demonstrate the generality of our proposed method by applying it to two different types of data. In the following section we show a temporal-data analysis scenario. After that, our method assists the user during segmentation editing on three-dimensional spatial data from the medical domain.

4.4.1 Time Series Analysis

In the following we analyze a single time series. We use the general realizations of our abstract concepts in order to investigate it. As a concrete example, we take the maximal daily temperature in Melbourne, Australia in the period 1981–1990 (Figure 4.8a). The task is to find similar weather conditions (objective O2). The atomic regions span entire months. The user specifies a single marker from 1981-Jun-03 till 1981-Jun-25 (Figure 4.8b). The hierarchy of composite regions is then constructed and displayed in the radial tree view. The user gets an overview of all found time intervals with possibly similar temperature distributions. In this case, ten intervals are found (Figure 4.8c): July-August 1981, June-August 1989, June-July 1986, June-July 1982, June-August 1983, June-August 1984, June-August 1990, June-August 1987, June-September 1985, and June-July 1988. They are ordered by decreasing p-values. The user checks each interval (Figure 4.8f). The first seven time periods have the following p-values: $pv_1 = 0.388$, $pv_2 = 0.387$, $pv_3 = 0.276$, $pv_4 = 0.198$, $pv_5 = 0.146$, $pv_6 = 0.076$, $pv_7 = 0.057$. These p-values are higher than the set significance level ($\alpha = 0.05$). The user concludes that there is no significant difference so far in the weather conditions between these time intervals and June 1981. The last three time periods have the p-values below the significance level: $pv_8 = 0.025$, $pv_9 = 0.023$, $pv_{10} = 0.014$. The user decides, that the weather conditions exhibit significant differences during these time intervals, compared to June 1981. Time intervals that were discarded during the comparison with the significance level, *e.g.*, September 1983-May 1984, have weather conditions with significant differences compared to June 1981.

4.4.2 Segmentation Editing

Various applications may benefit from domain-specific realizations of atomic and composite regions. The main idea behind using the specialized realizations is to transfer additional information to the statistical analysis. As an example, we present an adaptation of our method for segmentation editing, where we include spatial features that are captured by skeletonization. Geometric operations on the input segmentation mask define the atomic and composite regions.

Let us assume that we have a three-dimensional data and a mask $\mathcal{L}$ of an object, which is defined as a set of voxels. The mask $\mathcal{L}$, generated (semi-)automatically, often contains regions which, in fact, do not belong to the object. Also, there are actual parts of the object that are not included in the mask $\mathcal{L}$. Both kinds of regions stem from data irregularities/abnormalities and their processing by the segmentation algorithms. The user searches for such regions and either removes or adds them to the mask.

Our statistical approach assists the user during the data exploration process. First, the user specifies the markers that definitely belong to the object. According to our null hypothesis H_0, we identify the first kind of regions as having significant dissimilarities to the markers (objective O1). The second kind of regions, though, exhibits non-significant dissimilarities to the markers (objective O2). Having the dissimilarity-significance information, we suggest the user regions that are relevant for subsequent editing.
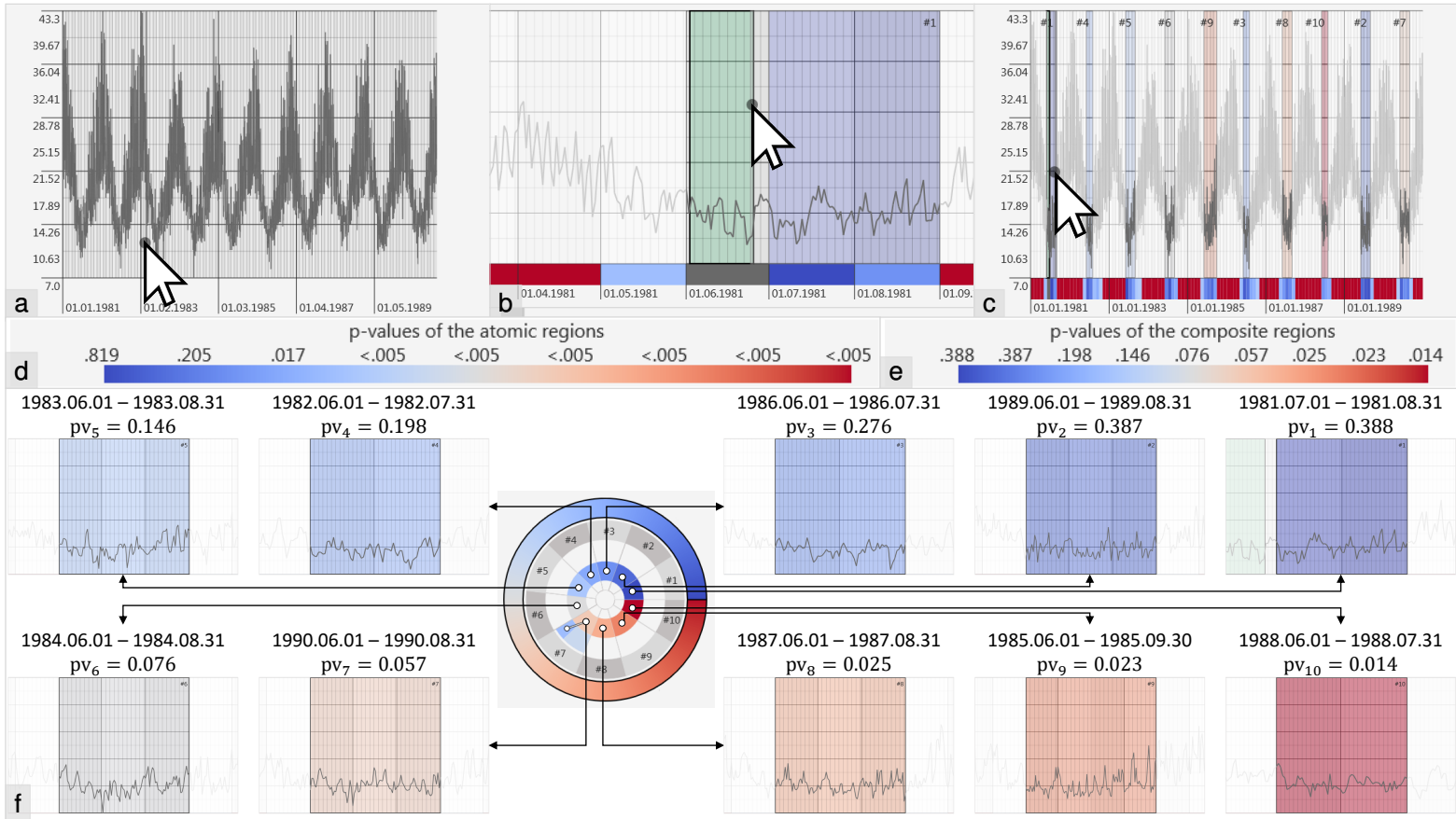
Figure 4.8: A time-series analysis using our dissimilarity-significance method. The task is to find similar (objective O2) weather conditions, considering maximal daily temperature in Melbourne, Australia: a) the measurements throughout the entire observation period (years 1981-1990), b) the user specifies the marker (green) being interested in the temperature distribution from 1981-Jun-03 till 1981-Jun-25, c) the user gets an overview of all found time intervals with possibly similar weather conditions (highlighted with red-white-blue colors), d,e) the p-values legend for the atomic and the composite regions, f) the user investigates ten found time intervals in detail via the radial tree. The first seven time intervals have p-values higher than the set significance level ($\alpha = 0.05$) and indicating no significant difference so far to the weather in June, 1981. The last three time periods have p-values that are lower than the level α and show significant differences in the weather conditions, compared to June, 1981.

The spatial features, captured by the skeletonization, are vital for this scenario, as demonstrated in the previous chapter. Therefore, we provide the specialized realizations of the atomic and composite regions.

Atomic regions. The concept of the influence zones, introduced in Chapter 2, perfectly matches the atomic regions. The influence zones are constructed as follows. We apply the thinning-based skeletonization by Lee *et al.* [44] on the mask $\mathcal{L}$. During the skeletonization iterations we assign to each peeled voxel the current iteration number. With minor arithmetic operations, these numbers are converted into skeleton distances. We link each voxel in $\mathcal{L}$ to the closest skeleton voxel, building the influence zones. The influence zones cover the mask $\mathcal{L}$ entirely and do not overlap with each other. Each zone can be thought of as a section of the object "orthogonal" to the skeleton. We assume that each influence zone belongs to a single spatial feature of the object if the segmentation mask is correct. Influence zones at segmentation defects, however, may belong to two or more spatial features and exhibit mixed data-value distributions. Such mixed distributions have significant dissimilarities compared to markers, indicating the underlying defects. If the object consists of several tissues or materials, data-value distributions of influence zones should be partially matched with the markers that represent individual tissues or materials. Currently, the partial match is only possible by placing additional markers at interfaces between different tissues or materials. The partial match with reference data-value distributions, generated from groups of weighted markers, is a promising future work. We realize the atomic regions $\mathcal{R}_1,\ldots,\mathcal{R}_k$ as the influence zones, constructed by Algorithms 2.1 and 2.2.

Composite regions. This time we adopt the concepts of the correction regions and the correction operations, introduced in Chapter 3. The entire mask $\mathcal{L}$ is split into the correction regions according to the histogram dissimilarity metric. The metric analyzes differences between the influence zones and inside individual influence zones in order to find discrepancies, specific to the aforementioned two kinds of regions. To the user this technique suggests a vast variety of correction operations with the corresponding correction regions. The user manually inspects them and selects the ones which actually have to be removed or added to the mask $\mathcal{L}$. The correction regions consist of the influence zones and form a hierarchical representation of the entire mask $\mathcal{L}$. We realize the composite regions $\mathcal{C}_1,\ldots,\mathcal{C}_h$ as the correction regions.

Instead of manually inspecting all correction regions, the user follows the paths in the radial tree view. This way we guide the user to the correction regions that are relevant for the editing task. Each time the user edits the mask $\mathcal{L}$, the p-values timeline plot provides an indication whether the mask $\mathcal{L}$ is improving during the correction process. Non-significant p-values in this plot can be used as a termination criterion of the editing process.

As a concrete example, we perform the segmentation editing of Computed Tomography Angiography (CTA) datasets of the human lower extremities. The goal is to obtain the masks of the vessels, separated from the bones. The given medical datasets exhibit severe pathological conditions as well as major anatomical variability between the patients (missing vessel branches, different branching points). The fact that vascular structures are touching bones causes severe segmentation defects due to the administered contrast agent (Figure 4.9a). In Figure 4.9 we present an editing process of one of these datasets. The user starts with the segmentation that contains severe

defects, *e.g.*, bones and table fragments (Figure 4.9a). Four markers are specified inside of the vessels (Figure 4.9b). In the slice view the user notices, that the vessels are colored with blue tones (relatively low dissimilarity significance), and the bones are mostly shown in red tones (relatively high dissimilarity significance). The user removes the table by removing connected components without markers. The user finds the first three correction operations, following the automatically suggested paths in the radial tree view (Figure 4.9c-e). This removes major bone structures, but fragments of the vertebrae, the tibia, and the fibula are still present. Along the alternative path the user finds the fourth correction operation that removes the vertebrae fragments (Figure 4.9f). The remaining bone fragments are removed by the fifth correction operation, found at the automatically suggested path (Figure 4.9g). With this the user finishes the editing process, reaching a satisfying result (Figure 4.9h). For validation purposes, the resulting segmentation is checked against the initial one. The user selects the last step in the timeline plot, where the segmentation is not modified. As expected, all the regions get higher ranks, indicating their relatively low dissimilarity significance (Figure 4.9j). After editing, the vessel mask has a quality level comparable to the results of a technique used in the daily clinical routine (Figure 4.9i). All major vessels are preserved, while bones and table fragments have been correctly removed.

Domain Experts Feedback

In order to validate our statistical method with domain experts, we choose the segmentation-editing scenario in the medical image processing domain. Two domain experts evaluated our approach with respect to general usability, benefits of having statistical information available, and interaction aspects. They are radiologists, experienced in segmenting vascular structures from various locations of the human body. They are experienced in correcting segmentation masks, since the results of their currently employed methods do not meet the required quality standards. The resulting vessel mask is used for rendering of vessel reformation images for diagnostic purposes.

During the evaluation session, the domain experts corrected the vessel segmentation masks with our proposed technique T1. Our domain experts also performed the same tasks with the technique T2, described in Chapter 3, as well as with the technique T3 in the AngioVis framework [1] that they use in the clinical routine. The techniques T1 and T2 do not use any domain-specific knowledge, in contrast to the technique T3.

We used ten CTA datasets from the daily routine of our domain experts. First, we showed to the experts the correction of one dataset with our proposed technique T1. The author presented the marker specification and the data exploration protocol to provide a basic understanding. The experts were introduced to the workflow of T1 as well as to the method for a statistical assessment of suggested correction operations. Interaction examples were presented. Then, the experts tried our proposed technique T1 themselves on four other datasets in order to train, as they were completely unfamiliar with it. The same introduction to the technique T2 was made.

Each expert corrected five test datasets with techniques T1, T2, and T3. For the initial segmentation, we employed thresholding with a manually set constant threshold, followed by a few morphological operations and a connected components analysis. The technique T3 used
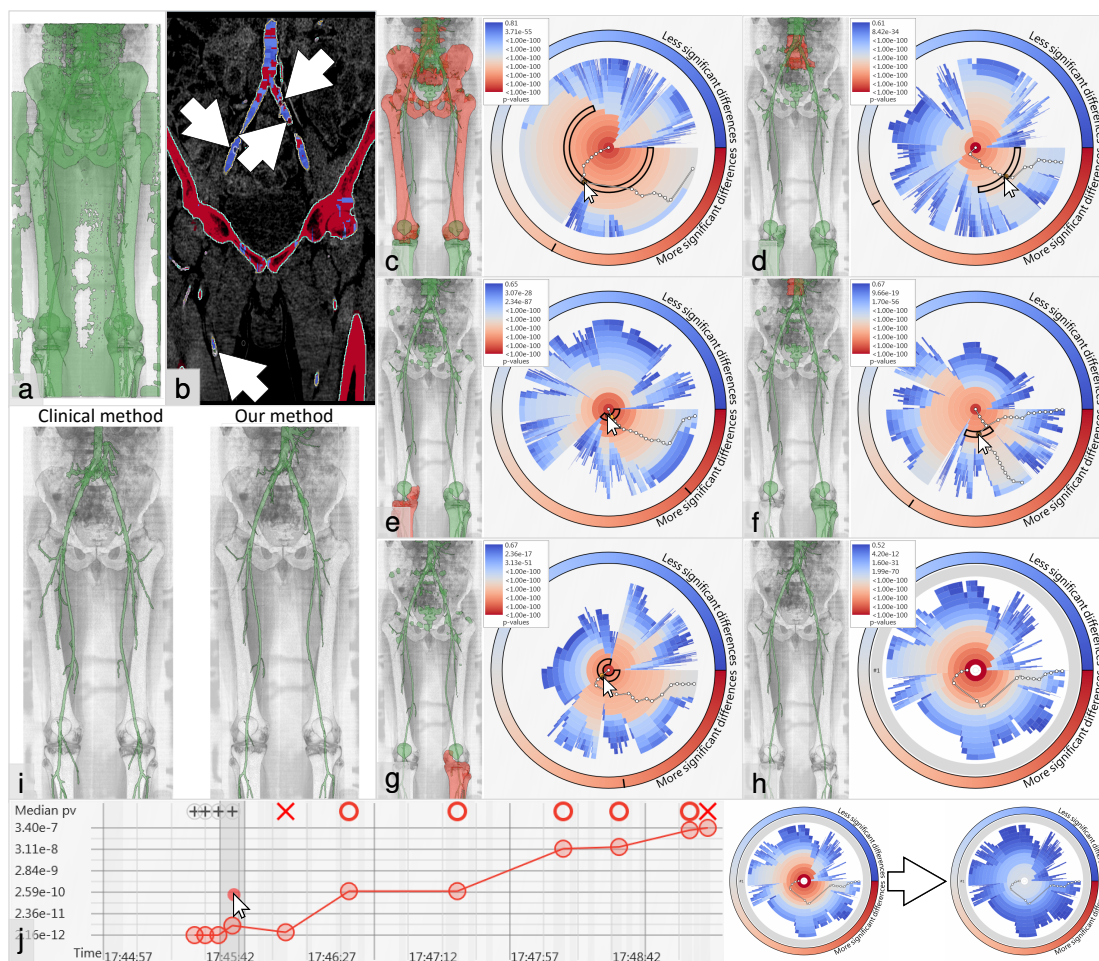
Figure 4.9: The segmentation editing scenario using our dissimilarity significance method. The task is to edit a segmentation mask of vascular structures in CTA data of human lower extremities. The composite regions represent the mask with different levels of detail. In order to correct the mask, the user selects regions for removal. The user is interested in the regions exhibiting significant differences (objective O1) to the object-of-interest (vessels): a) the automatic vessel segmentation (green) exhibiting defects (fragments of the scanning table and bones are included), b) the user specifies four markers inside of vessels (indicated with arrows), c-e) following the automatically suggested paths, the user finds the first three correction operations (removal of red parts), f) the user follows the alternative path and finds the fourth correction operation (removal of red part), g) the user finds the fifth correction operation, following the automatically suggested path (removal of red part), h) the user achieves a satisfying result after only five operations, i) comparison of our method's result with a result obtained by employing a technique from clinical routine: the results are of the comparable quality, few small vessels are not properly segmented as they are out of interest for the radiologists, j) the user validates the segmentation-quality improvement by comparing the result (h) to the original state (a) (selected in the timeline plot) – the composite regions are finally more in the blue color range and achieve higher ranks, indicating higher p-values than those of the original segmentation mask.

the masks generated by an advanced vessel segmentation technique that distinguishes bones and vessels via an intensity criterion. Thus, the quality of the initial segmentation was superior for the technique T3 than for the techniques T1 and T2. However, the technique T3 and its preceding segmentation method are parts of the well-established clinical workflow, so we did not compensate for this difference in our results. Each domain expert first corrected the test datasets with T1, then processed them with T2, and after three days processed them with T3. After the experts finished their tasks, they answered our questionnaire (see Appendix B). We measured the time that was required to correct the vessel masks. On average, with our technique T1 the experts spent five minutes fifty seconds on each dataset. The users never finished their tasks with the technique T2, as even one single dataset was not completely corrected after twenty minutes. With the base-line technique T3 the experts achieved the desired vessel segmentation in six minutes twenty seconds per dataset on average.

The domain experts assessed the usability of our method with the System Usability Score by Brooke [87]. The average score was 86 points out of 100, which approximately corresponds to the 90th percentile of the perceived **usability**.

The experts felt **confident** while performing their tasks, having the statistical information from our proposed method. They consider this information as **useful**. According to our domain experts, the radial tree view significantly improved the **overview** of suggested correction operations compared to the technique T2. Also, the experts liked the **exploration** of the suggested correction operations with the paths, which indicate the statistically most significant dissimilarities. The markers' **specification** via the slice view was rather easy and straightforward to our experts. With the markers the experts were able to **precisely** indicate the regions of interest, *i.e.*, the vascular structures. Our chosen **color-coding** scheme proved to be good for both the overview and exploration purposes, judging by responses of the experts and the achieved segmentation results. Finally, with the help of the timeline plot as an **overview**, the experts were able to see a progressive improvement of segmentation quality during the correction. Details on the scores are available in Figure 4.10a.

Even after a very short adaption period of ninety minutes, the experienced radiologists liked the radial tree view of our technique T1. It enabled the experts to efficiently navigate between different correction operations. The paths, displayed in the radial tree view, were beneficial for the experts, who used this feature at each editing step. These paths, being generated from the statistical information, indicate the usefulness of our proposed statistical dissimilarity evaluation method. The results of the technique T2 strengthen this fact further. Although the correction operation suggestions were completely the same for the techniques T1 and T2, the experts were unable to finish the editing with the technique T2 in a reasonable amount of time (twenty minutes per dataset) due to the data complexity. The main reason was a lack of overview of the correction-operation suggestions, as pointed out by the mediocre score in the evaluation in Chapter 3. In our technique T1, the radial tree view and the paths, carefully investigated by the radiologists, rectify this deficiency and reduce necessary interaction. We may conclude that our technique T1 enables the exploration of data with complex structures, such as vessels.

Compared to the base-line technique T3, our technique T1 required less **interaction**. The task completion **time** was slightly shorter as well. Both techniques were considered **easy** to use by
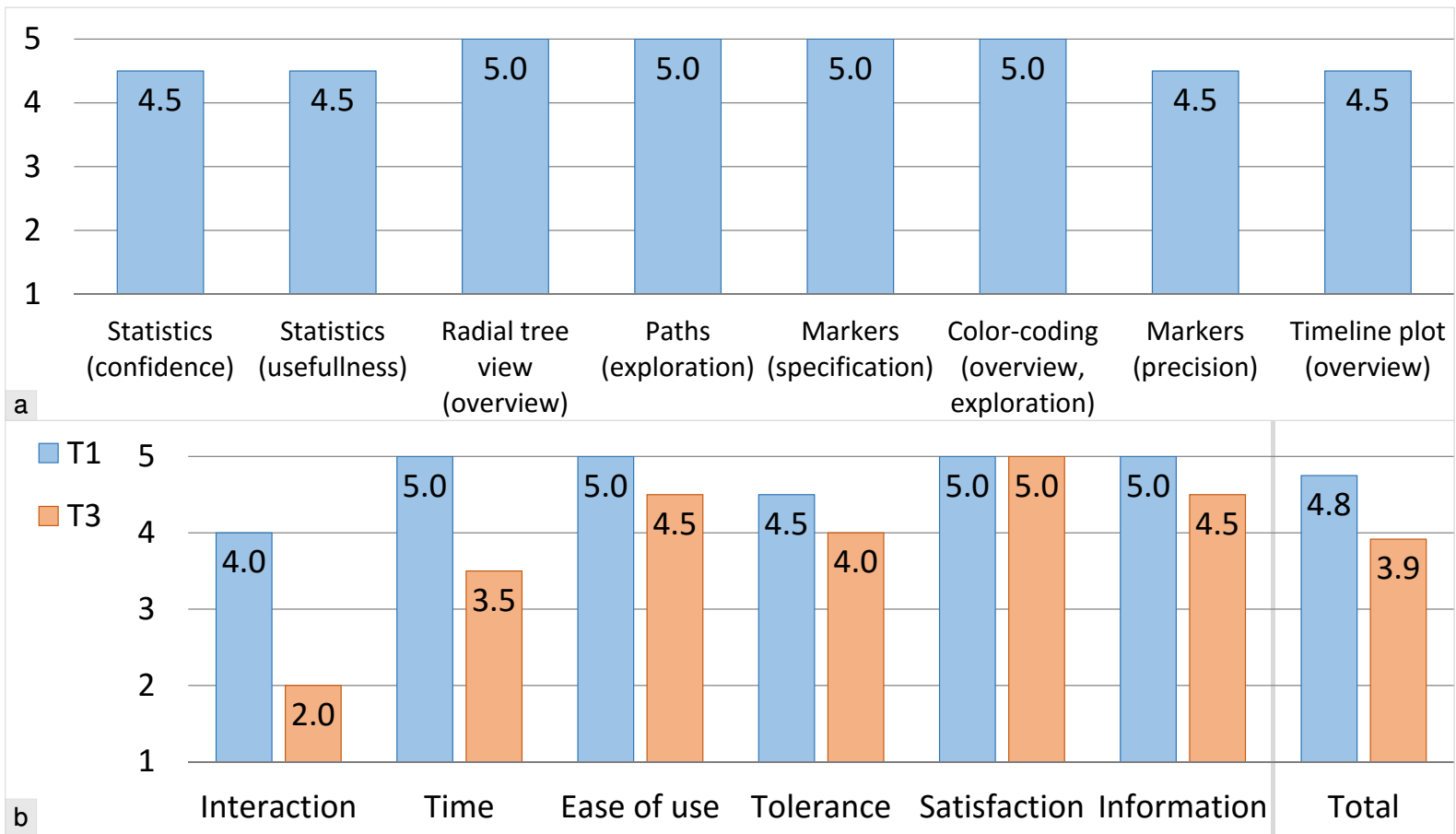
Figure 4.10: a) Evaluation of our technique. b) Comparison of our method (T1) to the clinical method (T3). Grades range from 1 (worst) to 5 (best).

the experts. The **tolerance** against imprecise input was rated slightly higher for our technique T1. Both techniques T1 and T3 allowed the experts to reach **completely satisfying** results. As for support of the users during the interactive editing with relevant **information**, our technique T1 was ranked slightly better than the base-line technique T3. Detailed information on a comparative evaluation can be found in Figure 4.10b. In **total**, our technique T1 got a better score than the base-line technique T3. This was achieved by extending the existing technique T2 with our dissimilarity significance computation and novel visualization components. Without these extensions, according to one of our domain experts, the technique T2 "was unable to process complex datasets such as peripheral arteries". Still, the technique T2 properly handles less complex objects, such as single organs or specimens.

Robustness Tests

In order to verify the robustness of our statistical method, we conducted several tests. For these tests we used three-dimensional spatial data and the specialized realizations based on the influence zones and correction regions. We chose it because such kind of data is usually harder to analyze than a time series due to the higher dimensionality.

Robustness of the Statistical Tests

In order to evaluate the **robustness** of the employed **statistical tests** (the KS test and Stouffer's Z-score method), we conducted checks with a phantom 3D object composed of two parts, each exhibiting different data value distributions (six cases). The object is depicted in Figure 4.11. A single marker was placed in one of two positions: the middle point of the upper part and the middle point of the lower part of the object. The marker radius was 16 voxels. The differences between the distributions of two object parts were varied with delta $\Delta d \in \{0, 1, 2, 4, 8, 16, 32, 64\}$. As a sanity check, we included the situation where there is no difference between the distributions (*i.e.*, $\Delta d = 0$). The original parameters were $\mu_1 = 30, \sigma_1 = 10$. The cases were:

- case *C1* - the lower part has the Gaussian distribution $\mathcal{N}(\mu_1, \sigma_1)$, and the upper part has the Gaussian distribution $\mathcal{N}(\mu_1 + \Delta d, \sigma_1)$

- case *C2* - the lower part has the Gaussian distribution $\mathcal{N}(\mu_1, \sigma_1)$, and the upper part has the Gaussian distribution $\mathcal{N}(\mu_1, \sigma_1 + \Delta d)$

- case *C3* - the lower part has the Gaussian distribution $\mathcal{N}(\mu_1, \sigma_1)$, and the upper part has the Poisson distribution $\mathcal{P}(\mu_1 + \Delta d)$

- case *C4* - the lower part has the Gaussian distribution $\mathcal{N}(\mu_1, \sigma_1)$, and the upper part has the distribution $A_1 \sim \mathcal{P}((\sigma_1 + \Delta d)^2) + \mu_1 - (\sigma_1 + \Delta d)^2$ with $E[A_1] = \mu_1$ and $Var(A_1) = (\sigma_1 + \Delta d)^2$

- case *C5* - the lower part has the Poisson distribution $\mathcal{P}(\mu_1)$, and the upper part has the Poisson distribution $\mathcal{P}(\mu_1 + \Delta d)$

- case *C6* - the lower part has the Poisson distribution $\mathcal{P}(\mu_1)$, and the upper part has the distribution $A_2 \sim \mathcal{P}(\mu_1 + 3\Delta d) - 3\Delta d$ with $E[A_2] = \mu_1$ and $Var(A_2) = \mu_1 + 3\Delta d$
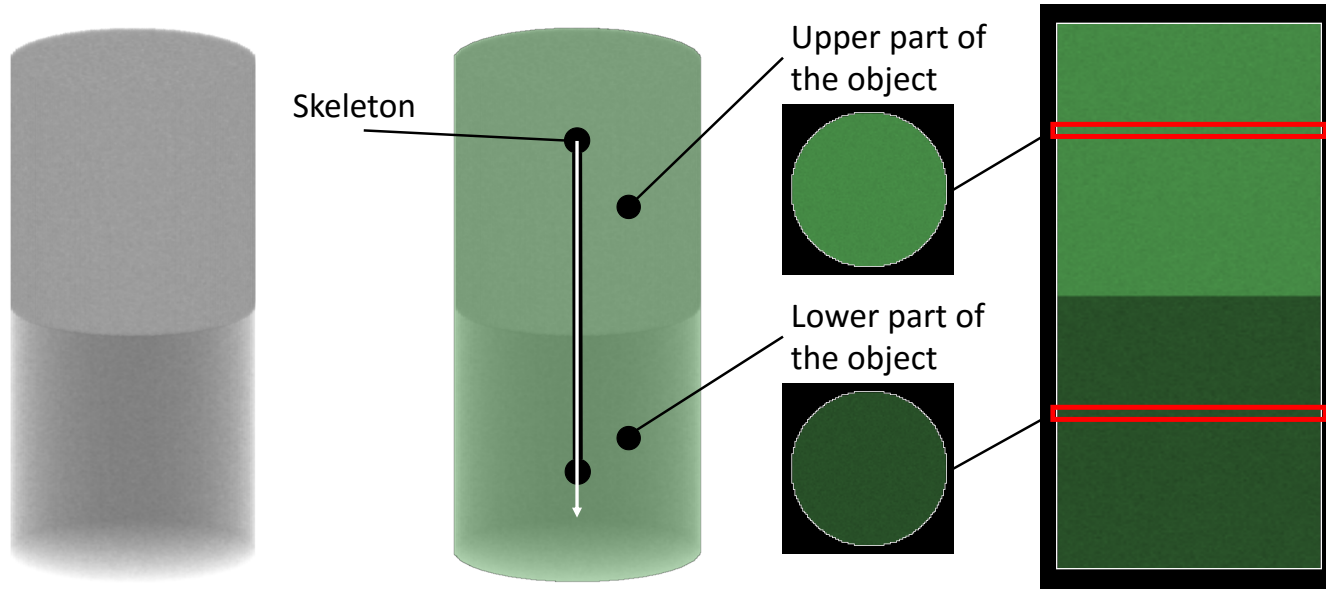
Figure 4.11: The phantom object, its two parts and its skeleton.

Following the common representation of three-dimensional spatial data, the generated sample values were rounded to integral numbers. The data values generated were numbers from -32768 to 32767 (signed 16-bit integers).

The detailed information on the p-values along the skeleton is shown in Figures 4.12-4.17. The significance level was $\alpha = 0.01$. We note that the differences are statistically significant starting from $\Delta d = 1$ in the case C1, $\Delta d = 2$ in the case C2, $\Delta d = 0$ in the case C3, $\Delta d = 1$ in the case C4, $\Delta d = 1$ in the case C5, and $\Delta d = 2$ in the case C6. We conclude that our statistical pipeline detects significant dissimilarities in all tested cases, even if the differences are rather small ($\Delta d \geq 2$). If $\Delta d \geq 1$, the p-values of the object part without a marker indicate significant dissimilarities ($pv \leq 0.01$), as illustrated in Figure 4.18. The p-values of the object part with a marker, however, convey no significant dissimilarities. In case of no differences ($\Delta d = 0$), the reported p-values show no significant dissimilarity as well (only applicable for cases C1, C2, C5, C6).

Robustness against Noise

To verify the **robustness** to **noise**, two kinds of artificial noise were added to the medical dataset depicted in Figure 4.7. The first kind of noise was normally distributed with a mean $\mu = 0$ and varying standard deviation σ according to Table 4.1. The second kind of noise had a Poisson distribution with a varying mean μ as shown in Table 4.1. The Peak Signal-to-Noise Ratio (PSNR) was used to evaluate the degree of artificial noise. We used 2048 as the maximal value (upper range for integer values used in CTA data). The segmentation mask was generated anew for each μ and σ. Our statistical method was aiding an interactive segmentation-editing framework. With a thresholding algorithm, masks of bones and vessels were generated. The author then edited them to keep only the vessels. For each dataset, only four markers were specified. In each corrected segmentation mask, the markers were placed at exactly the same positions. The markers' positions were determined in the original dataset without noise. In the test we used one medical CTA dataset of the human lower extremities. The dataset had 800 slices; each slice had resolution of 512×512 pixels.

At each tried noise level, the user achieved the desired result. As a quality measure, we employed the Jaccard coefficient J (Equation 2.1) between the output mask and the reference solution, obtained from the data without the artificial noise and manually refined by one of our domain experts. The Jaccard coefficient is a normalized similarity measure, which reaches one in case of a complete match between two tested sets. The reference solution required three editing steps as well as some voxel level editing, accomplished in 90 seconds. The input masks had a Jaccard coefficient lower than 0.05, indicating a large deviation from the reference solution. With all tried noise models, the editing was successful, as shown by a significant improvement of the quality measure. Moreover, our statistical method was discriminating bones from vessels, as demonstrated by the resulting segmentation masks of the vessels. Detailed information on editing time, number of steps and the output quality can be found in Table 4.1. For comparison purposes, the resulting masks are shown in Figure 4.19. The quality of the result, measured with the Jaccard coefficient between the output mask and the reference solution, was above 0.88.
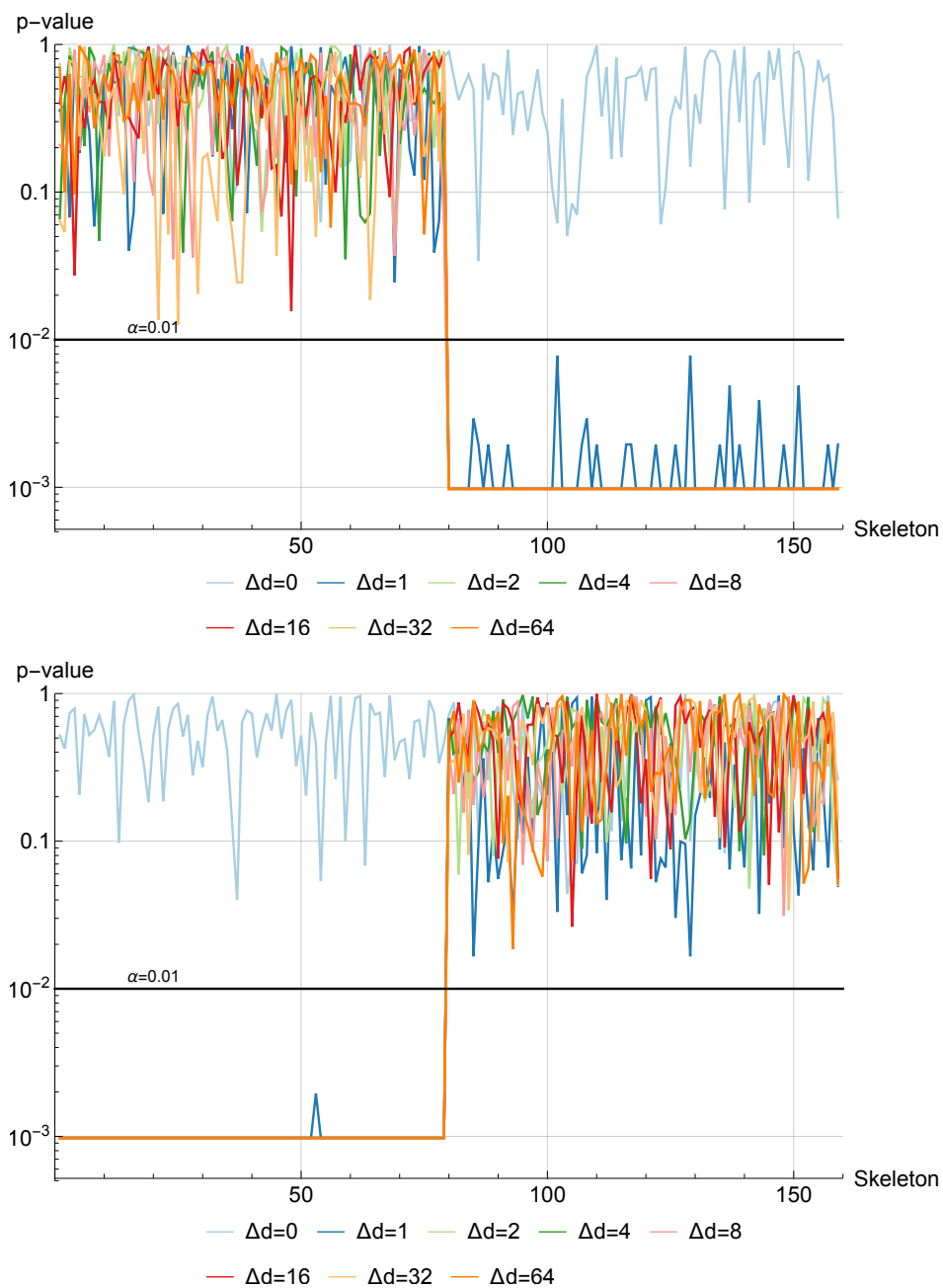
Figure 4.12: Case C1: p-values along the skeleton with respect to different Δd. The top image shows the p-values if the marker is placed in the upper object part. The bottom image reports the p-values if the marker is located in the lower object part.
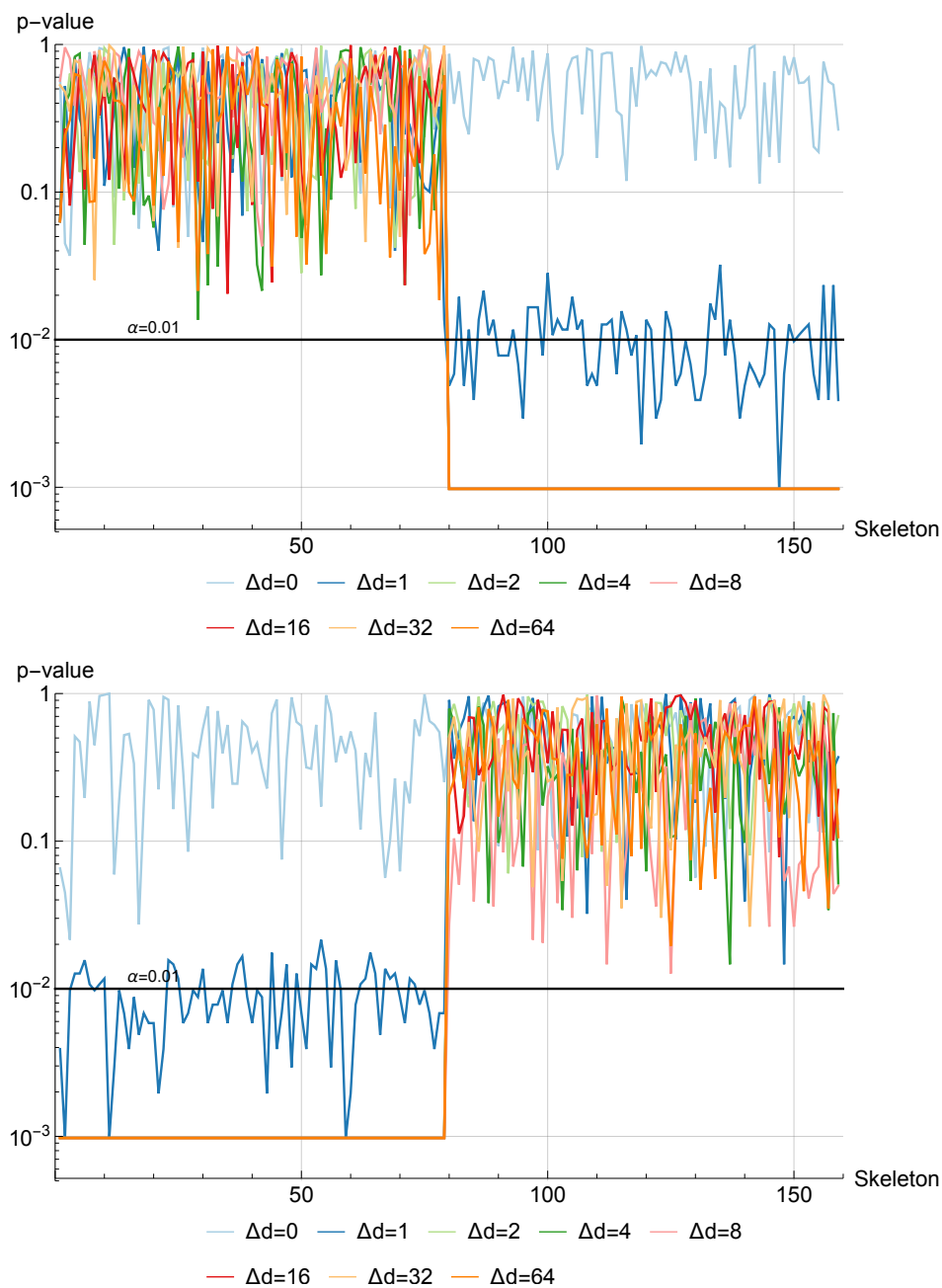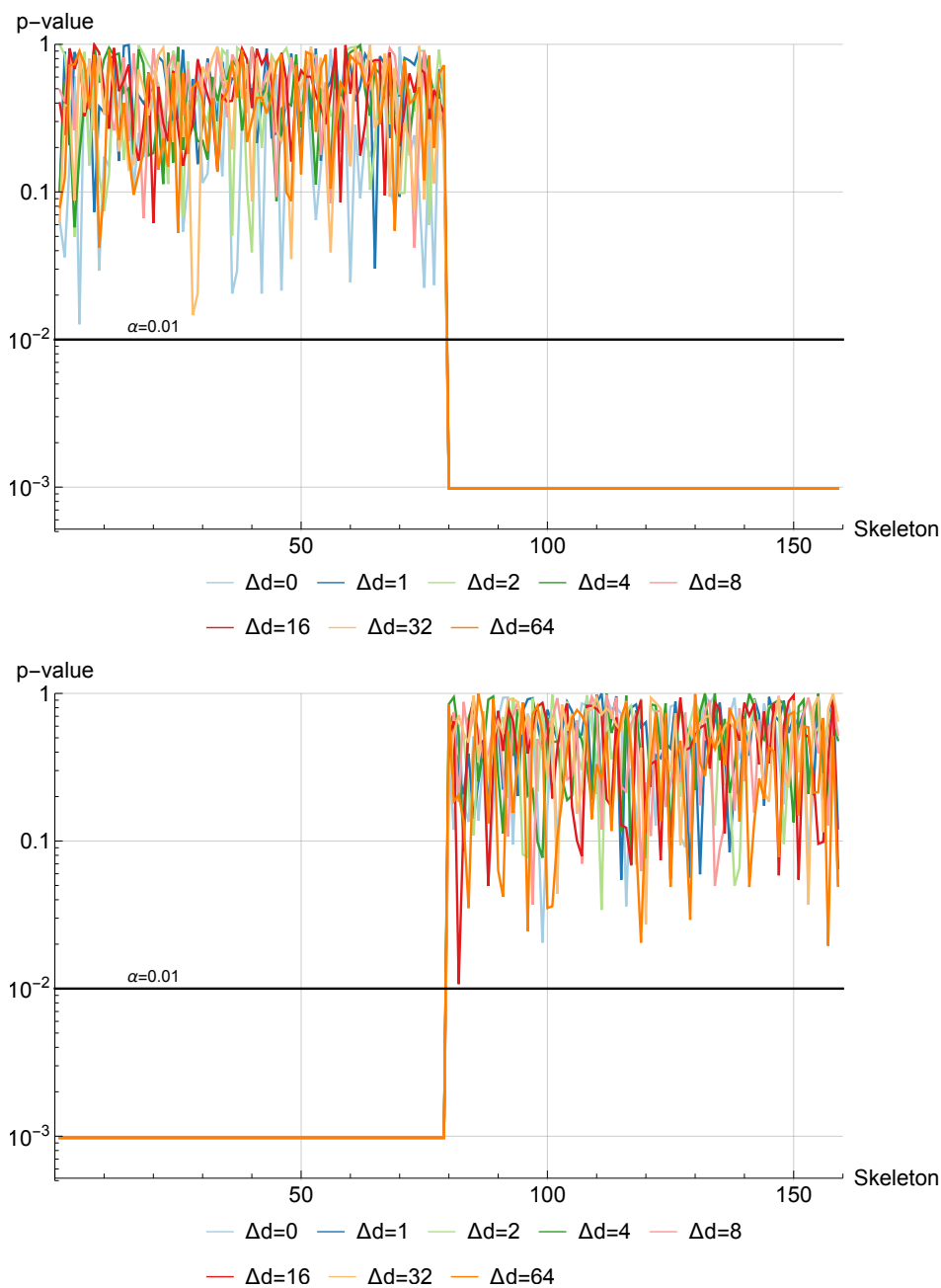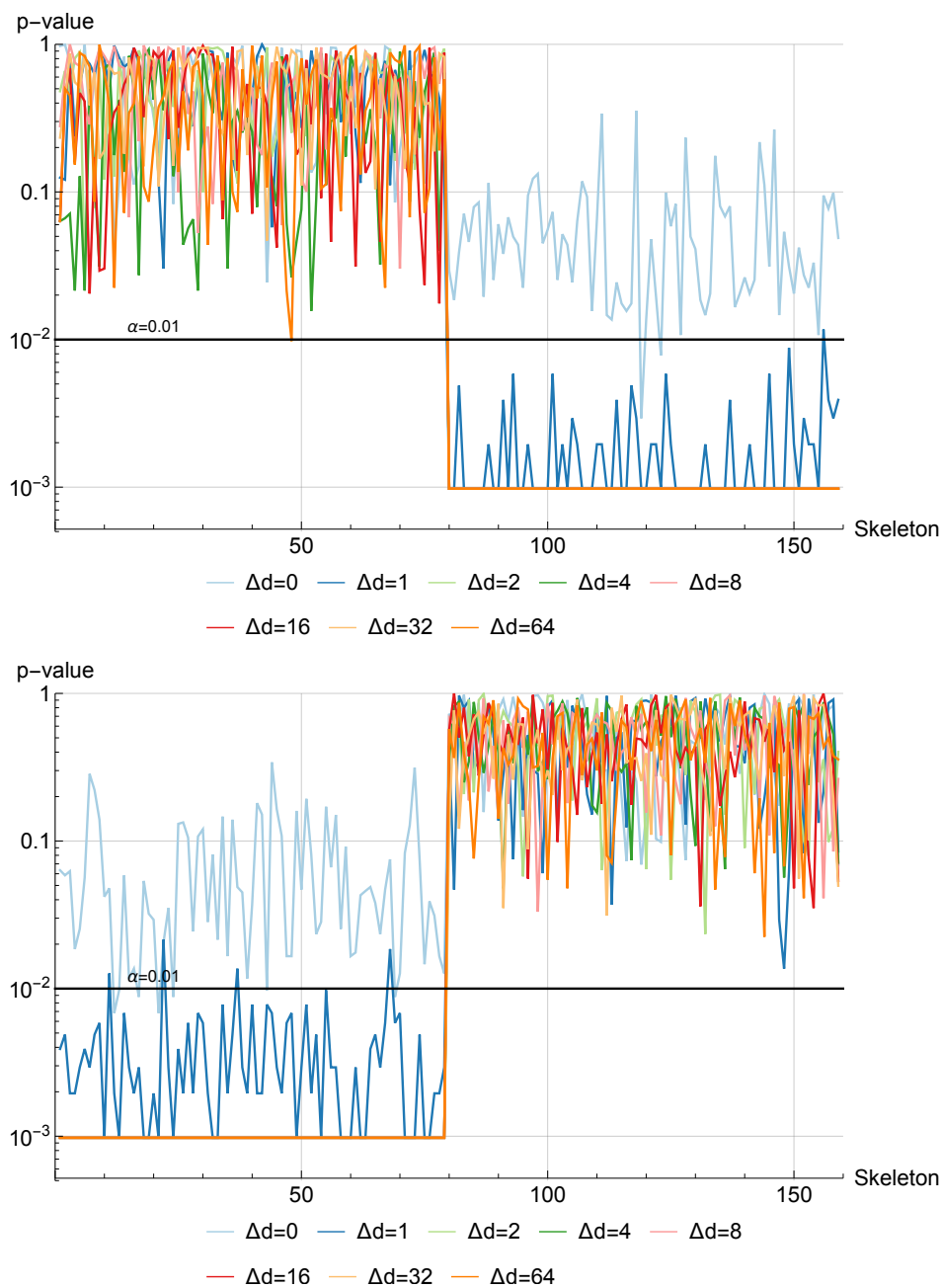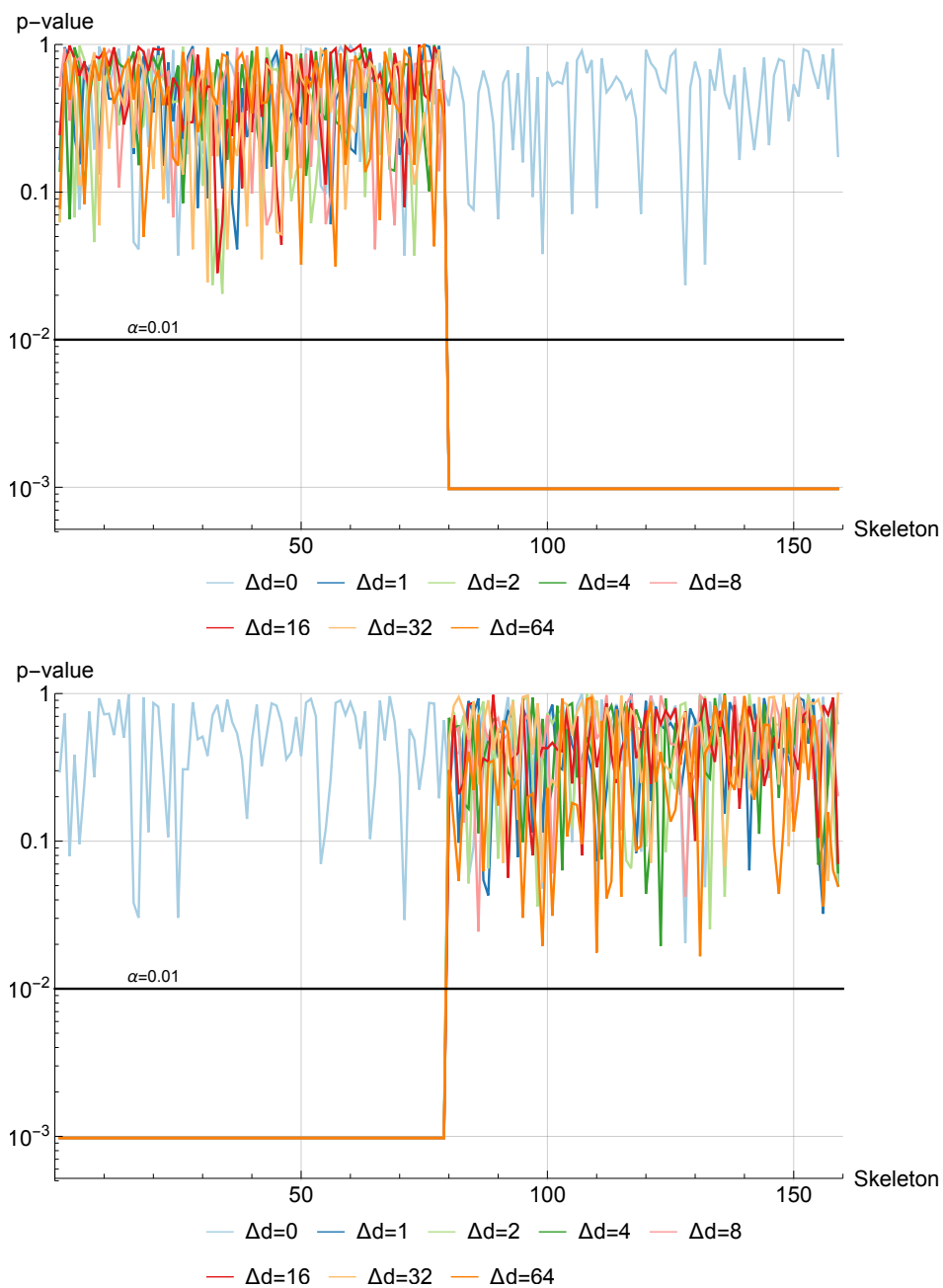
Figure 4.13: Case C2: p-values along the skeleton with respect to different Δd. The top image shows the p-values if the marker is placed in the upper object part. The bottom image reports the p-values if the marker is located in the lower object part.

Figure 4.14: Case C3: p-values along the skeleton with respect to different Δd. The top image shows the p-values if the marker is placed in the upper object part. The bottom image reports the p-values 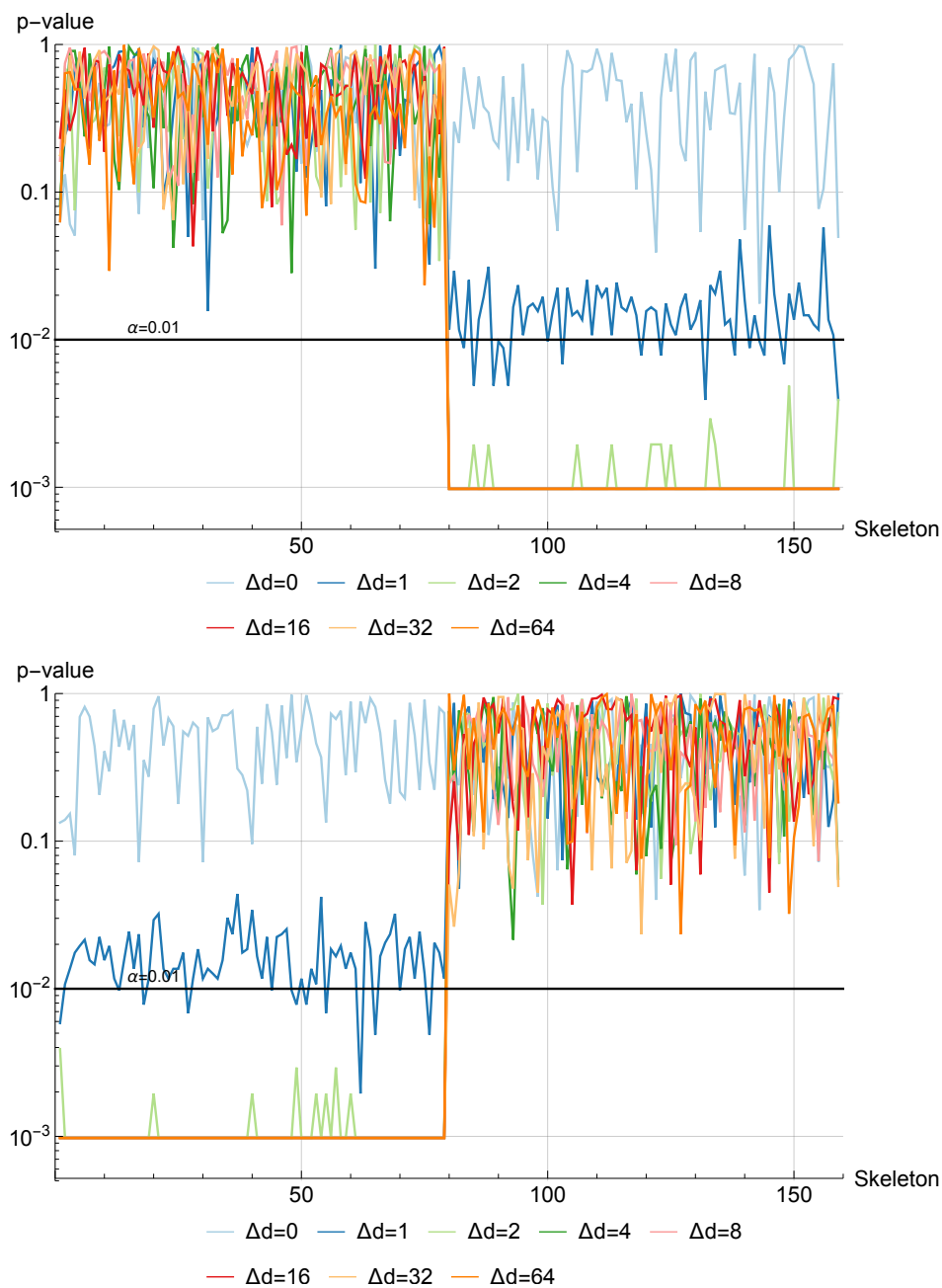if the marker is located in the lower object part. The significant differences in the case of $\Delta d = 0$ are caused by two different distributions (Gaussian, Poisson), used for the upper and the lower parts of the object respectively.

Figure 4.15: Case C4: p-values along the skeleton with respect to different Δd. The top image shows the p-values if the marker is placed in the upper object part. The bottom image reports the p-values if the marker is located in the lower object part. The significant differences in the case of $\Delta d = 0$ are caused by two different distributions (Gaussian, Poisson), used for the upper and the lower parts of the object respectively.

Figure 4.16: Case C5: p-values along the skeleton with respect to different Δd. The top image shows the p-values if the marker is placed in the upper object part. The bottom image reports the p-values if the marker is located in the lower object part.

Figure 4.17: Case C6: p-values along the skeleton with respect to different Δd. The top image shows the p-values if the marker is placed in the upper object part. The bottom image reports the p-values if the marker is located in the lower object part.
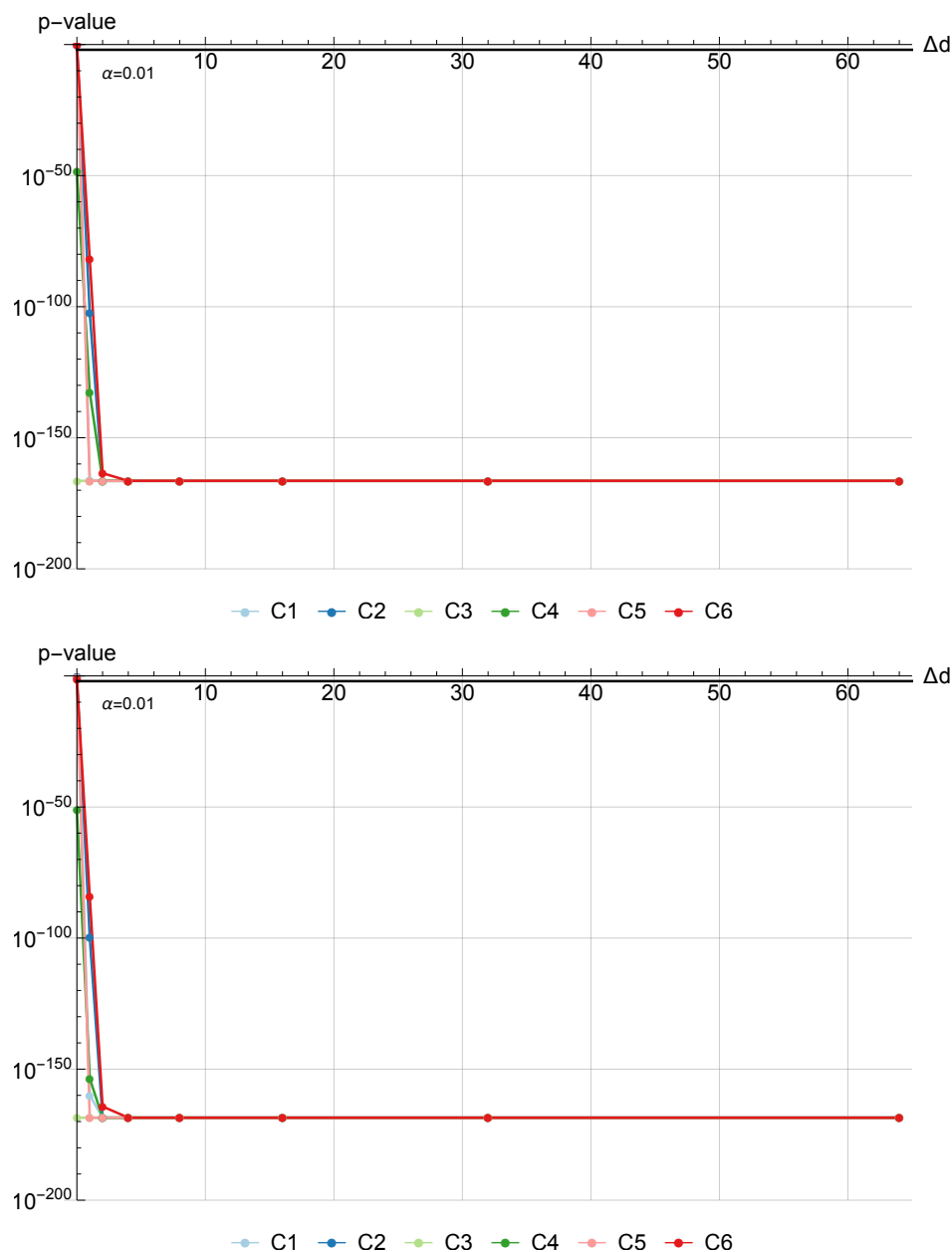
Figure 4.18: P-values for the upper (top image) and the lower (bottom image) parts of the object with respect to different cases and Δd. The single marker is placed either in the upper part (bottom image) or in the lower part (top image) of the object. At $\Delta d = 0$, the reported p-values are higher than the significance level α in the cases C1, C2, C5, C6, indicating no significant differences. In the cases C3 and C4, however, the p-values are below α. This corresponds to significant differences between two different distributions. With $\Delta d \geq 1$, the reported p-values are also below α. This implies significant differences between distributions.

No noise

Input mask from the data without noise

Reference solution, refined by the domain expert

Gaussian noise

| Pass 1 | Pass 2 | Pass 3 | Pass 4 | Pass 5 | Pass 6 |

Poisson noise

| Pass 1 | Pass 2 | Pass 3 | Pass 4 | Pass 5 | Pass 6 |
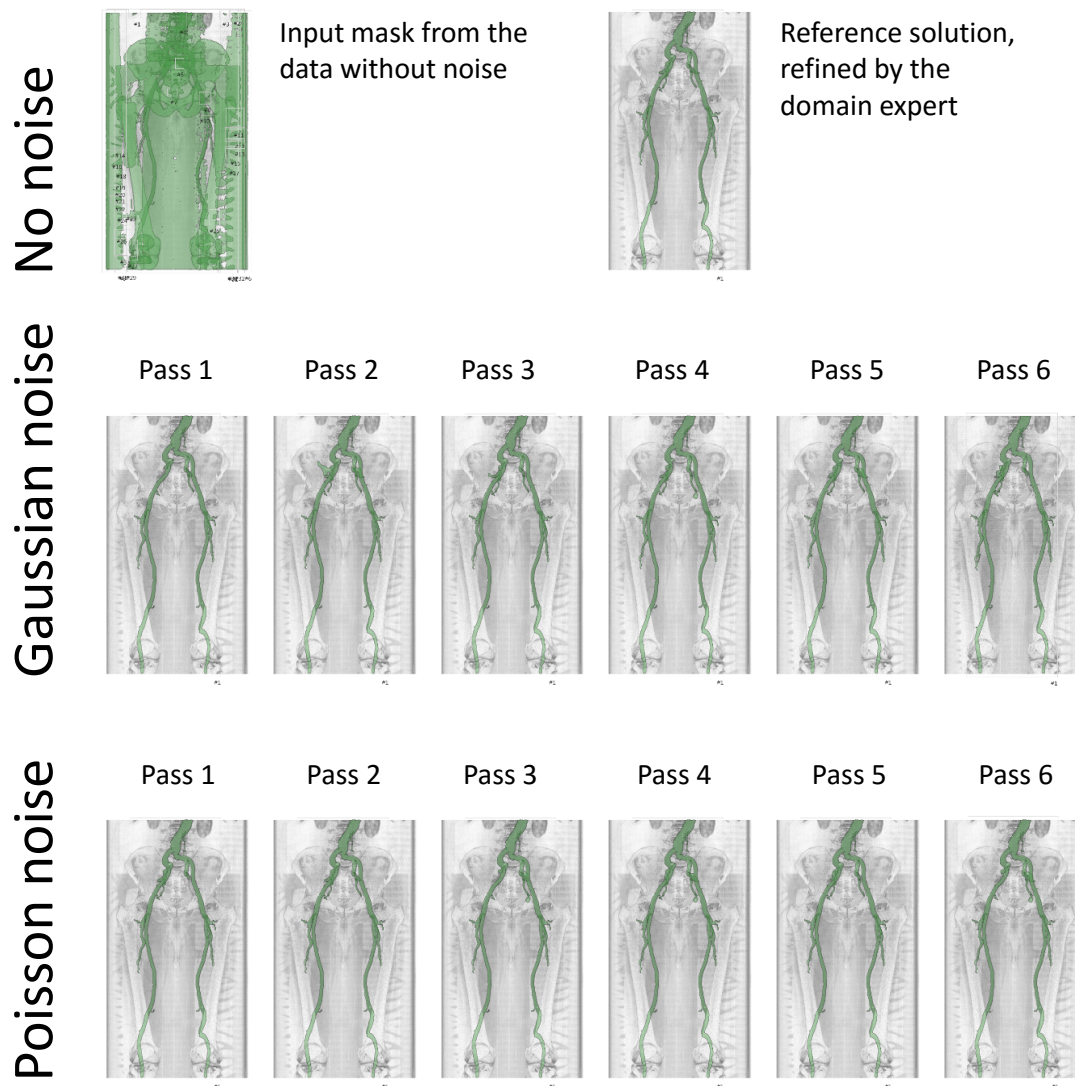
Figure 4.19: Comparison of the results with respect to different noise models.

Table 4.1: Detailed information on the volume editing experiment with artificial noise

	Gaussian noise					
	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5	Pass 6
Standard deviation σ	1	2	4	8	16	32
Estimated PSNR (dB)	65.1	59.9	54.3	48.4	42.6	36.9
Editing time (sec.)	103	79	116	150	86	101
Number of steps	8	3	7	6	5	5
Quality of result	0.972	0.949	0.971	0.943	0.935	0.884
	Poisson noise					
	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5	Pass 6
Mean μ	1	2	4	8	16	32
Estimated PSNR (dB)	66.3	63.3	60.4	57.3	54.3	51.4
Editing time (sec.)	91	81	116	122	192	102
Number of steps	4	3	7	10	9	5
Quality of result	0.979	0.980	0.971	0.974	0.969	0.948

Robustness of the Marker-based Reconstruction of the EDFs

We evaluated how different numbers of markers can reconstruct the complex distribution function of a real-world dataset with the following test. For this purpose, we used the median p-value across all atomic regions as a measure of consistency between the empirical distribution functions of the markers and the underlying data value distribution.

We randomly placed 50 markers inside the vessels of the human lower extremities in a rather challenging medical dataset (Figure 4.7). To estimate the stability of marker placement, we repeated the procedure 50 times. We set the significance level α to 0.05. In the test we used one CTA medical dataset. The dataset had 800 slices; each slice had resolution of 512×512 pixels. We used the vessel segmentation mask, edited and manually refined by the domain expert.

Resulting median p-values for 50 sets of randomly placed markers are shown in Figure 4.20. As expected, the more markers we allocate, the more stable is the reconstruction of the distribution function. This is reflected in relatively large median p-values, which became non-significant after five markers in all 50 trials. The more markers we use, the less is the median p-values variation. We may conclude that EDFs from five or more markers reconstruct the data-value distribution of the real-world object with sufficient precision.

4.4.3 Performance Tests

In order to test the performance of our statistical approach, we used an Intel Core i7-2600K 3.4 GHz CPU with 16 GB of RAM and an NVidia GeForce Titan X GPU. In the case of the temporal data with 3650 samples, the calculation time is 0.004 seconds with one marker, 0.006 seconds
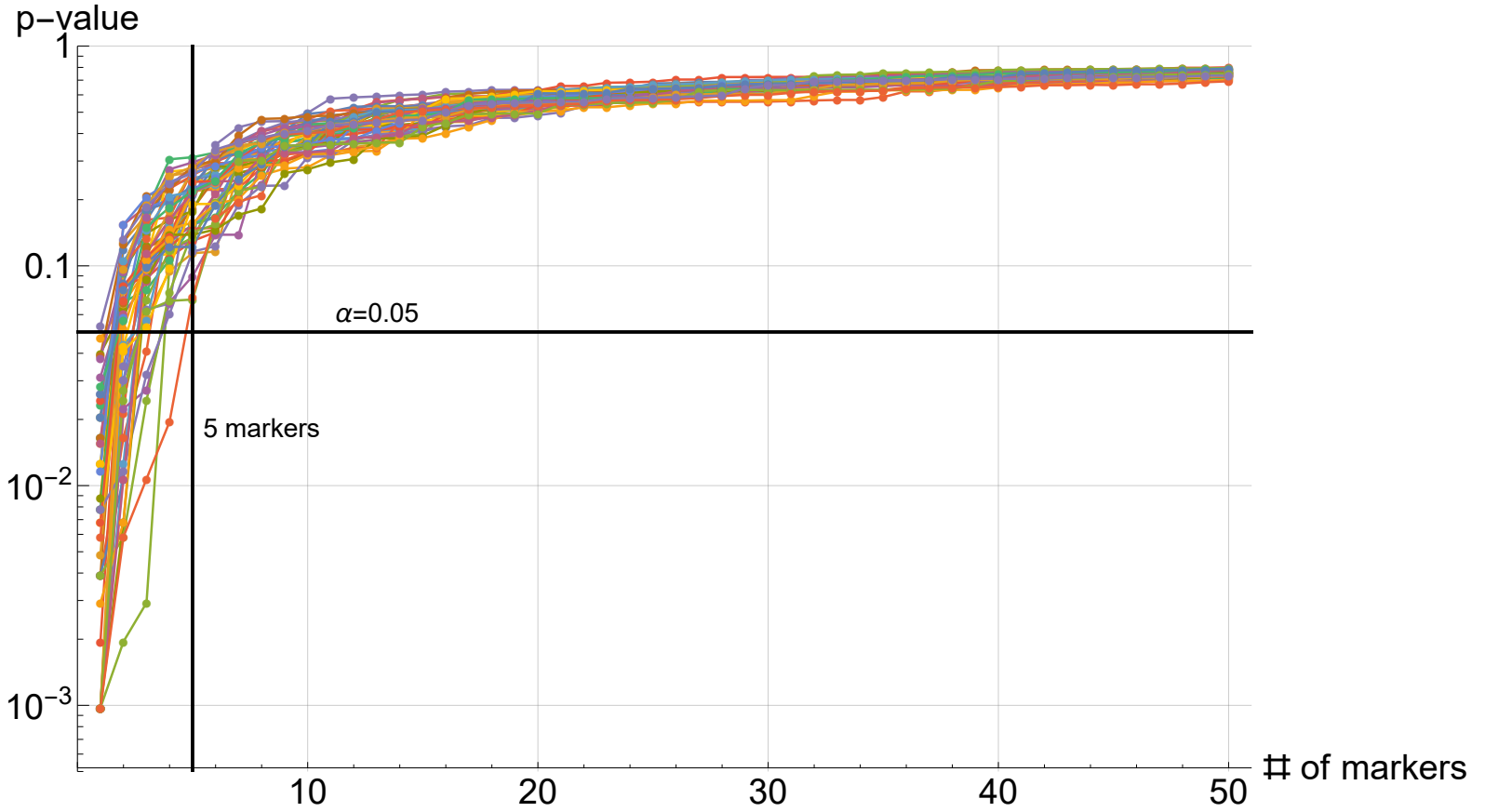
Figure 4.20: Median p-values for different number of markers, placed randomly inside the vessels. Each polyline represents one marker set. 50 marker sets were tried.

with two markers, 0.01 seconds with four markers, and 0.017 seconds with eight markers. For the spatial example we use ten CTA datasets of the human lower extremities with a slice resolution of 512×512 pixels and the number of slices ranging from 700 to 975. The datasets capture vessels and bones. The average calculation time is 0.51 seconds with one marker, 0.72 seconds with two markers, 1.16 seconds with four markers, and 2.05 seconds with eight markers.

4.5 Discussion and Limitations

We identified certain limitations of our statistical approach. Construction of the EDFs requires a certain number of samples to achieve a sufficient precision. However, this requirement may be violated in some atomic regions of the data. Therefore, the statistical tests obviously cannot be used in these regions. We display such regions with an additional color that is not associated with the ranking color scheme. The user has to evaluate dissimilarities in this kind of regions, which is a conventional data-analysis task. For example, this happens with the few small vessels of the human lower extremities in CTA data. One possibility to address this issue is to improve the sampling of the data acquisition technique.

The proper atomic regions realization is crucial for our statistical approach. If the atomic regions for some reason do not reflect important features of the data, then the results of our method, namely the composite regions, their hierarchy, and the statistical information, may be sub-optimal for the data exploration task. If this happens, the user is required to check the entire hierarchy manually in order to find regions that satisfy the chosen objective. For example, in the case of the three-dimensional spatial data it is important to capture spatial features. Our specialized realization, described in Section 4.4.2, does this by representing the data with the influence zones. In certain cases, such a representation is sub-optimal, *e.g.*, if two different structures are connected with each other by a large contact area. As a result, these two structures are not separated at the level of the atomic regions, and, therefore, they are treated by our statistical approach as belonging together. Particularly, this situation occurs in regions, where the small vessels touch the bones in the human lower extremities (CTA data). Additional domain knowledge may rectify this deficiency.

If there were a systematic difference between distributions of conceptually same regions of the data, caused by the data acquisition modality, our statistical approach would over-estimate the dissimilarity significance. In particular, data from Electron Microscopy (EM) imaging has a significant variation between the slices, acquired from the specimen, which is physically cut into thin slabs. One way to deal with such an over-estimation is to pre-process the data, reducing the undesired variation in it.

In case of degenerate distributions, which have only a single outcome value, the statistical tests may report extreme p-values – zeros and ones. However, this does not pose a limitation for our statistical approach, as the comparison by the p-values is still valid. Our statistical method does not work with data, where regions that are different according to the domain-specific logic exhibit the same data-value distribution. However, in this case other general approaches of comparison would probably fail as well, as domain-specific knowledge is required to differentiate such regions, indistinguishable by the data values alone.

4.6 Summary

We proposed a novel abstract concept for statistically quantifying dissimilarities between arbitrary regions of m-dimensional data. The dissimilarity significances are computed by hypothesis testing, based on robust and sound statistical concepts. To facilitate data exploration, we represent the data with different levels of detail. At each level, we localize the regions with the most and the least significant dissimilarities, aiding the user during data exploration and analysis. We evaluated the generality of our method with two concrete applications: temporal data exploration and segmentation editing. Our proposed data exploration protocol streamlines the user interaction in both scenarios, which is strengthened by an evaluation with domain experts.

Conclusion

The barriers are not erected which can say to aspiring talents and industry,
'Thus far and no farther.'

— Ludwig van Beethoven

*T*HIS thesis addresses the topic of guided interactive volume editing applied to medical image segmentation. First, a classification of the segmentation defects was introduced. Examples of each defect type were provided with respect to two different application scenarios: segmentation of the liver from abdominal data, extraction of the vessel tree from lower extremities data. Then, we investigated the related questions **Q1**-**Q5** and contributed to their solutions:

- **Q1**: How to interactively select the defects?

- **Q2**: How to choose the appropriate correction operations?

- **Q3**: How to automatically detect the introduced types of defects?

- **Q4**: How to visualize the detected defects?

- **Q5**: How to design the entire editing process?

We start with our basic volume-editing method, described in Chapter 2. This method uses the object representation by the topological means of the skeleton. The skeleton distance field and the influence zones allow us to transform interaction operations on the skeleton into the selection of parts of the object for the subsequent editing (**Q1**). After manually specifying the correction operations for the selected parts (**Q2**), the users can rectify deficiencies of the segmentation masks, reaching satisfactory results even in the case of severe pathological conditions. However, the precise selection of the defects is required.

To address this issue, we develop a method that automatically detects the following types of defects (**Q3**): O_P, O_L, U_P, and U_L. Combining the detected defects with pre-defined correction operations (**Q2**), we form suggestions for the editing process (**Q4**). The user selects necessary corrections from the suggested ones (**Q5**), instead of manually exploring the object and specifying the defects and the correction operations. Our defect detection method is described in detail in Chapter 3.

To streamline the exploration of the possibly vast space of the suggestions, we utilize the statistical approach from Chapter 4 (**Q3**). Each defective region is compared against reference parts of the object. The dissimilarities in data-value distributions are computed and conveyed to the user. All the suggestions are then ordered according to their dissimilarities (**Q4**) and presented in such a way, that the users can focus on correcting severe defects first (**Q5**). The time of the correction process is shorter, compared to a technique utilized in clinical routine on a daily basis. In terms of interaction efforts, our proposed method requires less interaction than the clinical technique.

Even though the methods, described in this thesis, bring improvements to the medical image segmentation field, certain aspects still require further research attention. First, the influence zones can be generalized to support membership functions, which would map the skeleton voxels to object voxels with a certain probability. Such a representation may improve our proposed techniques, as currently many issues come from the discrete nature of the influence zones. One issue is the assignment of each voxel to the most common influence zone in the neighborhood. Contribution of a voxel to other influence zones is discarded. In fact, such a heuristic introduces deviations from the original topology of the object that might be sub-optimal for defects detection and correction.

Correction of the under-estimation defects can be revisited, as currently it relies on the complement of the object. As such, editing of the complement makes sense for individual organs, where U_P defects are rather compact. However, in case of objects with complex structure, *e.g.*, the entire vessel trees, the complement is a sub-optimal choice. In order to capture U_P defects in vessel trees, the complement should take a large part of the background that contains numerous structures. This leads to a high number of false positives at the defects detection. One option could be to track the skeleton branches locally and extend them outside of the object. In this case, a smaller part of the background is analyzed, therefore, the false positive rate can be acceptable. This idea is also a heuristic for tubular objects, therefore, a general solution of this problem is an open scientific question.

More elaborate evaluations of the proposed methods are required in order to assess their suitability for the clinical environment. As demonstrated in Chapter 3, our techniques can be used for different data modalities, such as industrial X-Ray Computed Tomography (XCT) or biological Electron Microscopy (EM) acquisition techniques. To this end, a broad evaluation of our proposed methods, possibly with new classes of objects, can lead to interesting findings as well as further improvements to the volume-editing domain.

Currently, a lot of research in the domain is done in the area of uncertainty quantification and visualization. The idea here is to provide to the user uncertainty information related to the

segmentation task. This may help with the segmentation of datasets exhibiting severe pathologies. This research direction is already outlined by the works of Praßni et al. [30] and Top et al. [31].

The major open question in the field, in our opinion, is how to compare numerous segmentation methods together. Absence of the standardized quality metrics and benchmark datasets makes it almost impossible to quantitatively assess the benefits of the newly developed techniques. Certain steps in this direction are made by MICCAI Grand Challenges on biomedical image analysis [128]. However, it is not yet clear how to compare interaction efforts required by different techniques. Extended evaluation of the existing methods in terms of the interaction and usability, therefore, is an interesting direction for future work.

A

Questionnaire for Guided Volume Editing based on Histogram Dissimilarity

Focus on competition has always been a formula for mediocrity.

— Daniel Burrus

*T*HIS questionnaire is designed for the evaluation of the Guided Volume Editing technique, based on Histogram Dissimilarity. Each question is answered according to the five-level Likert scale. The technique is described in Chapter 3. The main contribution of this technique is an automatic detection of segmentation defects and generation of suggestions for their correction. First, the System Usability Score [87] of the proposed method is assessed. Then, the user gives detailed scores to individual components of the technique. Finally, the comparison of our technique with two other approaches is performed.

System usability study	Strongly disagree		...		Strongly agree	
I think that I would like to use this system frequently	☐	☐	☐	☐	☐	
I found the system unnecessarily complex	☐	☐	☐	☐	☐	
I thought the system was easy to use	☐	☐	☐	☐	☐	
I think that I would need the support of a technical person to be able to use this system	☐	☐	☐	☐	☐	
I found the various functions in this system were well integrated	☐	☐	☐	☐	☐	
I thought there was too much inconsistency in this system	☐	☐	☐	☐	☐	
I would imagine that most people would learn to use this system very quickly	☐	☐	☐	☐	☐	
I found the system very cumbersome to use	☐	☐	☐	☐	☐	
I felt very confident using the system	☐	☐	☐	☐	☐	
I needed to learn a lot of things before I could get going with this system	☐	☐	☐	☐	☐	

System functionality study	Strongly disagree		...		Strongly agree	
I felt very confident by being guided throughout the correction process	☐	☐	☐	☐	☐	
I got a good overview of the correction possibilities with the miniatures and correction previews	☐	☐	☐	☐	☐	
I could apply precise corrections using the level-of-detail feature	☐	☐	☐	☐	☐	
I found the suggested corrections, provided by this system, useful	☐	☐	☐	☐	☐	
I found the miniatures helpful for exploring the available correction possibilities	☐	☐	☐	☐	☐	
I found the preview of the correction operation together with the automatically shown slice views sufficiently informative for deciding to accept or reject the operation	☐	☐	☐	☐	☐	

Comparative study

T1	Structure-aware guided volume editing
T2	Skeleton-based volume editing
TX	Other

Evaluation of technique T1	Strongly disagree		...		Strongly agree	
The correction with the technique **T1** requires little amount of interaction	☐	☐	☐	☐	☐	
I achieved appropriate results with the technique **T1** in reasonably short time	☐	☐	☐	☐	☐	
I found the technique **T1** easy to interact with	☐	☐	☐	☐	☐	
I found that the technique **T1** is tolerant regarding user input	☐	☐	☐	☐	☐	
I am satisfied with the results obtained with the technique **T1**	☐	☐	☐	☐	☐	

Evaluation of technique T2	Strongly disagree		...		Strongly agree	
The correction with the technique **T2** requires little amount of interaction	☐	☐	☐	☐	☐	
I achieved appropriate results with the technique **T2** in reasonably short time	☐	☐	☐	☐	☐	
I found the technique **T2** easy to interact with	☐	☐	☐	☐	☐	
I found that the technique **T2** is tolerant regarding user input	☐	☐	☐	☐	☐	
I am satisfied with the results obtained with the technique **T2**	☐	☐	☐	☐	☐	

Evaluation of technique of my choice (TX)	Strongly disagree		…		Strongly agree	
Technique (please, state the name of the technique and provide the reference, if possible):						
The correction with the technique **TX** requires little amount of interaction	☐	☐	☐	☐	☐	
I achieved appropriate results with the technique **TX** in reasonably short time	☐	☐	☐	☐	☐	
I found the technique **TX** easy to interact with	☐	☐	☐	☐	☐	
I found that the technique **TX** is tolerant regarding user input	☐	☐	☐	☐	☐	
I am satisfied with the results obtained with the technique **TX**	☐	☐	☐	☐	☐	

Consent

I voluntarily agreed to this user study and had option to terminate it at any time	☐ I disagree	☐ I agree
I agree with usage of this user study data in publication (after mandatory anonymization)	☐ I disagree	☐ I agree

Name of participant: Click here to enter text.

Date: Click here to enter a date.

Questionnaire for Statistics-Driven Localization of Dissimilarities in Data

Cherish the friend who tells you a harsh truth,
wanting ten times more to tell you a loving lie.

— Robert Brault

W E used the following questionnaire during the evaluation of Statistics-Driven Localization of Dissimilarities in Data, applied to the segmentation editing topic. The questionnaire is based on a five-level Likert scale. The method was implemented according to the description in Chapter 4. This technique contributes to the correction process with a statistical assessment of the detected defective regions and their ordered presentation to the user. In order to assess the usability of the technique, we use the System Usability Score [87]. Individual components are then assessed by the users. We compare this technique with the previous method from Chapter 3 as well as with the AngioVis framework [1], used in clinical routine on a daily basis.

System usability study	Strongly disagree		...		Strongly agree	
I think that I would like to use this system frequently	☐	☐	☐	☐	☐	
I found the system unnecessarily complex	☐	☐	☐	☐	☐	
I thought the system was easy to use	☐	☐	☐	☐	☐	
I think that I would need the support of a technical person to be able to use this system	☐	☐	☐	☐	☐	
I found the various functions in this system were well integrated	☐	☐	☐	☐	☐	
I thought there was too much inconsistency in this system	☐	☐	☐	☐	☐	
I would imagine that most people would learn to use this system very quickly	☐	☐	☐	☐	☐	
I found the system very cumbersome to use	☐	☐	☐	☐	☐	
I felt very confident using the system	☐	☐	☐	☐	☐	
I needed to learn a lot of things before I could get going with this system	☐	☐	☐	☐	☐	

System functionality study	Strongly disagree		...		Strongly agree	
I felt very confident by having statistical information available	☐	☐	☐	☐	☐	
I found the statistical significance, provided by this system, very useful	☐	☐	☐	☐	☐	
I got a good overview of the correction possibilities with the radial tree view	☐	☐	☐	☐	☐	
I could find precise corrections following the suggested paths in the radial tree view	☐	☐	☐	☐	☐	
I found the marker-based specification of the object of interest very easy	☐	☐	☐	☐	☐	
I located necessary corrections using color-based encoding of statistical information	☐	☐	☐	☐	☐	
I found the marker-based specification of the object of interest precise enough	☐	☐	☐	☐	☐	
I got a good overview of the correction process with the time line view	☐	☐	☐	☐	☐	

Comparative study

T1	Statistics-guided volume editing
T2	Dissimilarity-guided volume editing
T3	Base-line volume editing

Evaluation of technique T1	Strongly disagree		…		Strongly agree	
The correction with the technique **T1** requires little amount of interaction	☐	☐	☐	☐	☐	
I achieved appropriate results with the technique **T1** in reasonably short time	☐	☐	☐	☐	☐	
I found the technique **T1** easy to interact with	☐	☐	☐	☐	☐	
I found that the technique **T1** is tolerant regarding user input	☐	☐	☐	☐	☐	
I am satisfied with the results obtained with the technique **T1**	☐	☐	☐	☐	☐	
I got sufficient information while performing the correction with the technique **T1**	☐	☐	☐	☐	☐	

Evaluation of technique T2	Strongly disagree		…		Strongly agree	
The correction with the technique **T2** requires little amount of interaction	☐	☐	☐	☐	☐	
I achieved appropriate results with the technique **T2** in reasonably short time	☐	☐	☐	☐	☐	
I found the technique **T2** easy to interact with	☐	☐	☐	☐	☐	
I found that the technique **T2** is tolerant regarding user input	☐	☐	☐	☐	☐	
I am satisfied with the results obtained with the technique **T2**	☐	☐	☐	☐	☐	
I got sufficient information while performing the correction with the technique **T2**	☐	☐	☐	☐	☐	

Evaluation of technique T3	Strongly disagree		…		Strongly agree	
The correction with the technique **T3** requires little amount of interaction	☐	☐	☐	☐	☐	
I achieved appropriate results with the technique **T3** in reasonably short time	☐	☐	☐	☐	☐	
I found the technique **T3** easy to interact with	☐	☐	☐	☐	☐	
I found that the technique **T3** is tolerant regarding user input	☐	☐	☐	☐	☐	
I am satisfied with the results obtained with the technique **T3**	☐	☐	☐	☐	☐	
I got sufficient information while performing the correction with the technique **T3**	☐	☐	☐	☐	☐	

Additional comments (if any)

Click here to enter text.

Consent

I voluntarily agreed to this user study and had option to terminate it at any time	☐ I disagree	☐ I agree
I agree with usage of this user study data in publication (after mandatory anonymization)	☐ I disagree	☐ I agree

Name of participant: Click here to enter text.

Date: Click here to enter a date.

Bibliography

[1] D. Fleischmann, M. Straka, J. Roos, J. Lammer, R. Schernthaner, R. Schernthaner, M. Šrámek, A. Varchola, V. Solcany, and E. Gröller. (2007). AngioVis framework. Accessed: 2015-Oct-25, [Online]. Available: http://www.angiovis.org/.

[2] E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition", in *SIGGRAPH '95 Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 191–198. DOI: 10.1145/218380.218442.

[3] K. C.-H. Wong, P.-A. Heng, and T.-T. Wong, "Accelerating "intelligent scissors"using slimmed graphs", *Journal of Graphics Tools*, vol. 5, no. 2, pp. 1–13, 2000. DOI: 10.1080/10867651.2000.10487520.

[4] A. Mishra, E. Wong, W. Zhang, D. Clausi, and P. Fieguth, "Improved interactive medical image segmentation using enhanced intelligent scissors (EIS)", in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2008, pp. 3083–3086. DOI: 10.1109/IEMBS.2008.4649855.

[5] P. Hastreiter and T. Ertl, "Fast and interactive 3D-segmentation of medical volume data", in *Computer Graphics International 98*, 1998, pp. 78–85.

[6] A. X. Falcão and J. K. Udupa, "A 3D generalization of user-steered live-wire segmentation", *Medical image analysis*, vol. 4, pp. 389–402, 2000. DOI: 10.1016/S1361-8415(00)00023-2.

[7] G. Hamarneh, J. Yang, C. McIntosh, and M. Langille, "3D live-wire-based semi-automatic segmentation of medical images", in *SPIE*, vol. 5747, 2005, pp. 1597–1603. DOI: 10.1117/12.596148.

[8] M. Poon, G. Hamarneh, and R. Abugharbieh, "Efficient interactive 3D livewire segmentation of complex objects with arbitrary topology", *Computerized Medical Imaging and Graphics*, vol. 32, no. 8, pp. 639–650, 2008. DOI: 10.1016/j.compmedimag.2008.07.004.

[9] C. J. Armstrong, B. L. Price, and W. A. Barrett, "Interactive segmentation of image volumes with Live Surface", *Computer & Graphics*, vol. 31, no. 2, pp. 212–229, 2007. DOI: 10.1016/j.cag.2006.11.015.

[10] A. Beck and V. Aurich, "HepaTux - A semiautomatic liver segmentation system", in *Proceedings of MICCAI workshop on 3D segmentation in the clinic: A Grand Challenge*, T. Heimann, M. Styner, and B. van Ginneken, Eds., 2007, pp. 225–233.

[11] L. Liu, D. Raber, and D. Nopachai, "Interactive separation of segmented bones in CT volumes using graph cut", *Lecture Notes in Computer Science*, vol. 5241, pp. 296–304, 2008. DOI: 10.1007/978-3-540-85988-8_36.

[12] R. Beichel, C. Bauer, A. Bornik, E. Sorantin, and H. Bischof, "Liver segmentation in CT data: A segmentation refinement approach", in *Proceedings of MICCAI Workshop on 3D Segmentation in The Clinic*, T. Heimann, M. Styner, and B. van Ginneken, Eds., 2007, pp. 235–245.

[13] J. Egger, R. R. Colen, B. Freisleben, and C. Nimsky, "Manual refinement system for graph-based segmentation results in the medical domain", *Journal of Medical Systems*, vol. 36, no. 5, pp. 2829–2839, 2012. DOI: 10.1007/s10916-011-9761-7.

[14] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation", *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006. DOI: 10.1007/s11263-006-7934-5.

[15] Y. Boykov and O. Veksler, "Graph cuts in vision and graphics: Theories and applications", in *Handbook of Mathematical Models in Computer Vision*, N. Paragios, Y. Chen, and O. Faugeras, Eds., Springer US, 2006, ch. 5, pp. 79–96, ISBN: 0-387-26371-3. DOI: 10.1007/0-387-28831-7.

[16] F. Heckel, J. Moltz, C. Tietjen, and H. K. Hahn, "Sketch-based editing tools for tumour segmentation in 3D medical images", *Computer Graphics Forum*, vol. 32, no. 8, pp. 144–157, 2013. DOI: 10.1111/cgf.12193.

[17] T. Heimann, M. Thorn, T. Kunert, and H.-P. Meinzer, "New methods for leak detection and contour correction in seeded region growing segmentation", in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, 2004, pp. 317–322.

[18] P. Salz, A. Reske, H. Wrigge, G. Scheuermann, and H. Hagen, *User-guided segmentation of thoracic computed tomography data for electrical impedance tomography image reconstruction*, Poster at the 2nd IEEE Symposium on Biological Data Visualization, 2012.

[19] A. Bornik, R. Beichel, and D. Schmalstieg, "Interactive editing of segmented volumetric datasets in a hybrid 2D/3D virtual environment", in *ACM Symposium on Virtual Reality Software and Technology*, 2006, pp. 197–206. DOI: 10.1145/1180495.1180536.

[20] D. Proksch, J. Dornheim, and B. Preim, "Interaktionstechniken zur Korrektur medizinischer 3D-Segmentierungen", in *Bildverarbeitung für die Medizin*, vol. 574, 2010, pp. 420–424.

[21] M. Jackowski, M. Satter, and A. Goshtasby, "Approximating digital 3D shapes by rational Gaussian surfaces", *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 1, pp. 56–69, 2003. DOI: 10.1109/TVCG.2003.1175097.

[22] H. Timinger, V. Pekar, J. von Berg, K. Dietmayer, and M. Kaus, "Integration of interactive corrections to model-based segmentation algorithms", in *Bildverarbeitung für die Medizin*, 2003, pp. 171–175.

[23] N. El-Zehiry, M. Jolly, and M. Sofka, "A splice-guided data driven interactive editing", in *Proceedings of IEEE International Symposium on Biomedical Imaging*, 2013, pp. 6–9. DOI: `10.1109/ISBI.2013.6556670`.

[24] L. Grady and G. Funka-Lea, "An energy minimization approach to the data driven editing of presegmented images / volumes", *Lecture Notes in Computer Science*, vol. 4191, pp. 888–95, 2006. DOI: `10.1007/11866763_109`.

[25] H. K. Hahn, M. T. Wenzel, O. Konrad-Verse, and H.-O. Peitgen, "A minimally-interactive watershed algorithm designed for efficient CTA bone removal", *Computer Vision Approaches to Medical Image Analysis*, vol. 4241, pp. 178–189, 2006. DOI: `10.1007/11889762_16`.

[26] F. Heckel and S. Braunewell, "A concept for the application of a hierarchical image subdivision to the segmentation editing problem", in *Proceedings of MICCAI Workshop on Interactive Medical Image Computing*, 2014.

[27] H. K. Hahn and H.-O. Peitgen, "IWT – interactive watershed transform: A hierarchical method for efficient interactive and automated segmentation of multidimensional grayscale images", in *Proceedings of SPIE Medical Imaging*, vol. 5032, 2003, pp. 643–653. DOI: `10.1117/12.481097`.

[28] R. Beichel, S. Mitchell, E. Sorantin, F. Leberl, A. Goshtasby, and M. Sonka, "Shape- and appearance-based segmentation of volumetric medical images", in *International Conference on Image Processing*, 2001, pp. 589–592. DOI: `10.1109/ICIP.2001.958561`.

[29] A. Neumann and C. Lorenz, "Statistical shape model based segmentation of medical images", *Computerized Medical Imaging and Graphics*, vol. 22, no. 2, pp. 133–143, 1998. DOI: `10.1016/S0895-6111(98)00015-9`.

[30] J. S. Praßni, T. Ropinski, J. Mensmann, and K. Hinrichs, "Shape-based transfer functions for volume visualization", in *Proceedings of PacificVis*, 2010, pp. 9–16. DOI: `10.1109/PACIFICVIS.2010.5429624`.

[31] A. Top, G. Hamarneh, and R. Abugharbieh, "Spotlight: Automated confidence-based user guidance for increasing efficiency in interactive 3D image segmentation", *Lecture Notes in Computer Science*, vol. 6533, pp. 204–213, 2011. DOI: `10.1007/978-3-642-18421-5_20`.

[32] T. Torsney-Weir, A. Saad, T. Möller, H. C. Hege, B. Weber, and J. M. Verbavatz, "Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration", *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1892–1901, 2011. DOI: `10.1109/TVCG.2011.248`.

[33] B. Fröhler, T. Möller, and C. Heinzl, *GEMSe: Visualization-guided exploration of multi-channel volume segmentation algorithms*, Accepted to EuroVis 2016, 2016.

[34] S. D. Olabarriaga and A. W. M. Smeulders, "Interaction in the segmentation of medical images: A survey", *Medical image analysis*, vol. 5, no. 2, pp. 127–142, 2001. DOI: `10.1016/S1361-8415(00)00041-4`.

[35] F. Heckel, J. H. Moltz, H. Meine, B. Geisler, A. Kießling, M. D'Anastasi, D. P. dos Santos, A. J. Theruvath, and H. K. Hahn, "On the evaluation of segmentation editing tools", *Journal of Medical Imaging*, vol. 1, no. 3, pp. 034005-01–034005-16, 2014. DOI: 10.1117/1.JMI.1.3.034005.

[36] T. Mönch, S. Adler, and B. Preim, "Staircase-aware smoothing of medical surface meshes", in *Visual Computing for Biology and Medicine*, 2010, pp. 83–90. DOI: 10.2312/VCBM/VCBM10/083-090.

[37] H. Hauser, L. Mroz, G.-I. Bischi, and M. E. Gröller, "Two-level volume rendering", *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 3, pp. 242–252, 2001. DOI: 10.1109/2945.942692.

[38] M. Hadwiger, C. Berger, and H. Hauser, "High-quality two-level volume rendering of segmented data sets on consumer graphics hardware", *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, pp. 301–308, 2003. DOI: 10.1109/VISUAL.2003.1250386.

[39] S. Lindholm and A. Bock, "Poor man's rendering of segmented data", in *SIGRAD 2013, Visual Computing, June 13-14, 2013, Norrköping, Sweden*, 2013, pp. 49–54.

[40] Z. Salah, D. Bartz, and W. Straßer, "Illustrative rendering of segmented anatomical data", in *Simulation und Visualisierung*, 2005, pp. 175–184.

[41] H. Müller and M. Wehle, "Visualization of implicit surfaces using adaptive tetrahedrizations", in *Scientific Visualization Conference*, 1997, pp. 243–250, ISBN: 0-7695-0505-8.

[42] G. Taubin, "Curve and surface smoothing without shrinkage", in *IEEE International Conference on Computer Vision*, 1995, pp. 852–857. DOI: 10.1109/ICCV.1995.466848.

[43] P. Knowles, G. Leach, and F. Zambetta, "Fast sorting for exact OIT of complex scenes", *Visual Computer*, vol. 30, no. 6-8, pp. 603–613, 2014. DOI: 10.1007/s00371-014-0956-z.

[44] T. C. Lee, R. L. Kashyap, and C. N. Chu, "Building skeleton models via 3D medial surface/axis thinning algorithms", *Lecture Notes in Computer Science*, vol. 56, pp. 462–478, 1994. DOI: 10.1006/cgip.1994.1042.

[45] K. Palágyi and A. Kuba, "A parallel 3D 12-subiteration thinning algorithm", *Graphical Models and Image Processing*, vol. 61, no. 4, pp. 199–221, 1999. DOI: 10.1006/gmip.1999.0498.

[46] K. Palágyi, E. Sorantin, E. Balogh, A. Kuba, C. Halmai, B. Erdohelyi, and K. Hausegger, "A sequential 3D thinning algorithm and its medical applications", in *Proceedings of the 17th International Conference on Information Processing in Medical Imaging*, ser. IPMI '01, 2001, pp. 409–415. DOI: 10.1007/3-540-45729-1_42.

[47] C. Pudney, "Distance-ordered homotopic thinning: A skeletonization algorithm for 3D digital images", *Computer Vision and Image Understanding*, vol. 72, no. 3, pp. 404–413, 1998. DOI: 10.1006/cviu.1998.0680.

[48] Y. Zhou, A. Kaufman, and A. W. Toga, "Three-dimensional skeleton and centerline generation based on an approximate minimum distance field", *The Visual Computer*, vol. vi, pp. 303–314, 1998. DOI: `10.1007/s003710050142`.

[49] N. Gagvani and D. Silver, "Parameter-controlled volume thinning", *Graphical Models and Image Processing*, vol. 61, no. 3, pp. 149–164, May 1999. DOI: `10.1006/gmip.1999.0495`.

[50] L. Wade and R. E. Parent, "Automated generation of control skeletons for use in animation", *Visual Computer*, vol. 18, no. 2, pp. 97–110, 2002. DOI: `10.1007/s003710100139`.

[51] R. Strzodka and A. Telea, "Generalized distance transforms and skeletons in graphics hardware", in *VisSym 2004, Symposium on Visualization*, 2004, pp. 221–230. DOI: `10.2312/VisSym/VisSym04/221-230`.

[52] M. A. M. M. van Dortmont, H. M. M. van de Wetering, and A. C. Telea, "Skeletonization and distance transforms of 3D volumes using graphics hardware", in *Proceedings of the 13th international conference on Discrete Geometry for Computer Imagery*, ser. DGCI'06, 2006, pp. 617–629. DOI: `10.1007/11907350_52`.

[53] H. Xia and P. G. Tucker, "Distance solutions for medial axis transform", in *Proceedings of 18th International Meshing Roundtable*, ser. IMR '09, 2009, pp. 247–265. DOI: `10.1007/978-3-642-04319-2_15`.

[54] M. W. Jones, J. A. Baerentzen, and M. Šrámek, "3D distance fields: A survey of techniques and applications", *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 581–599, 2006. DOI: `10.1109/TVCG.2006.56`.

[55] R. L. Ogniewicz and O. Kübler, "Hierarchic Voronoi skeletons", *Pattern Recognition*, vol. 28, no. 3, pp. 343–359, 1995. DOI: `10.1016/0031-3203(94)00105-U`.

[56] D. Attali and A. Montanvert, "Computing and simplifying 2D and 3D continuous skeletons", *Computer Vision and Image Understanding*, vol. 67, no. 3, pp. 261–273, 1997. DOI: `10.1006/cviu.1997.0536`.

[57] M. Näf, G. Székely, R. Kikinis, M. E. Shenton, and O. Kübler, "3D Voronoi skeletons and their usage for the characterization and recognition of 3D organ shape", *Computer Vision and Image Understanding*, vol. 66, no. 2, pp. 147–161, 1997. DOI: `10.1006/cviu.1997.0610`.

[58] R. Ogniewicz and M. Ilg, "Voronoi skeletons: Theory and applications", in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1992, pp. 63–69. DOI: `10.1109/CVPR.1992.223226`.

[59] D. Reniers, J. van Wijk, and A. Telea, "Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure", *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 2, pp. 355–368, 2008. DOI: `10.1109/TVCG.2008.23`.

[60] H. Blum, "A transformation for extracting new descriptors of shape", in *Models for the perception of speech and visual form*, 5, W. Wathen-Dunn, Ed., vol. 19, Cambridge, MA: MIT Press, 1967, pp. 362–380.

[61] N. D. Cornea, D. Silver, X. Yuan, and R. Balasubramanian, "Computing hierarchical curve-skeletons of 3D objects", *The Visual Computer*, vol. 21, pp. 945–955, 2005. DOI: `10.1007/s00371-005-0308-0`.

[62] N. D. Cornea, D. Silver, and P. Min, "Curve-skeleton applications", in *Proceedings of 16th IEEE Visualization Conference (VIS 2005)*, 2005, pp. 95–102. DOI: `10.1109/VISUAL.2005.1532783`.

[63] D. Reniers and A. Telea, "Segmenting simplified surface skeletons", in *Proceedings of the 14th IAPR international conference on Discrete geometry for computer imagery*, ser. DGCI'08, 2008, pp. 262–274. DOI: `10.1007/978-3-540-79126-3_24`.

[64] S. Prohaska, "Skeleton-based visualization of massive voxel objects with network-like architecture", PhD thesis, Potsdam University, 2007.

[65] D. Reniers, "Skeletonization and segmentation of binary voxel shapes", PhD thesis, Technische Universiteit Eindhoven, 2009. DOI: `10.6100/IR639872`.

[66] P. K. Saha, G. Borgefors, and G. Sanniti di Baja, "A survey on skeletonization algorithms and their applications", *Pattern Recognition Letters*, pp. 1–10, 2015. DOI: `10.1016/j.patrec.2015.04.006`.

[67] A. Karimov, G. Mistelbauer, J. Schmidt, P. Mindek, E. Schmidt, T. Sharipov, S. Bruckner, and E. Gröller, "ViviSection: Skeleton-based volume editing", *Computer Graphics Forum*, vol. 32, pp. 461–470, 2013. DOI: `10.1111/cgf.12133`.

[68] T. Heimann, B. van Ginneken, M. Styner, Y. Arzhaeva, V. Aurich, C. Bauer, A. Beck, C. Becker, R. Beichel, G. Bekes, F. Bello, G. K. Binnig, H. Bischof, A. Bornik, P. Cashman, Y. Chi, A. Cordova, B. M. Dawant, M. Fidrich, J. D. Furst, D. Furukawa, L. Grenacher, J. Hornegger, D. Kainmüller, R. Kitney, H. Kobatake, H. Lamecker, T. Lange, J. Lee, B. Lennon, R. Li, S. Li, H.-P. Meinzer, G. Németh, D. S. Raicu, A.-M. Rau, E. M. van Rikxoort, M. Rousson, L. Ruskó, K. A. Saddi, G. Schmidt, D. Seghers, A. Shimizu, P. Slagmolen, E. Sorantin, G. Soza, R. Susomboon, J. M. Waite, A. Wimmer, and I. Wolf, "Comparison and evaluation of methods for liver segmentation from CT datasets", *IEEE Transactions on Medical Imaging*, vol. 28, no. 8, pp. 1251–1265, 2009. DOI: `10.1109/TMI.2009.2013851`.

[69] H. Homann. (2007). Insight Journal - Implementation of a 3D thinning algorithm. Published: 2007-Dec-10. Accessed: 2013-Mar-25, [Online]. Available: `http://hdl.handle.net/1926/1292`.

[70] S. Ramesh and A. J. Sanyal, "Evaluation and management of non-alcoholic steatohepatitis", *Journal of Hepatology*, vol. 42, no. 1, Supplement, S2–S12, 2005.

[71] E. O'Riordan, C. M. Craven, D. Wilson, and P. J. Robinson, "Dual phase hepatic CT: Influence of scanning direction on liver attenuation", *American Journal of Roentgenology*, vol. 174, no. 5, pp. 1417–1421, 2000.

[72] K. Suzuki, M. L. Epstein, R. Kohlbrenner, S. Garg, M. Hori, A. Oto, and R. L. Baron, "Quantitative radiology: Automated CT liver volumetry compared with interactive volumetry and manual volumetry", *American Journal of Roentgenology*, vol. 197, no. 4, W706–W712, 2011. DOI: `10.2214/AJR.10.5958`.

[73] A. Karimov, G. Mistelbauer, T. Auzinger, and S. Bruckner, "Guided volume editing based on histogram dissimilarity", *Computer Graphics Forum*, vol. 34, pp. 91–100, 2015. DOI: `10.1111/cgf.12621`.

[74] G. Mistelbauer, A. Morar, A. Varchola, R. Schernthaner, I. Baclija, A. Köchl, A. Kanitsar, S. Bruckner, and E. Gröller, "Vessel visualization using curvicircular feature aggregation", *Computer Graphics Forum*, vol. 32, pp. 231–240, 2013. DOI: `10.1111/cgf.12110`.

[75] M. Haidacher, D. Patel, S. Bruckner, A. Kanitsar, and M. E. Gröller, "Volume visualization based on statistical transfer-function spaces", in *Proceedings of PacificVis*, 2010, pp. 17–24. DOI: `10.1109/PACIFICVIS.2010.5429615`.

[76] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval", *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000. DOI: `10.1023/A:1026543900054`.

[77] M. Harrower and C. A. Brewer, "ColorBrewer.org: An online tool for selecting colour schemes for maps", *The Cartographic Journal*, vol. 40, no. 1, pp. 27–37, 2003. DOI: `10.1179/000870403235002042`.

[78] T. Schultz, H. Theisel, and H.-P. Seidel, "Segmentation of DT-MRI anisotropy isosurfaces", in *Proceedings of EuroVis*, 2007, pp. 187–194. DOI: `10.2312/VisSym/EuroVis07/187-194`.

[79] T. Ropinski, I. Viola, M. Biermann, H. Hauser, and K. Hinrichs, "Multimodal visualization with interactive closeups", in *Proceedings EGUK (Theory and Practice of Computer Graphics)*, 2009, pp. 17–24. DOI: `10.2312/LocalChapterEvents/TPCG/TPCG09/017-024`.

[80] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls", *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, 1996. DOI: `10.1145/235815.235821`.

[81] D. Sehnal. (2012). MIConvexHull. Published: 2012-Nov-21. Accessed: 2014-Mar-16, [Online]. Available: `http://miconvexhull.codeplex.com/`.

[82] K. M. Hanson, "Noise and contrast discrimination in computed tomography", *Radiology of the Skull and Brain*, vol. 5, no. 1, pp. 3941–3955, 1981.

[83] P. Gravel, G. Beaudoin, and J. A. De Guise, "A method for modeling noise in medical images", *IEEE Transactions on Medical Imaging*, vol. 23, no. 10, pp. 1221–1232, 2004. DOI: `10.1109/TMI.2004.832656`.

[84] V. Kaynig, A. Vazquez-Reina, S. Knowles-Barley, M. Roberts, T. R. Jones, N. Kasthuri, E. Miller, J. Lichtman, and H. Pfister, "Large-scale automatic reconstruction of neuronal processes from electron microscopy images", *Medical Image Analysis*, vol. 22, pp. 77–88, 1 2015. DOI: `10.1016/j.media.2015.02.001`.

[85] D. Haehn, S. Knowles-Barley, M. Roberts, J. Beyer, N. Kasthuri, J. W. Lichtman, and H. Pfister, "Design and evaluation of interactive proofreading tools for connectomics", *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2466–2475, 2014. DOI: 10.1109/TVCG.2014.2346371.

[86] M. Šrámek and L. Dimitrov, "Segmentation of tomographic data by hierarchical watershed transform", *Journal of Medical Informatics and Technologies*, vol. 3, pp. 161–169, 2003.

[87] J. Brooke, "SUS: A "quick and dirty"usability scale", in *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester, and A. L. McClelland, Eds., London: Taylor and Francis, 1996, ch. 21, pp. 189–194.

[88] A. Karimov, G. Mistelbauer, T. Auzinger, and E. Gröller, "Statistics-driven localization of dissimilarities in data", Institute of Computer Graphics and Algorithms, Technische Universität Wien, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, Tech. Rep. 1, 2016, human contact: technical-report@cg.tuwien.ac.at. [Online]. Available: https://www.cg.tuwien.ac.at/research/publications/2016/karimov-2016-SD/.

[89] J. M. Kniss, R. Van Uitert, A. Stephens, G. S. Li, T. Tasdizen, and C. Hansen, "Statistically quantitative volume visualization", in *Proceedings of IEEE Visualization Conference*, vol. 6, 2005, p. 37. DOI: 10.1109/VISUAL.2005.1532807.

[90] T. Tasdizen, S. P. Awate, R. T. Whitaker, and N. L. Foster, "MRI tissue classification with neighborhood statistics: A nonparametric, entropy-minimizing approach", *Lecture Notes in Computer Science*, vol. 3750, pp. 517–525, 2005. DOI: 10.1007/11566489_64.

[91] C. Lundström, P. Ljung, and A. Ynnerman, "Local histograms for design of transfer functions in direct volume rendering", *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 1570–1579, 2006. DOI: 10.1109/TVCG.2006.100.

[92] C. Heinzl, J. Kastner, T. Möller, and E. Gröller, "Statistical analysis of multi-material components using Dual Energy CT", in *Proceedings of Vision, Modeling, and Visualization*, 2008, p. 179.

[93] C. R. Johnson and J. Huang, "Distribution-driven visualization of volume data", *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 5, pp. 734–746, 2009. DOI: 10.1109/TVCG.2009.25.

[94] A. Saad, G. Hamarneh, and T. Möller, "Exploration and visualization of segmentation uncertainty using shape and appearance prior information", *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1366–1375, 2010. DOI: 10.1109/TVCG.2010.152.

[95] C. M. Jarque and A. K. Bera, "A test for normality of observations and regression residuals", *International Statistical Review*, vol. 55, no. 2, pp. 163–172, 1987. DOI: 10.2307/1403192.

[96] B. L. Welch, "The generalization of 'Student's' problem when several different population variances are involved", *Biometrika*, vol. 34, no. 1, pp. 28–35, 1947. DOI: 10.2307/2332510.

[97] H. Hochheiser and B. Shneiderman, "Dynamic query tools for time series data sets: Timebox widgets for interactive exploration", *Information Visualization*, vol. 3, no. 1, pp. 1–18, 2004.

[98] P. Buono, A. Aris, C. Plaisant, A. Khella, and B. Shneiderman, "Interactive pattern search in time series", *Proceedings of SPIE*, vol. 5669, no. 1, pp. 175–186, 2005. DOI: 10.1117/12.587537.

[99] P. Buono, C. Plaisant, A. Simeone, A. Aris, B. Shneiderman, G. Shmueli, and W. Jank, "Similarity-based forecasting with simultaneous previews: A river plot interface for time series forecasting", in *Proceedings of International Conference on Information Visualisation*, 2007, pp. 191–196. DOI: 10.1109/IV.2007.101.

[100] M. Bögl, W. Aigner, P. Filzmoser, T. Lammarsch, S. Miksch, and A. Rind, "Visual analytics for model selection in time series analysis", *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2237–46, 2013. DOI: 10.1109/TVCG.2013.222.

[101] M. A. Stephens, "EDF statistics for Goodness-of-Fit and some comparisons", *Journal of the American Statistical Association*, vol. 69, no. 347, pp. 730–737, 1974. DOI: 10.1080/01621459.1974.10480196.

[102] T. Duong, B. Goud, and K. Schauer, "Closed-form density-based framework for automatic detection of cellular morphology changes", in *Proceedings of the National Academy of Sciences*, vol. 109, 2012, pp. 8382–8387. DOI: 10.1073/pnas.1117796109.

[103] A. N. Kolmogorov, "Sulla determinazione empirica di una legge di distribuzione", *Giornale dell'Istituto Italiano degli Attuari*, vol. 4, pp. 83–91, 1933.

[104] N. Smirnov, "On the estimation of the discrepancy between empirical curves of distribution for two independent samples", *Rec. Math. [Mat. Sbornik] N.S.*, vol. 6, no. 48, pp. 3–26, 1939.

[105] T. B. Arnold and J. W. Emerson, "Nonparametric Goodness-of-Fit tests for discrete null distributions", *The R Journal*, vol. 3, no. 2, pp. 34–39, 2011.

[106] J. M. Dufour and A. Farhat. (2001). Exact nonparametric two-sample homogeneity tests for possibly discrete distributions. Accessed: 2015-Oct-25, [Online]. Available: http://hdl.handle.net/1866/362.

[107] J. T. Praestgaard, "Permutation and bootstrap Kolmogorov-Smirnov tests for the equality of two distributions", *Scandinavian Journal of Statistics*, vol. 22, no. 3, pp. 305–322, 1995.

[108] B. F. J. Manly, *Randomization, Bootstrap, and Monte Carlo Methods in Biology*, Second. Chapman and Hall, London, 1997.

[109] D. A. Jackson and K. M. Somers, "Are probability estimates from the permutation model of Mantel's test stable?", *Canadian Journal of Zoology*, vol. 67, no. 3, pp. 766–769, 1989. DOI: 10.1139/z89-108.

[110] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Frannery, *Numerical Recipes in FORTRAN 77; The Art of Scientific Computing*, Second. The Press Syndicate of the University of Cambridge, 1993, ISBN: 052143064X. DOI: 10.1016/0378-4754(93) 90043-T.

[111] T. W. Anderson, "On the distribution of the two-sample Cramér-von Mises criterion", *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1148–1159, 1962. DOI: 10.1214/aoms/1177704477.

[112] A. N. Pettitt, "A two-sample Anderson-Darling rank statistic", *Biometrika*, vol. 63, no. 1, pp. 161–168, 1976. DOI: 10.2307/2335097.

[113] E. D. Feigelson and G. J. Babu. (2012). Beware the Kolmogorov-Smirnov test! Accessed: 2015-Oct-25, [Online]. Available: https://asaip.psu.edu/Articles/ beware-the-kolmogorov-smirnov-test.

[114] T. Nichols, M. Brett, J. Andersson, T. Wager, and J. B. Poline, "Valid conjunction inference with the minimum statistic", *NeuroImage*, vol. 25, no. 3, pp. 653–660, 2005. DOI: 10.1016/j.neuroimage.2004.12.005.

[115] S. A. Stouffer, E. A. Suchman, L. C. DeVinney, S. A. Star, and R. M. J. Williams, *The American Soldier: Adjustment during Army Life*. Princeton University Press, Princeton, 1949, vol. 1.

[116] T. Liptak, "On the combination of independent tests", *Magyar Tud. Akad. Mat. Kutato Int. Kozl.*, vol. 3, pp. 171–197, 1958.

[117] F. Mosteller and R. R. Bush, "Selected quantitative techniques", in *Handbook of Social Psychology*, vol. 1, Addison-Wesley, Cambridge, 1954, pp. 289–334.

[118] M. C. Whitlock, "Combining probability from independent tests: The weighted Z-method is superior to Fisher's approach", *Journal of Evolutionary Biology*, vol. 18, no. 5, pp. 1368–1373, 2005. DOI: 10.1111/j.1420-9101.2005.00917.x.

[119] A. M. Winkler, S. M. Smith, and T. E. Nichols, *Non-parametric combination for analyses of multi-modal imaging*, Poster at the Organization for Human Brain Mapping in Seattle, 16-20 June, 2013.

[120] E. S. Edgington, "An additive method for combining probability values from independent experiments", *The Journal of Psychology*, vol. 80, pp. 31–363, 1972. DOI: 10.1080/ 00223980.1972.9924813.

[121] R. A. Fisher, *Statistical Methods for Research Workers*. Oliver and Boyd (Edinburgh), 1925, ISBN: 0-05-002170-2.

[122] K. J. Friston, A. P. Holmes, C. J. Price, C. Büchel, and K. J. Worsley, "Multisubject fMRI studies and conjunction analyses", *NeuroImage*, vol. 10, no. 4, pp. 385–396, 1999. DOI: 10.1006/nimg.1999.0484.

[123] L. H. C. Tippett, *The Methods of Statistics*, ser. Wiley publications in statistics. Williams and Norgate, 1952.

[124] R. D. Cousins, "Annotated bibliography of some papers on combining significances or p-values", pp. 1–15, 2008. arXiv: 0705.2209. [Online]. Available: http://arxiv.org/abs/0705.2209.

[125] K. Moreland, "Diverging color maps for scientific visualization", in *Proceedings of 5th International Symposium on Visual Computing*, 2009, pp. 92–103. DOI: 10.1007/978-3-642-10520-3_9.

[126] W. J. Cody, "Algorithm 715: SPECFUN - a portable FORTRAN package of special function routines and test drivers", *ACM Transactions on Mathematical Software*, vol. 19, no. 1, pp. 22–32, 1993. DOI: 10.1145/151271.151273.

[127] M. J. Wichura, "Algorithm AS 241: The percentage points of the normal distribution", *Applied Statistics*, vol. 37, no. 3, pp. 477–484, 1988. DOI: 10.2307/2347330.

[128] Consortium for Open Medical Image Computing. (2016). Grand challenges in biomedical image analysis. Accessed: 2016-Apr-29, [Online]. Available: http://grand-challenge.org/all_challenges/.

Curriculum Vitae

Contact Information

Name	Alexey Karimov
Address	Molkereistrasse 1 /2 /34, 1020 Vienna, Austria
E-mail	xkarimov@hotmail.com
	xkarimov@gmail.com
Skype	xcoderz
Phone	0043 680 225 40 28

Personal Details

Date of birth	18th of October, 1987
Place of birth	Ufa, Russian Federation
Citizenship	Russian Federation
Language skills	Russian (native), English (fluent)

Education

Sep. 1995 - May 2003	Secondary School #74, Ufa, Russian Federation **Secondary education**
Sep. 2003 - May 2005	Lyceum #153, Ufa, Russian Federation **Secondary education**
Sep. 2005 - May 2010	Ufa State Aviation Technical University, Ufa, Russian Federation **Master's studies in Computer Science** Graduated *summa cum laude* Faculty: Computation Mathematics and Cybernetics Specialty: Mathematical Models, Software and Information Systems Administration Thesis: Mathematical model, algorithms and software of system for Computed Tomography images processing

Oct. 2010 - May 2013	Ufa State Aviation Technical University, Ufa, Russian Federation **Doctoral studies in Computer Science** Discontinued Faculty: Computation Mathematics and Cybernetics Specialty: System Analysis, Data Management and Processing
Jul. 2011 - Dec. 2011	Technische Universität Wien, Vienna, Austria **Erasmus Mundus internship** Faculty of Informatics Institute of Computer Graphics and Algorithms
Apr. 2013 - present time	Technische Universität Wien, Vienna, Austria **Doctoral studies in Computer Science** Faculty of Informatics Institute of Computer Graphics and Algorithms Dissertation: Guided Interactive Volume Editing in Medicine Supervisor: Ao.Univ.Prof. Univ.-Doz. Dipl.-Ing. Dr. techn. Eduard Gröller

Employment History

Mar. 2012 - present time	Technische Universität Wien, Vienna, Austria **Project assistant** Natural Fetoscopic Rendering Collaboration with GE Healthcare Austria Gmbh

Reviewing Activities

Journals	Computer Graphics International, Computer & Graphics, Computerized Medical Imaging and Graphics
Conferences	IEEE Visualization, EuroVis, IEEE Pacific Visualization

Journal Publications

Alexey Karimov, Gabriel Mistelbauer, Johanna Schmidt, Peter Mindek, Elisabeth Schmidt, Timur Sharipov, Stefan Bruckner and Eduard Gröller. ViviSection: Skeleton-based Volume Editing. *Computer Graphics Forum* 32(3), pages 461–470, 2013.

Alexey Karimov, Gabriel Mistelbauer, Thomas Auzinger and Stefan Bruckner. Guided Volume Editing based on Histogram Dissimilarity. *Computer Graphics Forum* 34(3), pages 91–100, 2015.

Alexey Karimov, Gabriel Mistelbauer, Thomas Auzinger and Eduard Gröller. Statistics-Driven Localization of Dissimilarities in Data. *In submission*.

Manuela Waldner, **Alexey Karimov** and Eduard Gröller. Content-Preserving Highlighting. *In submission*.