Voxel-based Surface Area Estimation: From Theory to Practice

G. Windreich **N. Kiryati**[*]

Department of Electrical Engineering–Systems
Faculty of Engineering
Tel Aviv University
Tel Aviv 69978, Israel

G. Lohmann

Max-Planck-Institute of Cognitive Neuroscience
Stephanstr. 1a
04103 Leipzig, Germany

Abstract

Consider a complex, highly convoluted three dimensional object that has been digitized and is available as a set of voxels. How can one estimate the (original, continuous) area of a region of interest on the surface of the object? The problem appears in the analysis of segmented MRI brain data and in other three dimensional imaging applications. Several difficulties arise. First, due to the complexity of the surface and its foldings, the region of interest and its intended boundary can be concealed and are therefore difficult to delineate. Second, the correct surface topology on intricate voxel sets may not be obvious. Third, the surface area of a digital voxel world is generally very different than the area of the underlying continuous surface. These problems can be partly circumvented by transforming the voxel data to a polyhedral surface representation. Our challenge is to accomplish the task while maintaining the original voxel representation. Estimators for the continuous surface area of digital objects have been available for some time. However, the known methods are limited to fairly smooth and "well behaved" surfaces. This research bridges the gap between the available surface area estimation theory, that applies to idealized settings, and the reality of MRI brain data. Surface connectivity ambiguities are alleviated by considering the object/background boundary voxel faces rather than the border voxels themselves. The region of interest on the surface is delimited by growing geodesic paths between user-provided anchor points. Surface estimation is extended to admit surfaces with higher curvature than previously considered. Performance evaluation results are provided, and operation on MRI brain data is demonstrated.

Key words: surface area estimation, digital geometry, voxel objects, morphometric measurements, segmented white matter

[*] Corresponding author. Phone: +972 3 640 7767. Fax: +972 3 640 7095. E-mail: nk@eng.tau.ac.il.

1 Introduction

Consider a three dimensional object that has been digitized and is given as a set of voxels. How can one delimit a region of interest on the surface of the object and estimate its area? This problem may arise in various 3D image analysis contexts, e.g. in reverse engineering. It is most urgent, however, in medical imaging, especially in the analysis of the cortex in MR images. The cortex is the thin outermost layer of grey matter in the brain; cortical surface area is likely to be related to functional capacities [1].

Following 3D acquisition, segmentation is the first image analysis step in most medical and industrial applications. Due to the highly convoluted structure of the brain, topologically-correct brain segmentation is a significant challenge and the topic of extensive research [1, 2, 3]. The segmented brain, usually in the form of a tagged set of voxels, is the basis for further processing and analysis.

Given the segmented voxel set, surface area measurement can be divided into three essential steps. First, the user traces the boundary of the region of interest on the surface. Second, the region surrounded by the boundary is identified. Third, the area of the region is computed. None of these tasks is trivial.

Marking a boundary contour on a convoluted surface is not straightforward, because parts of the intended curve may not be visible. To overcome this limitation, the user should be able to select a sequence of key points, and have them connected automatically to form the boundary. This calls for an efficient algorithm for geodesic path generation, i.e., for finding shortest paths between points on a surface.

In the continuous world, Jordan's theorem ensures that a simple closed contour encloses a region and serves as its well defined boundary, separating the interior and exterior. In the discrete domain, identifying a region of interest by its outline is prone to paradoxes, since the discrete version of Jordan's theorem does not generally hold and different definitions of digital connectivity must be used for the region and its boundary [3].

Estimating the continuous area of a surface that is available only in digital form is fundamentally difficult. Different continuous surfaces, with different surface areas, may have the same digital representation. Furthermore, the voxel representation of smooth continuous surfaces is generally jagged, so the total area of exposed voxel faces is usually much greater than that of the original continuous surface.

Transforming the digital object and its surface from their original voxel representation to a triangle-based polyhedral representation, common in computer graphics and computer-aided design, has some advantages. In particular, an efficient algorithm for geodesic path generation on triangulated domains is available [4], facilitating key point based boundary generation. Moreover, on the triangulated domain the paths found are continuous, so the boundary outlines the region of interest in a well defined way. Two difficulties however arise. First, when using the marching cubes algorithm [5] to create the triangulated representation, topological ambiguities may occur and holes may be generated [6]. Second, the surface area estimate produced by summing up the area of the resulting triangles does not converge to the true surface area as the resolution increases [7]; this follows from the *locality* of the marching cubes algorithm [8].

Klette and Sun [8] suggest that surface area estimators that converge to the true surface area can be obtained by using a *global* polyhedrization method. Their hypothesis is supported by experimental results, though not yet by a formal proof. More efficient algorithms are required in order to make their method practical for large, high resolution data sets.

Our approach is different: the original voxel representation of the object is maintained and no polyhedrization is carried out. This considerably simplifies the overall system design and avoids potential difficulties, ambiguities and distortion that may arise due to the application of a polyhedrization process to a complex, convoluted surface.

A voxel-based surface area estimator was presented by Mullikin and Verbeek [9]. Extending the planar perimeter estimation methodology [10, 11], their estimator is designed to be unbiased and minimize the mean square error (MSE) for planes, and its operation is evaluated with spheres. The estimator of Mullikin and Verbeek [9] is at the core of the method presented here. However, as will be discussed, it cannot be directly applied to complex convoluted surfaces and various difficulties need to be addressed. The creation of a complete voxel-based surface area estimation method, applicable to surfaces as complex as that of the brain, is the focus of this research.

2 Delimiting the Region of Interest

2.1 Border vs. Boundary

Given a three dimensional object that is represented as a set of voxels, one can easily identify the voxels that are 6-connected to the background and view them as the border of the object. The border set can be represented as a graph; once the user defines keypoints on the border, they can be automatically connected using the algorithm of [12] to obtain a closed contour that encloses the region of interest.

The border-based approach to delimiting the region of interest may be suitable for smooth, thick, well behaved objects. It is, however, inadequate for complex convoluted surfaces such as the cortex. Consider for example the object detail shown in Fig. 1 (left). The voxels marked ‘1’ and ‘2’ both belong to the border set. When connecting them (as part of the contour generation process) using an algorithm like [12], the connection (solid) will not follow its intended path (dashed). A similar phenomenon is shown in Fig. 1 (right). Once a closed contour within the border set has been created, Fig. 2 demonstrates that the contour does not properly enclose the region of interest. A region grown from the seed voxel (black) within the outline will include not only the intended region of interest (darker voxels in the top layer) but also other voxels in the border set to which they are connected (bottom layer). Some of these difficulties can be alleviated by defining certain connections between voxels as illegal, by deleting (or adding) critical voxels or by increasing the spatial resolution of the digitization process. However, these are ad-hoc patches that do not remove the fundamental weaknesses of the border-based approach.

The difficulties associated with using the border voxel set can be eliminated by keeping track of the *boundary* of the object, the set of voxel *faces* that separates the object from the background [13]. This is easily verified for the pathological cases shown in Figs. 1 and 2. Note that boundary faces are adjacent if they share an edge; this implies 18-connectivity of the object itself. Given a 26-connected object, it might be separated into a number of 18-connected parts. Their surface areas are estimated separately and added up.

2.2 Data Structures

The tasks associated with delimiting the region of interest include the detection of the border-set and the boundary, selection of key-faces on the boundary, creation of a chain of faces that connects the key-faces and encloses the region of interest on the surface, seed selection within the region of interest, growing the region of interest on the boundary and associating it with the border-set. Maintaining the border-set representation is crucial, since the surface area estimation algorithm, based on [9], operates on the border-set.

The data structure used to represent the boundary must satisfy several requirements. To allow the selection of key-faces, direct or semi-direct access to each boundary face should be possible. Chain generation and region growing call for a graph-like representation, where each face is linked (by its four edges) to its four spatially adjacent faces. Serial traversal of the selected boundary elements is also necessary.

Each boundary face can be uniquely defined by the 4-tuple (x, y, z, F), where, (x, y, z) are the coordinates of the surface voxel to which the face belongs, and F is the *face direction*, which indicates at one of the six voxel faces [13]. Straightforward representation of the boundary as a three dimensional $X \times Y \times Z$ array (where X,Y and Z are the 3D image dimensions) would be wasteful, since the memory needed is proportional to the volume of the image rather than to the surface area of the object. We can meet the requirements by describing boundary faces as independent objects (containing the necessary x, y, z, F information) and using two separate data structures to provide both the graph representation and the direct access mechanisms.

As the non-directed graph representation of the boundary has a maximum degree of four, it can be implemented by allocating an array of four pointers to every face object, each pointing at one of the face's neighbors. At first glance, it seems that we could have used a more efficient representation of the boundary [13] as a directed graph with indegree and outdegree of two, thus requiring only *two* pointers for each face. However, in order to find shortest paths (connecting key-faces) on the boundary, each one of the *four* neighbors of each face has to be checked. Also, as later explained, the directed graph representation is not suitable for growing a region of interest that does not form a closed surface. The second data structure, that meets the requirement of semi-direct access to boundary faces while being compact, is a hash-table [3]. The memory required is proportional to the number of face objects, i.e., to the surface area. For the voxel border set, we need only serial traversal and random access. Therefore, the border-set can also be represented using a hash table, where each surface voxel is represented by a 3-tuple (X, Y, Z).

2.3 Algorithms

2.3.1 Border and boundary Detection

The straightforward approach to simultaneous detection of the border-set and the boundary is to visit each voxel in the 3-D image and determine whether it is 6-connected to a background voxel. Each surface voxel detected is inserted to the border hash table and each boundary face to the boundary hash table. Insertion requires $O(1)$ time on average, hence the time required for border and boundary detection is proportional to the number of image voxels. In practice, for the brain images used in this research, this operation took only 7 seconds on a 350MHz PC. Thus, sophisticated alternative algorithms were not needed.

2.3.2 Constructing the boundary adjacency graph

Following the detection of the border and the boundary, the boundary faces adjacency graph is constructed. For each boundary face, adjacent boundary faces are determined. A simple closed surface can be represented as a directed graph with indegree and outdegree two [13]. Here, we extract a region of interest on the closed surface of an object, i.e., a surface patch. The directed graph representation is not suitable for the surface patch: there will be nodes in the graph with indegree (or outdegree) smaller than two. For example, consider an object that consists of a single voxel. Suppose the region of interest contains five faces out of the six that form the surface of the voxel. Only one of these five faces has four neighbors (two of them are adjacent faces in the sense of [13], that share the outgoing edges with the source face. The other two share the incoming edges). Each of the other four faces has only *three* edges shared with other faces. Clearly, this simple region of interest cannot be described using the directed graph representation. Thus, to find the boundary faces adjacent to each face, we modify the method of [13] to allow four adjacent faces for each boundary face, one for each of its edges.

2.3.3 Connecting key-points on the boundary

Kiryati and Székely [12] described an efficient algorithm for finding shortest paths on voxel surfaces represented as graphs. Here, given the boundary graph representation, a similar algorithm can be devised to find reasonably short paths between the keypoints defining the region of interest on the surface. As in [12], the sparsity of the boundary adjacency graph allows very efficient search. Unlike [12], where different spatial adjacency relations (link types) between voxels induce different weights for arcs in the surface graph, here all four arcs connecting a boundary face to adjacent faces are equally weighted. The operation of the algorithm is demonstrated in Fig. 3. It requires $O(N \log N)$ time, where N is the number of boundary faces.

2.3.4 Growing the region of interest on the object boundary

In the boundary adjacency graph, the degree of each node is between one and four. Like [13], we use a breadth first search algorithm for graph traversal that begins at an arbitrary node. Unlike [13], we have the list of boundary faces in memory so we don't have to detect the boundary, but only to mark the nodes (boundary faces) that are within the region of interest. To allow the search to stop at the borderline of the region of interest, all the nodes representing the outline of the region of interest are marked as they are generated by the shortest-path algorithm. Thus, they are already marked when surface growing starts.

3 Surface Area Estimation

3.1 Background

Suppose that a continuous planar curve has been quantized and is given in digital form, e.g., as a chain code. Can the perimeter of the original continuous curve be determined? Since information is lost in the digitization process, exact recovery of the perimeter is not possible. However, if the curve is smooth except at a small number of corners, and the quantization is sufficiently fine, such that the curve can be considered nearly straight within most small pixel-neighborhoods, meaningful perimeter estimation is possible.

The 2-D perimeter estimation problem has received considerable attention [10, 11]. A standard approach is to classify the chain code elements according to certain criteria, and estimate the length as a weighted sum of the number of chain elements in each class. Once the classification criteria are selected, the corresponding weights are designed to yield unbiased and in some sense optimal estimates of the length of straight lines. Unbiasedness for straight lines implies that, when applied to general curves with varying tangent orientations, local length estimation errors are likely to cancel out, and excellent total perimeter estimation can be expected.

Following the progress in 3-D imaging, the 2-D perimeter estimation theory has been expanded to 3-D curves [12, 14, 15, 16] and plays an important role in modern 3-D shape analysis techniques [17].

3.2 The Surface Area Estimator of Mullikin & Verbeek [9]

Mullikin and Verbeek [9] extended the theory of 2-D perimeter estimation to 3-D surface area estimation. Their algorithm begins by detecting all surface voxels, i.e., object voxels that are 6-connected to background voxels. The surface voxels are classified into nine classes, and the surface area is estimated as a linear combination of the class membership values $\{N_i\}$:

$$\hat{S} = \sum_{i=1}^{9} W_i N_i$$

Each surface voxel is classified according to the number and configuration of its faces that are exposed to the background. Up to rotation and mirroring, there are exactly nine unique voxel classes (Fig. 4), denoted S_{1-9}. Only voxels of types S_{1-3} appear in digital planes. Voxel types S_{4-6} are found in curved border regions. Voxel types S_{7-9} exist in extreme situations, where the object is a plane, line or point respectively. An alternative voxel classification has recently been suggested in [18].

Having defined the voxel classification scheme, Mullikin and Verbeek determined the weights W_{1-3} associated with voxels in classes S_{1-3}, to make the surface area estimate *unbiased* for random plane orientations and to minimize the mean square error. These weights are $W_1 \approx 0.894$, $W_2 \approx 1.3409$ and $W_3 \approx 1.5879$; the coefficient of variation ($CV = \sigma/\mu$) for planes is 2.33%. Clearly, an unbiased estimator for planes will have very small errors when applied to curved surfaces, where local estimation errors, obtained at differently oriented patches, essentially cancel out.

This design methodology does not determine the weights W_{4-9} associated with voxel classes S_{4-9}. Adopting weights obtained using the spatial grid method [19], Mullikin and Verbeek set $W_4 = 2$, $W_5 = 8/3$ and $W_6 = 10/3$. Studying digitized spheres of radii varying between 5 and 50, it was observed that S_4 voxels do not appear, S_6 voxels are very rare, and S_5 voxels have some presence, that rapidly decreases with radius: from 5.5% of the surface voxels at $r = 5$ to 0.82% at $r = 15$. No weights were assigned by Mullikin and Verbeek to voxel types S_{7-9}. The experimental performance evaluation with spheres also revealed some bias related to the radius, that can be largely alleviated by averaging the surface area of the object with that of the background.

The surface area estimator of Mullikin and Verbeek is local. While it does not exhibit multigrid convergence, it operates directly on voxels, is easy to implement, fast to compute and achieves very reasonable accuracy. Note that multigrid convergence is related to surface area estimation accuracy with resolution approaching infinity (and surface curvature approaching zero). This is not the case in MR brain images, where resolution is limited and surface curvature is high.

3.3 Voxel Types in Brain Surfaces

When considering the application of the surface area estimator of Mullikin and Verbeek to brain surface measurement, several difficulties arise. Some are related to the definition of the surface and the extraction of a region of interest on the surface; these were considered in section 2. Here we focus on the presence of voxel types S_{4-9} in real brain data.

Table 1 shows the frequency of all nine surface voxel types in a $160 \times 200 \times 160$ segmented white matter MR brain image (in the grey-white matter interface). It is seen that all nine voxel classes are represented, and that voxel types S_{4-9} constitute 3.23% of the 187567 surface voxels. Fig. 5 shows voxels of types S_{7-9} as they appear within the segmented MR brain image. These classes contain 1.35% of the surface voxels in this case.

The presence of voxels types S_{4-9} in brain data necessitates the assignment of weights to these classes. Yet, their fairly low frequency means that the overall surface estimation accuracy is not too sensitive to the weights selected. As discussed above, weights for classes S_{4-6} were already proposed in [9]. For voxel classes S_{7-9}, we suggest the following.

For S_7, with two opposite faces exposed to the background, we take the weight to be twice the weight of voxel type S_1 (that has only one face exposed to the background), i.e., $W_7 = 1.79$. A voxel in S_8 has two pairs of adjacent faces exposed to the background; we can take its weight as twice that of voxel type S_2 (that has only one pair of adjacent faces exposed to the background), i.e., 2.68. Alternatively, one can argue that the weight should be 4/5 of the weight of S_6 (that has 5 faces exposed to the background). This gives an almost identical weight of 2.67. Thus, for all practical purposes we can take $W_8 = 2.68$. As to S_9, with all 6 faces exposed to the background, the weight can be taken to be the sum of the weights of S_6 and S_1 (4.23), or twice that of S_4 (4) or the sum of the weights of S_5 and S_2 (4.01). The difference between these values is insignificant considering the low frequency of these voxels. We take $W_9 = 4.08$.

Consider a flat, thin object consisting of a single layer of S_7 voxels. Suppose that the boundary region of interest is on one side of the object. Each of the S_7 voxels has two faces exposed to the background, but only one of them belongs to the region of interest! In this case only half of the weight W_7 should contribute to the surface area estimate. Generally, for a voxel with P faces exposed to the background, of which p faces belong to the region of interest on the boundary, we take

$$W_i^p = \frac{p}{P} \cdot W_i$$

where W_i is the voxel class weight. In most cases $p = P$, so $W_i^p = W_i$.

3.4 Performance Evaluation

Mullikin and Verbeek [9] evaluated the performance of their surface area estimator using simulated spheres and cylinders. Here we report on our simulation results, with synthetic spheres and ellipsoids. Small spheres represent objects with high surface curvature, that deviate greatly from the planar surface model used in the design of the estimator. Large spheres, with their uniformly distributed surface normals, can demonstrate the unbiasedness of the estimator. Unlike spheres, surface normal directions on ellipsoids are not uniformly distributed. Testing the surface area estimator on ellipsoids, suggested in [7], indicates the sensitivity of the estimation error to nonuniformity of the normal directions distribution. Note that planar objects, having a single normal direction (or two opposite directions if both sides are considered), are generally the worst case for the estimator: the coefficient of variation (the standard deviation divided by their mean) is 2.33% for randomly oriented planes.

Fig. 6 shows the relative mean estimation error (percent) in estimating the surface area of spheres (average of object and background surface areas), as a function of sphere radius. The left and right graphs focus on small and large radii respectively. Each point in the graphs is based on 50 spheres, whose center points are uniformly distributed within the unit voxel. It is seen that the relative mean estimation error is less than 1% even for spheres of radius 2, that it rapidly decreases as the radius is increased, and is practically zero for radii larger than 10. The coefficient of variation of these measurements is presented in Fig. 7. It rapidly decreases with the sphere radius, from about 4% with sphere radius 2, through 0.5% with radius 5 to negligibly small values at larger radii. These results are similar to those obtained with spheres in [9]; they demonstrate the outstanding accuracy of the estimator even when the surface curvature is very high. Note that surface curvature radii between 2 and 5 are common in the segmented MR white matter brain data used in this research.

Fig. 8 shows an ellipsoid centered at the origin, with semi-main axes a, b and c that coincide with the coordinate system axes. Each data point in Figs. 9 and 10 corresponds to surface area estimation of 50 such ellipsoids, with their center points uniformly distributed within the unit voxel[1]. The dots in Fig. 9 refer to the ellipsoid family $(a, b = 26, c = 25)$, and shows the relative mean surface area estimation as a function of a. The x signs refer to the ellipsoid family $(a, b = 51, c = 50)$. As expected, the error is almost zero for nearly spherical ellipses; it slowly grows as a increases and the ellipses become elongated. The respective coefficients of variation are shown in Fig. 10; they are very small.

4 Application to Brain Data

The methods presented in this paper have been implemented as a C++ program named *Surf3D*, for Unix platforms. *Surf3D* receives as input 3D binary images in the *Vista* format (Vista is a C library providing an environment for image handling [21]. A *Vista* input image is in fact a 3D binary array.) *Surf3D* visualizes the 3D image using the OpenGL imaging library. It allows the user to interact with the surface using the mouse, marking a set of key points that indicate the region of interest on the surface of the viewed object. The keypoints are connected automatically to create the surrounding contour. From a seed point selected by the user, *Surf3D* grows the region of interest and estimates its area. Batch processing of prespecified surface regions is also supported.

A typical work session with *Surf3D* is illustrated in Figs. 11-13. This session was per-

[1] The true ellipsoid surface area is computed via the method of [20].

formed using a 350MHz P-II PC, running Linux. The input data was a $160 \times 200 \times 160$ segmented white matter MR brain image, with 558363 object voxels, of which 156825 were surface voxels, having 300130 boundary faces.

The session begins by loading a Vista image, containing segmented white matter MR brain data. Within 8 seconds, the border and the boundary of the object are detected and displayed. The user defines the contour of the region of interest by selecting keypoints on the surface. Two mouse clicks define the first segment of this contour (Fig. 11). The two points are connected automatically within 2 seconds. Additional steps in this process are shown in Figs. 12a-c. The user now notifies the program to close the contour. The program connects the last key point selected by the user to first one, see Fig. 12d. The user clicks on a surface point inside within the region of interest, from which the program grows the region (Fig. 13) and estimates its surface area. The elapsed time from the selection of the seed point to the area estimation output is less than 1 second. The estimated surface area in this case is 2421 (boundary face units).

5 Conclusions

This research provides a fast, accurate and convenient scheme for estimating the surface area of regions of interest on the surface of digital objects. The input is a 3D binary digital image, i.e., a set of voxels. The voxel representation is maintained and no triangulation is carried out. The suggested technique bridges the gap between the theoretical results of Mullikin and Verbeek [9] and the reality of complex medical data. In particular, the method is well suited for highly convoluted surfaces. The accuracy is verified using synthetic surfaces: simulation results reported in [9] are corroborated, and augmented by results on ellipsoids. Successful operation is demonstrated on segmented white-matter MR brain data.

Acknowledgment

This research was supported by a grant from the G.I.F., the German-Israeli Foundation for Scientific Research and Development.

References

[1] X. Zeng, L.H. Staib, R.T. Schultz and J.S. Duncan, Segmentation and measurement of the cortex from 3-D MR images using coupled-surfaces propagation, IEEE Transactions on Medical Imaging 18 (1999) 927-937.

[2] H.E. Cline, W.E. Lorensen, R. Kikinis and F. Jolesz, Three dimensional segmentation of MR images of the head using probability and connectivity, J. Comp. Assisted Tomogr. 14 (1990) 1037-1045.

[3] G. Lohmann, Volumetric Image Analysis, Wiley, Chichester, UK & Teubner, Stuttgart, Germany, 1998.

[4] R. Kimmel and J. Sethian, Computing geodesics on manifolds, Proc. National Academy of Sciences 95 (1998) 8431-8435.

[5] W.E. Lorensen and H.E. Cline, Marching cubes: a high resolution 3D surface construction algorithm, ACM Computer Graphics 21 (1987) 163-169.

[6] R. Klette, F. Wu and S. Zhou, Multigrid convergence of surface approximations, Technical Report CITR-TR-25, Computer Science Department, University of Auckland, New Zealand, 1998.

[7] Y. Kenmochi and R. Klette, Surface area estimation for digital regular solids, Technical Report CITR-TR-62, Computer Science Department, University of Auckland, New Zealand, 2000.

[8] R. Klette and H.J. Sun, A global surface area estimation algorithm for digital regular solids, Technical Report CITR-TR-69, Computer Science Department, University of Auckland, New Zealand, 2000.

[9] J.C. Mullikin and P.W. Verbeek, Surface area estimation of digitized planes, Bioimaging 1 (1993) 6-16.

[10] L. Dorst and A.W.M. Smeulders, Length estimators for digitized contours, Computer Vision Graphics Image Processing 40 (1987) 311-333.

[11] J. Koplowitz and A.M. Bruckstein, Design of perimeter estimators for digitized planar shapes, IEEE Trans. Pattern Anal. Machine Intell. 11 (1989) 611-622.

[12] N. Kiryati and G. Székely, Estimating shortest paths and minimal distances on digitized three-dimensional surfaces, Pattern Recognition 26 (1993) 1623-1637.

[13] E. Artzy, G. Frieder and G.T. Herman, The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm, Computer Graphics and Image Processing 1 (1981) 1-24.

[14] N. Kiryati and O. Kübler, On chain code probabilities and length estimators for digitized three-dimensional curves, Pattern Recognition 28 (1995) 361-372.

[15] B.J.H. Verwer, Local distances for distance transformation in two and three dimensions, Pattern Recognition Letters 12 (1991) 671-682.

[16] A. Jonas and N. Kiryati, Length estimation in 3-D using cube quantization, Journal of Mathematical Imaging and Vision, 8 (1998) 215-238.

[17] R. Grossmann, N. Kiryati and R. Kimmel, Computational surface flattening: a voxel-based approach, IEEE Trans. Pattern Anal. Machine Intell. 24, (2002) 433-441.

[18] J. Lindblad and I. Nyström, Surface area estimation of digitized 3D objects using local computations, Proc. DGCI'2002, Lecture Notes in Computer Science 2301 (2002) 267-278.

[19] U. Hahn and K. Sandau, Precision of surface area estimation using spatial grids, Acta Stereologica 8 (1989) 425-430.

[20] The Surface Area of an Ellipsoid, http://functions.wolfram.com (Elliptic Integrals → EllipticE[z,m] → Theorems).

[21] A. Pope and D. Lowe, Vista: a software environment for computer vision research, http://www.cs.ubc.ca/nest/lci/vista/vista.html.

Biographical Sketches

Guy Windreich was born in 1969, in Ramat Gan, Israel. He received the B.Sc. degree in Electrical Engineering from the Technion–Israel Institute of Technology, Haifa, in 1992, and the M.Sc. degree in Electrical and Electronics Engineering from Tel Aviv University in 2002. He is currently with Orbotech Ltd., Yavne, Israel, in the field of image processing.

Nahum Kiryati received the B.Sc. degree in Electrical Engineering and the Post-B.A. degree in the Humanities from Tel Aviv University, Israel, in 1980 and 1986 respectively. He received the M.Sc. degree in Electrical Engineering in 1988 and the D.Sc. degree in 1991, both from the Technion, Israel Institute of Technology, Haifa, Israel. He was with the Image Science Laboratory, Institute for Communication Technology, ETH-Zurich, Switzerland, and with the Department of Electrical Engineering, Technion, Haifa, Israel. He is currently an associate professor in the Department of Electrical Engineering–Systems, Tel Aviv University. His research interests are in image analysis and computer vision.

Gabriele Lohmann received her diploma in mathematics and mathematical logic from the University of Münster in 1984, her doctorate in computer science from the Technical University of Munich in 1991, and her habilitation in applied computer science in 1999. She spent an academic year at Indiana University, Bloomington, IN supported by a Fulbright scholarship, and a six-month research stay at the Computer Vision Lab of the University of Massachusetts, Amherst, MA. From 1984 until 1991, she was a researcher at the German Aerospace Research Center working in the field of satellite remote sensing. She is currently a scientist at the Max-Planck-Institute of Cognitive Neuroscience where she leads a research group that specializes in mathematical methods of fMRI data analysis. Her research interests include computer vision, pattern recognition and neuroscience.

Figure Captions

Figure 1: In both examples, generation of the shortest path (solid) between border voxels '1' and '2' does not follow the intended contour (dashed), even though the path is constrained to border voxels.

Figure 2: A region grown from the seed voxel (black) within the contour will include not only the intended region of interest (darker voxels in the top layer) but also other voxels in the border set to which they are connected (bottom layer).

Figure 3: Finding the shortest path (white) between two boundary faces on a segmented white matter MR brain image.

Figure 4: The nine unique surface voxel classes (after [9]). Only voxels of types S_{1-3} appear in a planar surface.

Figure 5: Brain image details showing voxels of types S_{7-9} (dark). *(a,b)* Two images showing the same S_7 voxel (with two opposite faces are exposed to the background). In (b), the object detail is rotated by 180^o around the vertical axis with respect to (a). *(c)* An S_8 voxel, with two pairs of opposite faces exposed to the background. *(d)* Voxel of type S_9, with all its six faces exposed to the background.

Figure 6: The relative mean estimation error (percent) in estimating the surface area of spheres, as a function of sphere radius (with object-background averaging). *Left*: Small sphere radii. *Right*: Large sphere radii.

Figure 7: The coefficient of variation (percent) as a function of radius in estimating the surface area of spheres (with object-background averaging). *Left*: Small radii. *Right*: Large radii.

Figure 8: An ellipsoid with semi-main axes (a, b, c), aligned with the coordinate system.

Figure 9: Relative mean estimation error (percent) as a function of ellipsoid main semi-axis a (using object-background surface area averaging). The dots refer to the ellipsoid family $(a, 26, 25)$; the x's to $(a, 51, 50)$.

Figure 10: Coefficient of variation (percent) as a function of ellipsoid semi main-axis a (using object-background surface area averaging). The dots refer to the ellipsoid family $(a, 26, 25)$; the x's to $(a, 51, 50)$.

Figure 11: Grey/white matter interface in a segmented MR brain image. The first two key points entered by the user and the connecting path are shown (white).

Figure 12: Interactive definition of the region of interest on the surface. (a)-(c) Automatic connection of key points provided by the user. (d) Closure of the contour surrounding the region of interest.

Figure 13: The region of interest on the boundary surface, shown from two viewing points.

Table Caption

Table 1: The frequencies of surface voxels types in the grey-white matter interface of a segmented $160 \times 200 \times 160$ MR brain image.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	total
No. of voxels	65878	46532	37218	1151	2547	1383	468	923	725	187567
Frequency (%)	42.1	29.6	23.7	0.73	1.62	0.88	0.30	0.59	0.46	100

Table 1: The frequencies of surface voxels types in the grey-white matter interface of a segmented 160 × 200 × 160 MR brain image.

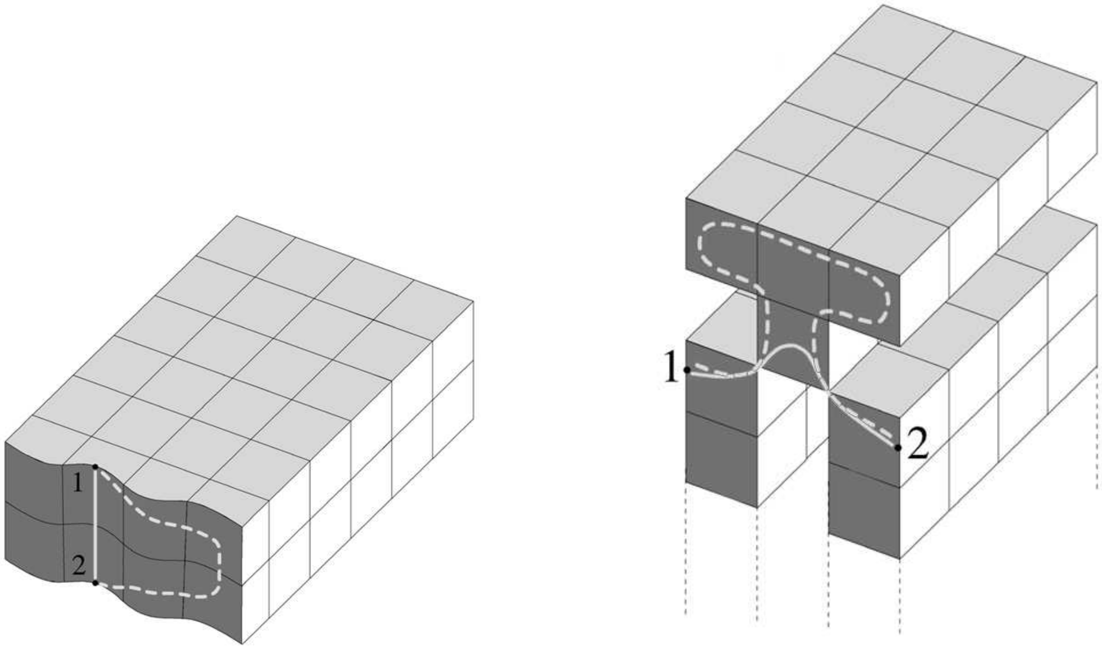

Figure 1: In both examples, generation of the shortest path (solid) between border voxels '1' and '2' does not follow the intended contour (dashed), even though the path is constrained to border voxels.

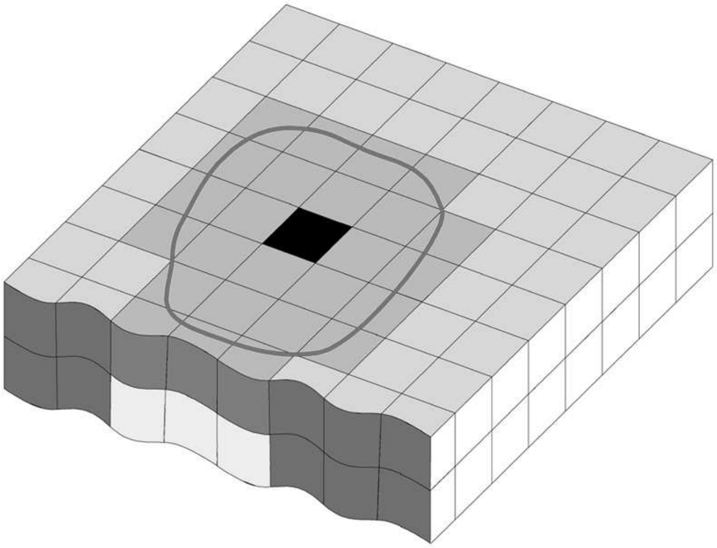

Figure 2: A region grown from the seed voxel (black) within the contour will include not only the intended region of interest (darker voxels in the top layer) but also other voxels in the border set to which they are connected (bottom layer).

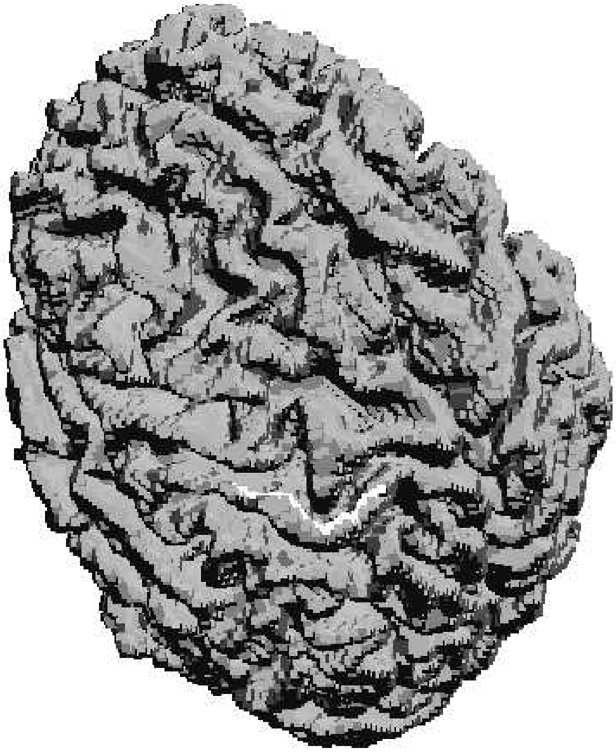

Figure 3: Finding the shortest path (white) between two boundary faces on a segmented white matter MR brain image.

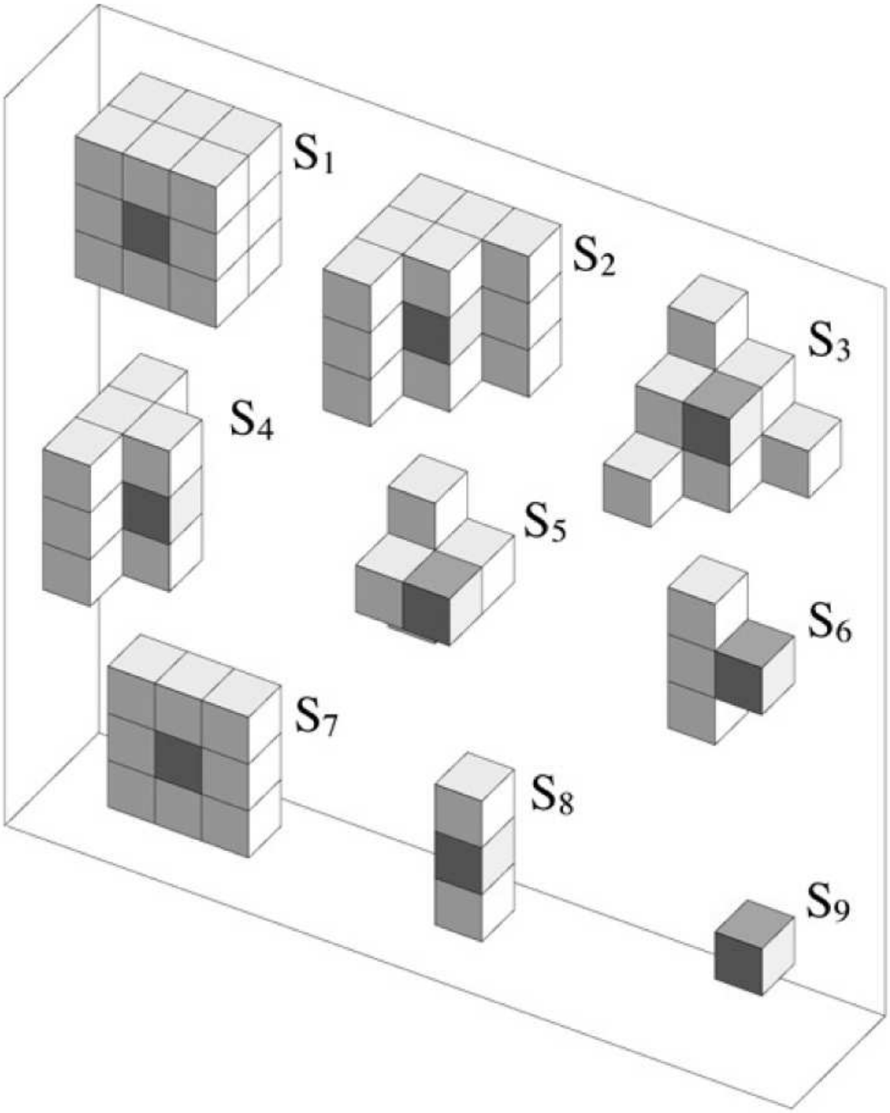

Figure 4: The nine unique surface voxel classes (after [9]). Only voxels of types S_{1-3} appear in a planar surface.

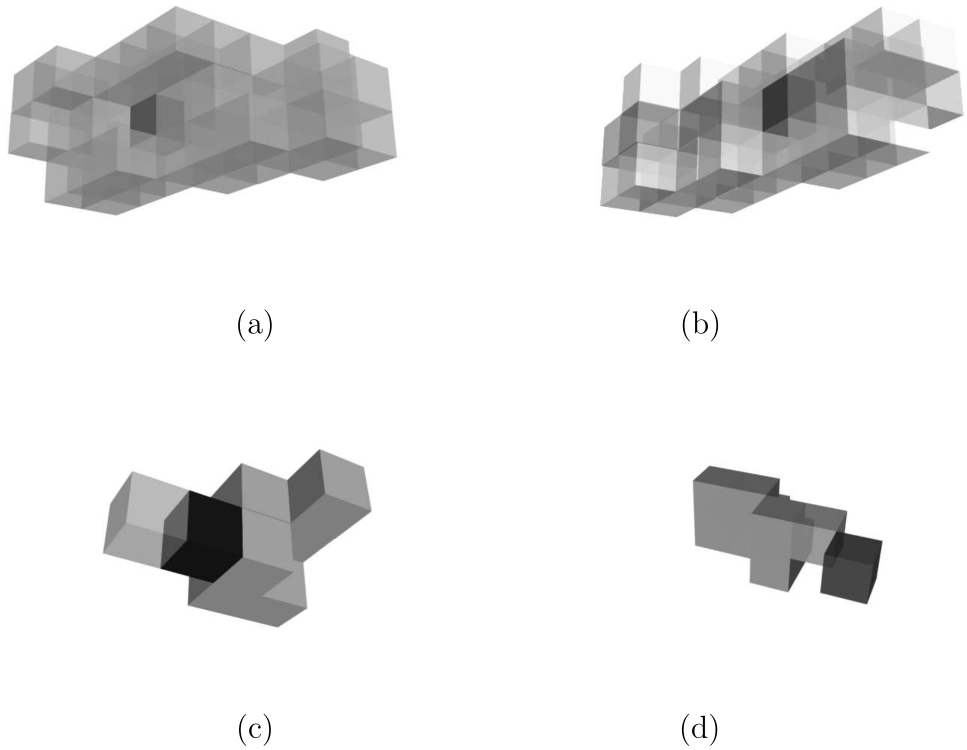

Figure 5: Brain image details showing voxels of types S_{7-9} (dark). *(a,b)* Two images showing the same S_7 voxel (with two opposite faces are exposed to the background). In (b), the object detail is rotated by 180^o around the vertical axis with respect to (a). *(c)* An S_8 voxel, with two pairs of opposite faces exposed to the background. *(d)* Voxel of type S_9, with all its six faces exposed to the background.

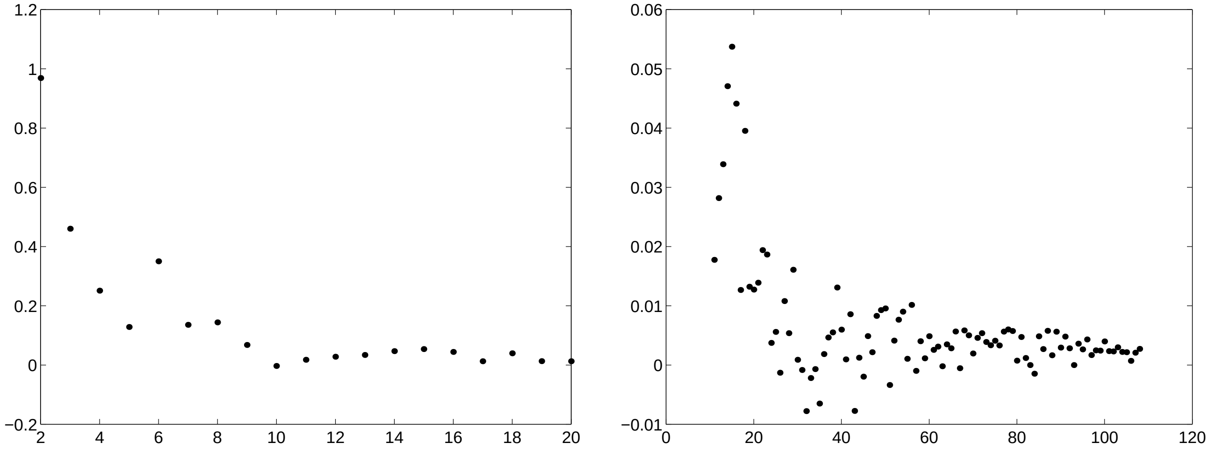

Figure 6: The relative mean estimation error (percent) in estimating the surface area of spheres, as a function of sphere radius (with object-background averaging). *Left*: Small sphere radii. *Right*: Large sphere radii.

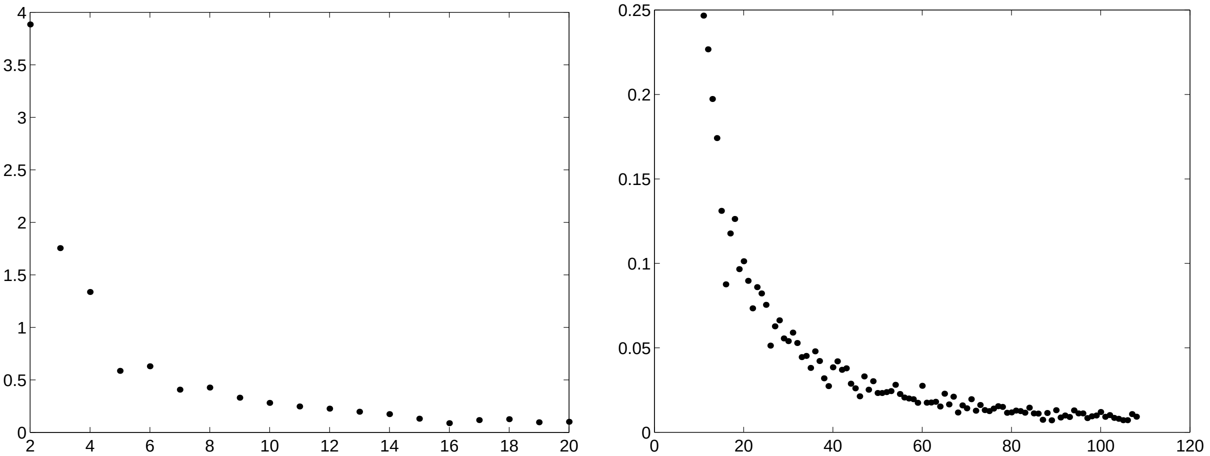

Figure 7: The coefficient of variation (percent) as a function of radius in estimating the surface area of spheres (with object-background averaging). *Left*: Small radii. *Right*: Large radii.

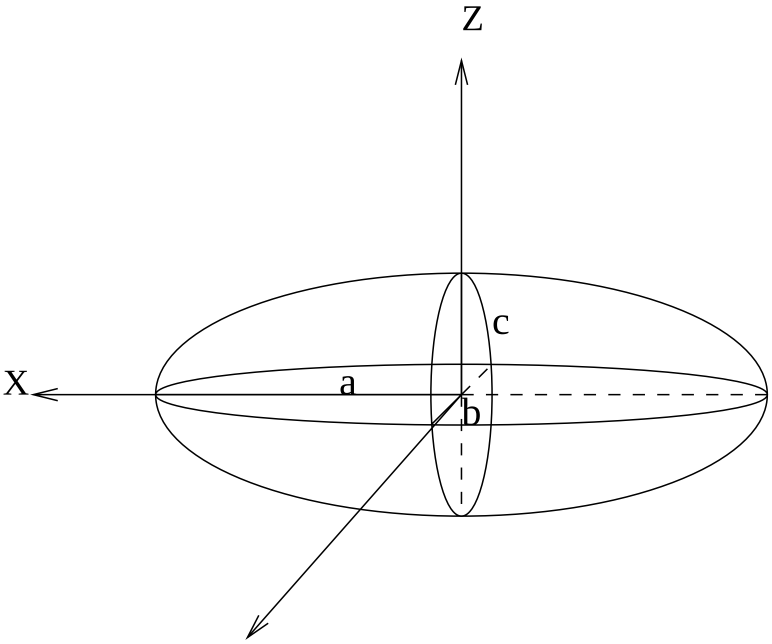

Figure 8: An ellipsoid with semi-main axes (a, b, c), aligned with the coordinate system.

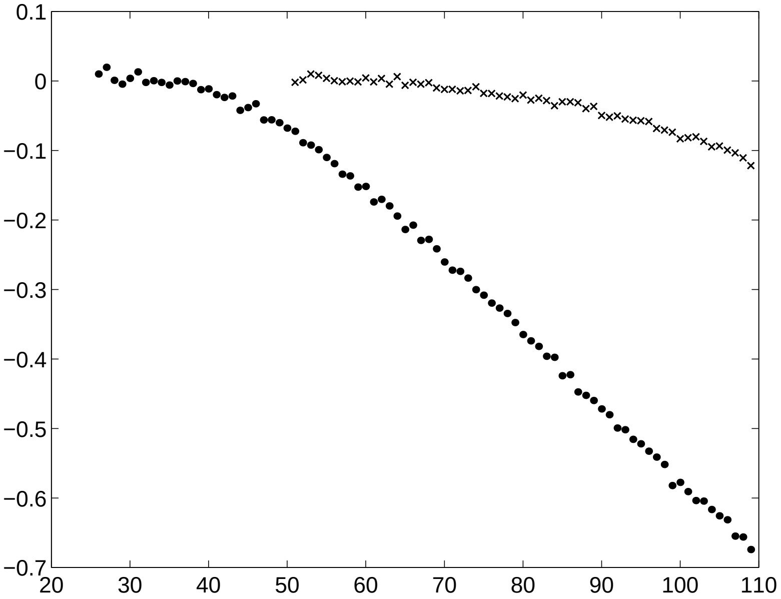

Figure 9: Relative mean estimation error (percent) as a function of ellipsoid main semi-axis a (using object-background surface area averaging). The dots refer to the ellipsoid family $(a, 26, 25)$; the x's to $(a, 51, 50)$.

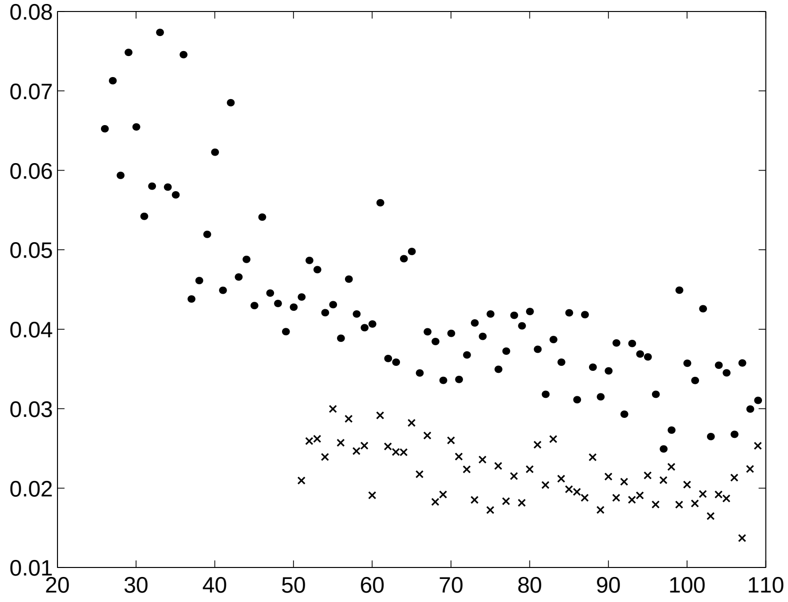

Figure 10: Coefficient of variation (percent) as a function of ellipsoid semi main-axis a (using object-background surface area averaging). The dots refer to the ellipsoid family $(a, 26, 25)$; the x's to $(a, 51, 50)$.

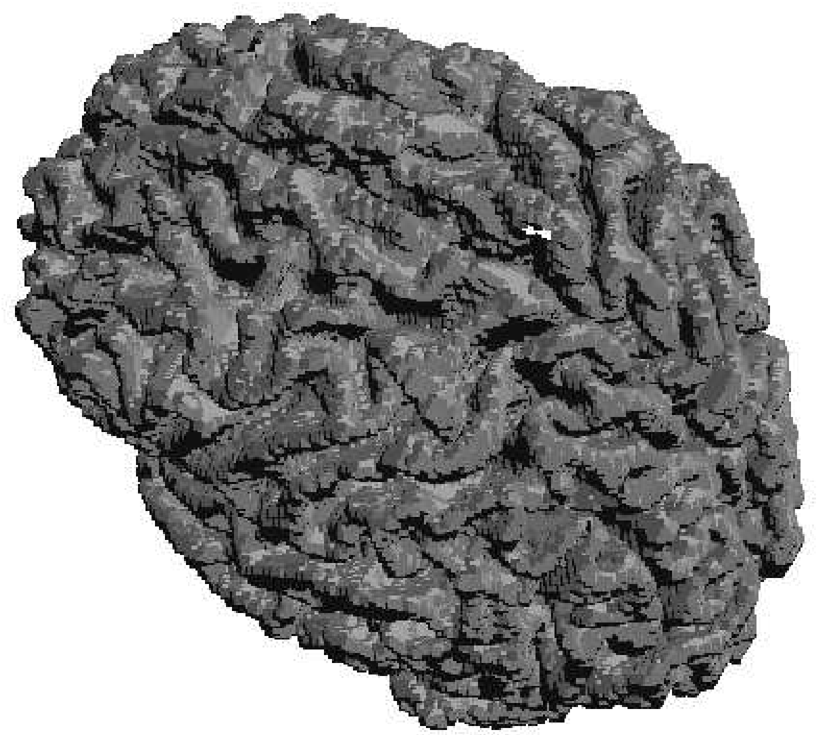

Figure 11: Grey/white matter interface in a segmented MR brain image. The first two key points entered by the user and the connecting path are shown (white).

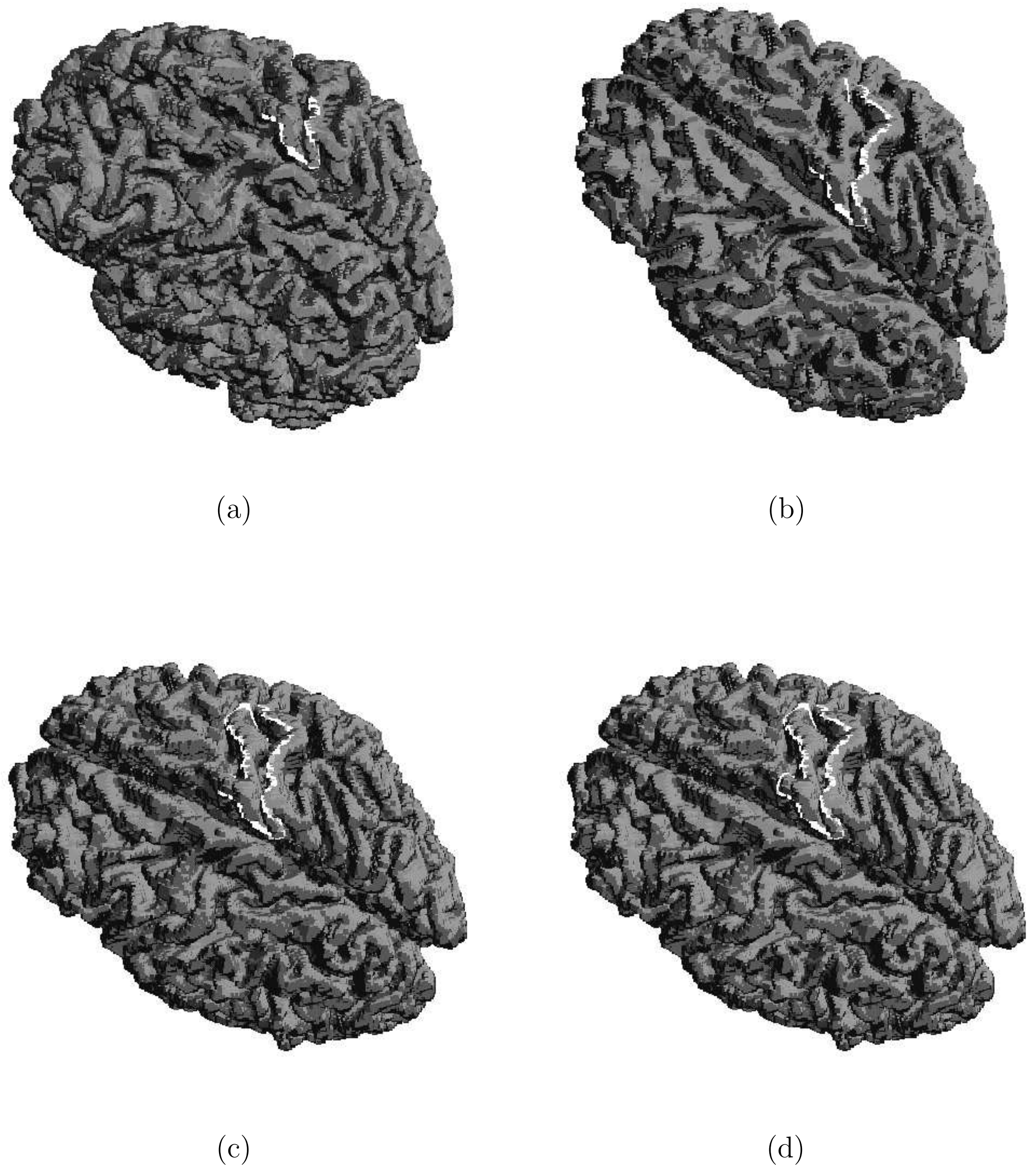

Figure 12: Interactive definition of the region of interest on the surface. (a)-(c) Automatic connection of key points provided by the user. (d) Closure of the contour surrounding the region of interest.

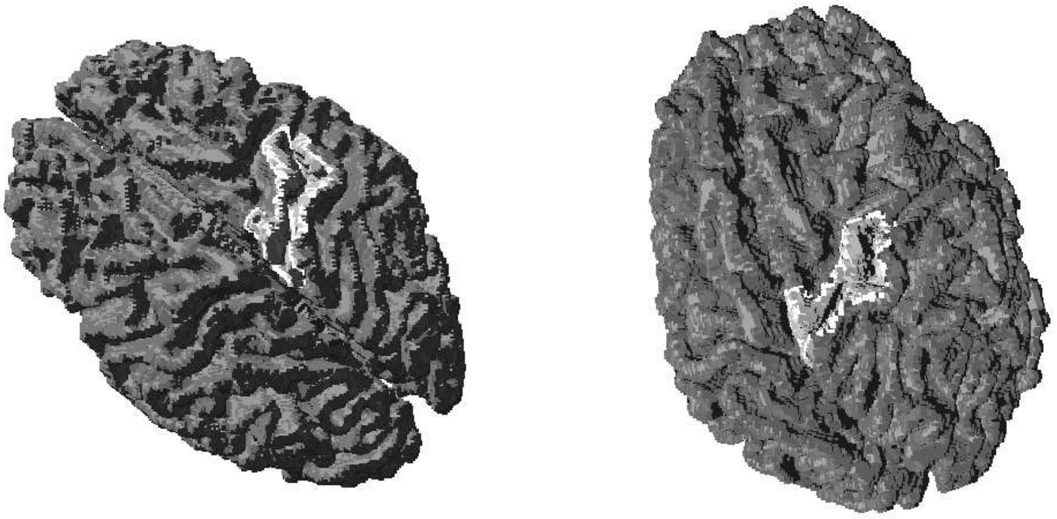

Figure 13: The region of interest on the boundary surface, shown from two viewing points.