Surface Extraction from Binary Volumes with Higher-Order Smoothness

Victor Lempitsky

Abstract—A number of 3D shape reconstruction algorithms, in particular 3D image segmentation methods, produce their results in the form of binary volumes, where a binary value indicates whether a voxel is associated with the interior or the exterior. For visualization purpose, it is often desirable to convert a binary volume into a surface representation. Straightforward extraction of the median isosurfaces for binary volumes using the marching cubes algorithm, however, produces jaggy, visually unrealistic meshes. Therefore, similarly to some previous works, we suggest to precede the isosurface extraction by replacing the original binary volume with a new continuous-valued embedding function, so that the zero-isosurface of the embedding function is smooth but at the same time consistent with the original binary volume. In contrast to previous work, computing such an embedding function in our case permits imposing a higher-order smoothness on the embedding function and involves solving a convex optimization problem. We demonstrate that the resulting separating surfaces are smoother and of better visual quality than minimal area separating surfaces extracted by previous approaches to the problem. We plan to make the code of our algorithm publicly available for researchers working on 3D image segmentation as well as other 3D shape reconstruction applications.

Index Terms—Object Modeling, 3D Image Segmentation, Shape Reconstruction, Volume Visualization, Convex programming, Quadratic programming methods.

◆

1 INTRODUCTION

The problem of surface extraction from binary volumes arises in the post-processing step of several computer vision applications. In particular, binary image segmentation algorithms such as region growing [1], graph cuts [2], or a simple thresholding proceed by labeling image elements as belonging to either foreground or background. The output of such an algorithm is, therefore, a binary-valued segmentation mask. For two-dimensional problems, the segmentation mask can be visualized in several ways, e.g. by superimposing it onto the original image. Over the recent years, there is, however, an ever-increasing demand for 3D image segmentation, where a three-dimensional segmentation mask (a binary volume) needs to be visualized after the segmentation is performed.

In many cases, the user expects the 3D image segmentation result to be presented in the form of a *separating surface*, i.e. a surface that separates the background and the foreground segments of a binary volume. In many applications such as the biomedical imaging, this surface may correspond to the actual physical interface, e.g. the boundary of an organ. The problem of extracting a separating surface from a binary volume also arises within the post-processing step in several other applications such as stereo- or silhouette-based multiview reconstruction [3], [4] or shape-from-range data [5], when the underlying algorithms work with voxel representations and make hard decisions about the voxel occupancy.

Given a binary volume, a separating surface can be extracted as an isosurface corresponding to the median

value (e.g. the zero-isosurface is taken, if the background label is interpreted as -1 and the foreground label is interpreted as 1). These isosurfaces can be efficiently extracted in a form of triangular meshes using the marching cubes algorithm [6]. Such isosurfaces, however, exhibit distracting aliasing artifacts (Figure 1a, Figure 2a). These artifacts have a regular structure and therefore are perceived as "signal" rather than "noise", which leads to their amplification by the human visual system.

The aliasing problem is caused by the fact that a binary volume does not define the separating surface uniquely. In fact, depending on the interpretation, a binary volume is typically consistent with the entire family of separating surfaces. Thus, in this work we interpret a binary volume as a set of hard constraints imposed on the separating surface. Under these constraints, the separating surface must contain the centers of all foreground voxels inside while having the centers of all background voxels outside; whether or not this interpretation has a "physical" meaning depends on the particular algorithm used to compute the binary volume. It can be demonstrated, for example, that this interpretation has a sound geometric justification for the graph cut (or more precisely *GeoCut*) framework [7], which became very popular for segmentation as well as other low-level vision tasks over the last years.

In the paper, we discuss a new method for the extraction of smooth separating surfaces based on the constrained convex optimization of the higher-order smoothness criterion. The results of the method for several binary volumes suggest that it yields higher-quality surfaces as compared to the previous approaches that are discussed in the following section.

V. Lempitsky is with Microsoft Research, Cambridge lab

(a) the binary volume and its zero-isoline   *(b)* minimal-length separating curve   *(c)* our separating curve   *(d)* embedding function computed for *(c)*
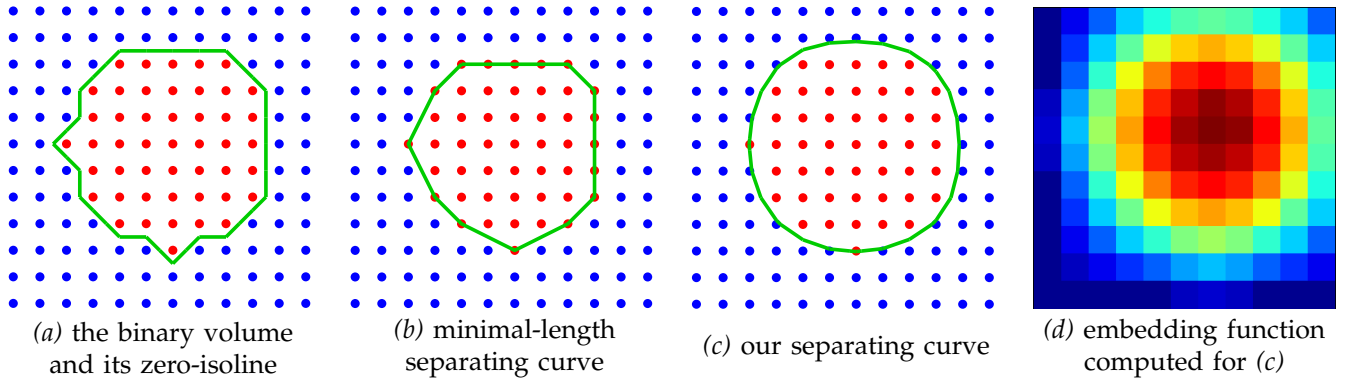
Fig. 1. (*better viewed in color*) A 2D example highlighting the performance of different approaches. The input binary volume is obtained by sampling a circle on a very coarse (12×12) grid, so that red nodes correspond to the interior (+1) and blue nodes correspond to the exterior (-1). Both the zero-isoline *(a)* and the minimal length separating curve *(b)* (as sought by [8]–[10]) suffer heavily from the aliasing artifacts, while our method computes a smooth separating curve *(c)* with the shape close to being circular. *(d)* shows the embedding function computed with our method, for which *(c)* is the zero-isoline.

2 RELATED WORK

The problem of extracting a separating surface can be regarded as the problem of picking one out of an infinitely-large class of surfaces that meet the hard constraints imposed by the binary volume. The zero-isosurface is a choice that is suboptimal from the perceptual point of view, as it lacks smoothness. Some of the initial approaches [11] suggested to overcome the aliasing artifacts (jaggedness and terracing) by a local Gaussian pre-filtering of the binary volume. The problem with that approach, however, is that the terracing/jaggedness effects often require very large kernel and strong filtration to be diminished (let alone completely eliminated), whereas such a filtration smooths out the fine details and, in general, produces oversmoothed surfaces incompatible with the original binary volumes.

To overcome this essential non-locality of the aliasing effects, strategies based on the optimization of global objectives were suggested. Towards this end, [8], [10] evolves the extracted surface to minimize its area, subject to the constraint that it has to remain compatible with the original binary volumes during the evolution. Following up on this approach, [9] introduced the surface extraction method, which solve essentially the same optimization problem (constrained minimal area separating surface) but in the implicit level-set framework [12]. The smoothness is thus introduced prior to the isosurface extraction by the modification of the underlying volume function. In this way, the function values are no longer restricted to be binary (1 or -1). Such an implicit smoothing strategy has several advantages over the explicit smoothing of the isosurface mesh including the ease of handling of topology changes during smoothing as well as the simplicity of imposing the hard constraints.

Choosing the surface with the smallest area [8]–[10] typically leads to separating surfaces that are significantly better perceptually than the zero-isosurface of the original binary volume. Yet, the resulting surfaces still suffer from visually noticeable aliasing, grid-induced artifacts (Figure 1b, Figure 2b). Furthermore, the shrinking bias introduced by the minimal area objective leads to artifacts on thin protrusions and sharp creases of the object.

In this paper, we suggest a new simple criterion that can be used to extract the separating surfaces from binary volumes, yielding surfaces with higher-order smoothness rather than the minimal area property. Such criterion can be used within the implicit smoothing framework, and, notably, has a nice property of leading to convex optimization problems (as opposed to non-convex problems in previous frameworks). Importantly, we demonstrate that the higher-order smoothness imposed by the resulting algorithm allows to obtain separating surfaces with much fewer aliasing artifacts as compared to the area minimization methods [8]–[10].

3 SURFACE EXTRACTION

Problem setting. Assume that a binary volume function $V : \mathcal{G} \to \{-1, +1\}$ is given, where $\mathcal{G}$ is the discrete grid domain $\mathcal{G} = \{1, 2, \ldots, L\} \times \{1, 2, \ldots, M\} \times \{1, 2, \ldots, N\}$. Then, denote with $v_{ijk} \in \{-1, +1\}$ the value of V on the respective node of the grid.

As a result of the smoothing, we are going to obtain a non-binary *embedding function* $F : \mathcal{G} \to \mathbf{R}$, where $f_{ijk} \in \mathbf{R}$ will again denote the value of F on the node of the grid. The obtained function F must be consistent with the binary volume V, so that the zero-isosurface extracted from (the continuous interpolation of) F contains all foreground nodes $\{i, j, k \,|\, v_{ijk} = +1\}$ inside (or on the boundary) and all background nodes $\{i, j, k \,|\, v_{ijk} = -1\}$ outside (or on the boundary). This requirement is equivalent to the following set of hard constraints imposed on F:

$$\forall i, j, k \qquad v_{ijk} \cdot f_{ijk} \geq 0 \,. \tag{1}$$

We seek to obtain F meeting constraints (1) so that its zero isosurface is smooth (in the sense that we discuss below). An idea that we use in our method is to impose smoothness on F directly, so that its isosurfaces also possess a certain degree of smoothness.

Imposing smoothness. The idea of imposing the smoothness directly on the embedding function is used extensively in image segmentation. Thus, the level set frameworks modify the embedding function locally, so that the area of the zero isosurface is also minimized. The TV-minimization framework [13] as well as the GeoCut framework [7] minimize the integral of the absolute value of variation of the embedding function. Again, although the smoothness is imposed on the embedding function rather than on its zero-isosurface[1], the area of the zero-isosurface is provably minimized in this case.

More recently, it has been argued (see e.g. [14]) that one can impose smoothness by minimizing the square (or other powers) of the variation of the embedding function. Unlike the case of the absolute value of the variation, minimizing such quantity for the embedding function does not translate into minimizing some clearly understood functional of the zero-isosurface. Yet, it has been demonstrated to achieve the desired effect of imposing smoothness on this isosurface [14].

The methods discussed above (except level-sets) minimize some function of the first-order variation $|\nabla F|$ or related quantities. As a result, these methods are biased towards the constant embedding functions, as this is the only class of the embedding functions, which are considered absolutely smooth under this definition of smoothness. As the zero-isosurface of the constant function is, in general, empty surface, this bias towards constancy of the embedded function manifests itself as the shrinking bias of the isosurface. The same shrinking bias is inherent to the level set framework [12], that is used in [9] to extract the separating surfaces from binary volumes.

To avoid the shrinking bias and achieve the higher-order smoothness, we suggest to regularize the higher-order derivatives of the embedding function. Thus, in the continuous limit, the following functional is minimized:

$$\int \left(\frac{\partial^2 F}{\partial x^2}\right)^2 + \left(\frac{\partial^2 F}{\partial y^2}\right)^2 + \left(\frac{\partial^2 F}{\partial z^2}\right)^2 dV \to \min. \quad (2)$$

In the discrete setting, the finite-difference approximation is used:

$$\sum_{ijk} \big[(f_{i+1\,jk} + f_{i-1\,jk} - 2f_{ijk})^2 + \\ (f_{i\,j+1\,k} + f_{i\,j-1\,k} - 2f_{ijk})^2 + \\ (f_{ij\,k+1} + f_{ij\,k-1} - 2f_{ijk})^2 \big] \to \min. \quad (3)$$

As in the case of some other functionals, it is hard to associate the minimization of the functional (2) defined

1. The term "0.5-isosurface" would be more consistent with the notation of papers concerned with these frameworks. We still use "zero-isosurface" here to be consistent with the rest of our paper.

on F with the minimization of some functional defined on the zero-isosurface of F. One can, however, analyze which kind of isosurfaces can be observed for the global minima of (2). Thus, the functional (2) clearly biases the embedding function towards a low-degree polynomial, so that the globally minimal embedding functions are the polynomial of degree at most three. In fact, by adding the squares of the mixed derivatives into (2) one could restrict the globally optimal embedding functions to be linear. In either case, however, it may be noticed that the class of globally optimal (i.e. "absolutely smooth") embedding functions is much reacher than just constant functions, and that the class of their isosurfaces include all planar surfaces (as well as some quadric surfaces in the case when mixed derivatives are not included).

Adding a margin. We now seek to obtain an embedding function that is smooth in the sense of (2) and meets the conditions (1). While the regularization (2) on its own does not bias the embedding function to be constant, one may notice however, that combining it with the hard constraints (1) would lead back to the unique and trivial optimal solution $F \equiv 0$. This can be avoided if the hard constraints (1) are made more stringent, ensuring some margin separating the resulting embedding function from the zero solution:

$$\forall i, j, k \qquad v_{ijk} \cdot f_{ijk} \geq m_{ijk}. \quad (4)$$

Here, m_{ijk} are non-negative values, which are strictly positive for some i, j, k, ensuring that the embeding function deviates from zero somewhere. There exist different reasonable choices of margin values that lead to perceptually plausible and similar separating surfaces. Thus, if we denote with B the set of boundary nodes in V, i.e. nodes adjacent (in 26-connectivity) to the nodes of the different values in V, the simple choice for m_{ijk} would be:

$$m_{ijk} = \begin{cases} 0, & \text{if } (i,j,k) \in B, \\ 1, & \text{otherwise.} \end{cases} \quad (5)$$

A marginally better results in our experiments were produced by the margin equal to the (unsigned) Euclidean distance to the set B:

$$m_{ijk} = dist\left((i,j,k), B\right) = \\ \min_{(\alpha,\beta,\gamma)\in B} \sqrt{(i-\alpha)^2 + (j-\beta)^2 + (k-\gamma)^2}. \quad (6)$$

Extracting the surface. To extract the separating surface from the binary volume, our method simply solves the following convex quadratic optimization problem:

$$SOLVE \quad (3) \qquad s.t. \quad (4). \quad (7)$$

The separating surface is then extracted as the zero-isosurface of the optimal embedding function using the marching cubes algorithm [6]. Alternatively, the recovered embedding function can be rendered directly using the volume rendering techniques such as [15].

As we are interested in the values of F near its zero isosurface, we can restrict our computations
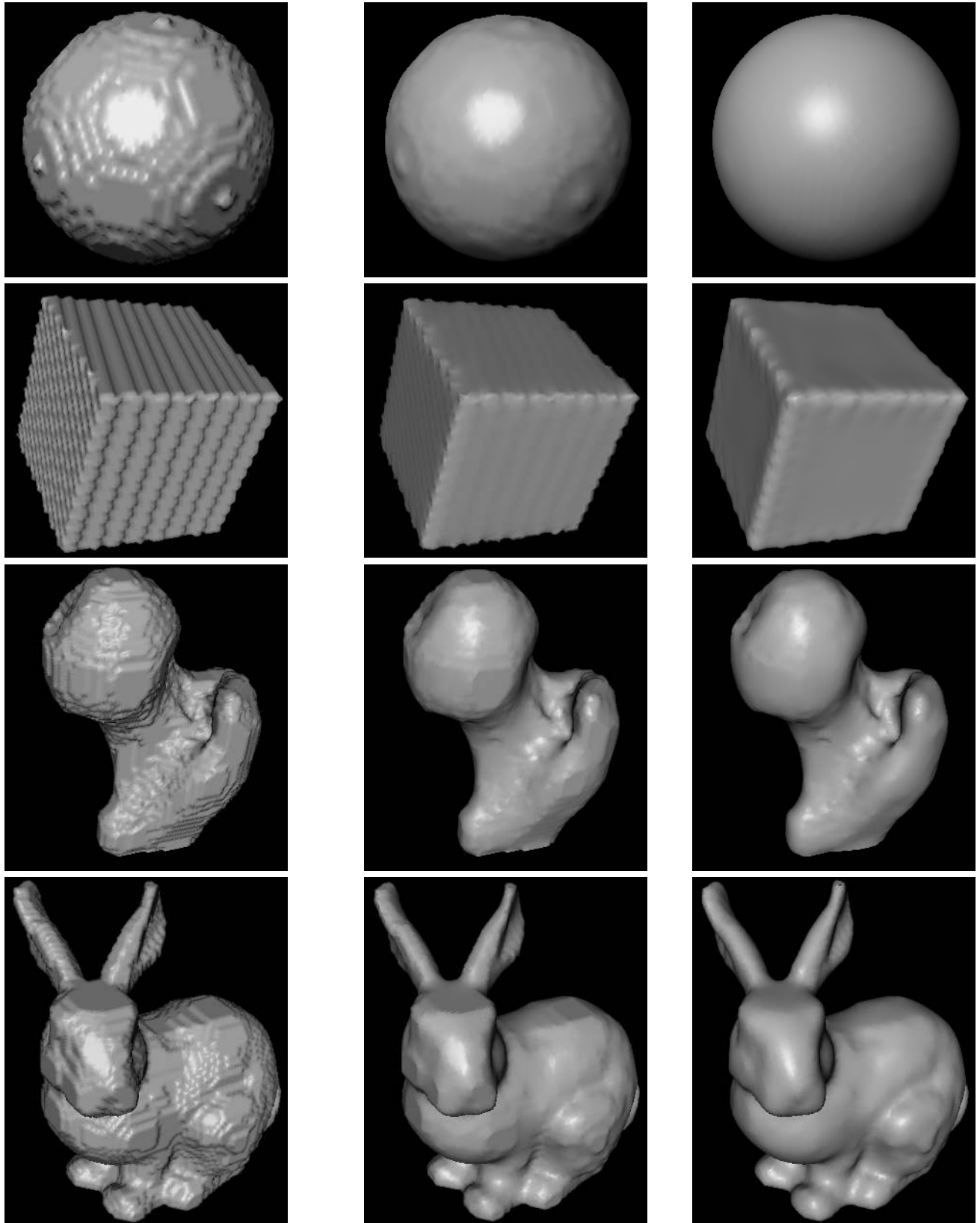
(a) Zero-isosurfaces of binary volumes   *(b)* Minimal area separating surfaces   *(c)* Our separating surfaces

Fig. 2. For a set of synthetic and real-data binary volumes, our method extracts separating surfaces with less aliasing artifacts, as compared to the minimal area approach or the straightforward application of the marching cubes. See the text for the description of the binary volumes.

to the nodes within the narrow band defined as $\{(i,j,k)|\ dist\left((i,j,k),B\right)<C\}$, where C is the constant defining the width of the band which can be set to a small value (e.g. 3) without visually affecting the resulting isosurface (as compared to the computations on the full grid).

The convex quadratic program (7) can be solved using any general purpose sparse second-order optimizer (e.g. the one available in Matlab), which converge to the F function having smooth zero-isosurface within few (e.g. 10) iterations. A simple, first-order (gradient descent) optimization also works well, although it requires more iterations. The distance function required to construct the band and to compute the margin values can be efficiently computed even for large volumes using e.g. the algorithm [16].

4 RESULTS

The suggested algorithm has been evaluated on several synthetic and real binary volumes. In Figure 2a, we present the zero-isosurfaces of the input volumes, while the minimal area separating surfaces computed using the method [9] are shown in Figure 2b. Finally, the separating surfaces extracted with our method are presented in Figure 2c (the margin as defined by (6) was used). It can be observed that while the minimial area approach yields the surfaces that look much better than the original zero-isosurfaces, they still suffer from aliasing (terracing) artifacts. These artifacts are removed and smoother, more naturally-looking surfaces are obtained with our method.

The top two rows in Figure 2 correspond to binary volumes obtained by rasterizing a ball and a cube on low-resolution ($64 \times 64 \times 64$) grids (the cube axes were rotated relative to the grid axes). The third row correspond to the binary segmentation result of an MRI hip joint dataset ($128 \times 128 \times 119$) obtained using the graph cut method [2]. The fourth row correspond to the binary volume ($100 \times 100 \times 79$) obtained from the set of range scans using the method [17]. While for the illustration purposes the volumes in Figure 2 have low resolution, our method scales to much larger volumes due to its banded nature.

For example, Figure 3 shows the closeups of the surface extracted from the $256 \times 256 \times 288$ binary volume. It also demonstrates the main failure mode for our method, which are thin objects (protrusions). Although, our method (Figure 3b) still does marginally better than the minimal area surface approach (in Figure 3a), it can be seen that very thin objects cannot be processed with the method described above in a satisfactory way. The problem occurs because the functional (3) drives the value of the embedding function towards zero, while the margin value (6) does not prevent that, as it is equal to zero for the voxels belonging to thin parts.
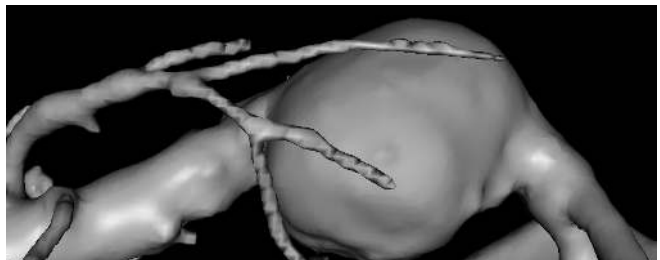
One easy way to improve the performance of the method on thin objects is first to identify the thin parts
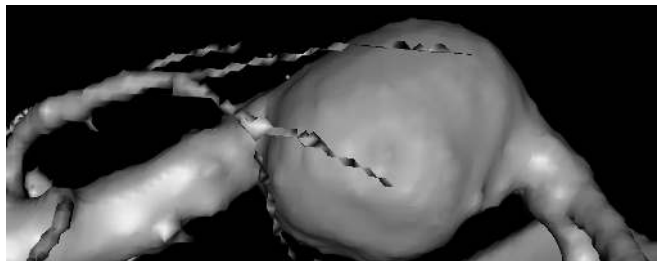
(a) – Minimal area separating surface

(b) – Our separating surface

(c) – Our separating surface, modified margin

Fig. 3. Close-ups of the surfaces extracted from the high-resolution binary volumes with different methods. Both the minimal area approach and our approach struggle to obtain visually consistent surfaces for very thin objects such as vessels. The performance of our method in these regions may be improved by automatically increasing the margin values locally; in other parts, the surface remains unchanged.

to be preserved and then to increase the margin values for these voxels. The thin parts can be identified using simple morphological operations. Thus, we perform morphological *opening* of the binary volume by eroding it in a 6-connected neighborhood, and subsequently dilating it in a 26-connected neighborhood. We then consider all voxels that are inside the object in the original volume but not inside the opening as protrusions that require further processing and raise their margin values m_{ijk} (which the rule (6) always set to zero in these parts) to $\epsilon = 0.25$. As shown in (Figure 3c), this improves the performance of the method in protrusion areas considerably (although, admittedly, the aliasing artifacts can be observed there). Importantly, the surface remains virtually unchanged in other parts.

5 DISCUSSION

We have presented a simple algorithm allowing to extract smooth isosurfaces from binary volumes, which

is a common post-processing task within a range of shape reconstruction applications, in particular 3D image segmentation. Unlike previous methods, that generally aim at minimizing the area of the separating surface, our approach minimizes a higher-order smoothness criterion imposed on the embedding function. Such minimization can be achieved via convex quadratic programming and yields smooth isosurfaces with fewer aliasing artifacts. It remains an interesting question whether similar kind of higher-order smoothness can be used for other tasks, e.g. whether it can be applied to image segmentation directly.

While we have focused on the surface extraction from the binary volumes, it is highly likely that our method will be useful for the segmentation results produced by the methods working with continuous representations [12]–[14], [19], [20] as the criteria within these methods are not designed to extract isosurfaces that are smooth at *subvoxel* levels. E.g., the level-set frameworks [12] typically optimize minimal area-related objectives and therefore isosurfaces extracted from the resulting continuous-valued volumes will be very similar to the minimal area separating surfaces computed with the level set method [9]. Therefore, we believe that our method may be as useful for the extraction of the surfaces from the segmentation results of these methods as it is for the segmentation methods with binary outputs.

It can be argued, at the same time, that for such tasks as multiview reconstruction or shape-from-points the input data may be reused at the surface extraction stage to achieve the subvoxel accuracy (since unlike image segmentation, the initial data for these problems are not sampled on a grid and typically have higher effective resolution). Still, our method may be useful for the approaches solving these problems within the increasingly popular graph-cut [17], [21] and TV-minimization [22] frameworks as a fast, "ready-to-use" solution for the surface extraction.

We plan to make the Matlab code of our approach available for researchers by the time of the publication.

6 ACKNOWLEDGEMENTS

REFERENCES

[1] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, 1994.

[2] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *ICCV*, 2001, pp. 105–112.

[3] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 2, pp. 150–162, 1994.

[4] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," in *ICCV*, 1999, pp. 307–314.

[5] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *SIGGRAPH*, 1996, pp. 303–312.

[6] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *SIGGRAPH*, 1987, pp. 163–169.

[7] Y. Boykov and V. Kolmogorov, "Computing geodesics and minimal surfaces via graph cuts," in *ICCV*, 2003, pp. 26–33.

[8] S. F. F. Gibson, "Constrained elastic surface nets: Generating smooth surfaces from binary segmented data," in *MICCAI*, 1998, pp. 888–898.

[9] R. T. Whitaker, "Reducing aliasing artifacts in iso-surfaces of binary volumes," in *Volume Vizualization*, 2000, pp. 23–32.

[10] R. Bade, O. Konrad, and B. Preim, "Reducing artifacts in surface meshes extracted from binary volumes," vol. 15, no. 1-3, 2007, pp. 67–74.

[11] S. W. Wang and A. E. Kaufman, "Volume-sampled 3d modeling," *IEEE Comput. Graph. Appl.*, vol. 14, no. 5, pp. 26–32, 1994.

[12] S. Osher and J. A. Sethian, "Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.

[13] M. Nikolova, S. Esedoglu, and T. F. Chan, "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM Journal of Applied Mathematics*, vol. 66, no. 5, pp. 1632–1648, 2006.

[14] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, 2006.

[15] S. F. F. Gibson, "Using distance maps for accurate surface representation in sampled volumes," in *IEEE Visualization*, 1998, pp. 23–30.

[16] T. Saito and J.-I. Toriwaki, "New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications," *Pattern Recognition*, vol. 27, no. 11, pp. 1551–1565, 1994.

[17] V. Lempitsky and Y. Boykov, "Global optimization for shape fitting," in *CVPR*, 2007.

[18] C. Schumann, S. Oeltze, R. Bade, B. Preim, and H.-O. Peitgen, "Model-free surface visualization of vascular trees." in *IEEE/Eurographics Symp. on Visualization*, 2007, pp. 283–290.

[19] X. Bai and G. Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting," in *ICCV*, 2007, pp. 1–8.

[20] A. Criminisi, T. Sharp, and A. Blake, "Geos: Geodesic image segmentation," in *ECCV (1)*, 2008, pp. 99–112.

[21] G. Vogiatzis, P. H. S. Torr, and R. Cipolla, "Multi-view stereo via volumetric graph-cuts," in *CVPR (2)*, 2005, pp. 391–398.

[22] K. Kolev and D. Cremers, "Integration of multiview stereo and silhouettes via convex functionals on convex domains," in *ECCV (1)*, 2008, pp. 752–765.