Interactive Segmentation of Volume Data Using Watershed Hierarchies

Michal Hučko
Comenius University, Slovakia
michal.hucko@fmph.uniba.sk

Miloš Šrámek
Austrian Academy of Sciences, Austria
milos.sramek@oeaw.ac.at

ABSTRACT

The available methods for volume data segmentation and/or classification differ in the amount of the required user input on the one side and precision and ability to tweak the obtained results on the other. Automation of the task is more difficult when a general case is considered. In this paper we present an interactive segmentation and classification tool for arbitrary volumetric data, which is based on pre-segmentation of the volume in a hierarchy of homogeneous regions. The hierarchical subdivision allows for interactive adaptation of scale and precision according to the user requirements. The data is processed in three dimensions which minimises the amount of the needed interaction and gives instant overview of the resulting segmentation.

Keywords

Segmentation, user interface, watershed, scale-space.

1 INTRODUCTION

Segmentation methods, which are commonly used with volume data, can be classified in two groups – general methods and model-based methods [PB07]. Unlike general ones, model-based methods are based on certain knowledge about the target objects in the data as, for example, the expected object shape, mean voxel intensity, etc. In this paper we omit these as our goal is to provide a general segmentation tool which can be used with any volume data to segment arbitrary objects.

As the general methods use no additional information about the data which would aid in the process of segmentation, they require a greater amount of user interaction either in the form of process control, specification of parameters tailored to the current task or post-processing of the result. Our aim is to minimise this interaction while still leaving full control of the segmentation process to the user.

2 RELATED WORK

Common approach in volume data segmentation consists of selection of a region or object of interest in slices of the volume. The most basic general segmentation method used is manual segmentation where a user delineates the object of interest in the slices by hand. Although it is applicable at all times, its heavy user interaction demands are apparent. To speed up delineation of contours the LiveWire method [MMBU92] may be used. To obtain the desired results, a suitable cost function has to be first specified. The LiveWire method speeds up the process of contour drawing if the object of interest is clearly separated from the rest of the data. If this condition is not satisfied for the current task, difficulties in cost function specification arise resulting in slow downs – user intervention is required and the method is reduced to manual segmentation.

As commonly used data sets have rather large dimensions, performing the segmentation on each slice is tedious and time consuming. An option is to segment only certain slices and interpolate the contour in the in-between slices [SPoP00]. Instead of the interpolation of the contour one may interpolate LiveWire control points instead and let the system compute contour in intermediate slices from the interpolated control points [SOB]. Precision of the resulting segmentation is dependent on the used interpolation and also on the set of key-slices. Problems might arise when topology or shape of the contour change rapidly between slices. Validation and potential correction of the interpolation is necessary.

Another common segmentation method is thresholding where voxels with intensities in certain range are selected. This method can be easily applied to certain data where tissue types can be distinguished by intensity (e.g. bone tissue in CT data), but applications to other data modalities or tissue types may pose a problem. If the task is to separate various objects of the same

tissue type thresholding may be used for preprocessing to reject clearly non-target voxels and thus localise the area of interest.

Two methods based on detection of homogeneous regions instead of contours are worth mentioning. General region growing requires user specified seed points and a suitable homogeneity criterion. Data is flooded from the seed points as long as the homogeneity criterion is satisfied, creating a homogeneous region. The watershed segmentation [VS91] is usually performed on a gradient image, which is treated as a topographic height map. Homogeneous regions with voxels with small gradient magnitude form valleys having voxels with high gradient magnitude on region borders as ridges. Although there exist various algorithms for computing the watershed transform, variants of two approaches are common – simulation of downhill water flow for each voxel or immersing of the relief into water. Exhaustive study of existing watershed algorithms is provided by Roedink and Meijster [RM00].

A broader overview and more detailed description of the existing segmentation methods can be found in [PB07].

2.1 Interaction techniques

All of the previously mentioned segmentation approaches differ in the way how a user can interact with the data to modify a partial result until the desired segmentation is achieved. For example, when delineating a contour the user directly sees the partial result and can undo recent steps if the contour starts to diverge from the desired position.

Thresholding requires numeric input – the threshold. Usually, a user is provided with a histogram from which the most suitable threshold value can be estimated. Clean separation of various tissue types based only on voxel intensity is rare and thus it may be difficult to find an optimal threshold value. Usage of other methods for refining the segmentation from thresholding (morphologic operations, connected component labeling, etc) is therefore convenient [STBH92].

Watershed segmentation produces a highly oversegmented result, especially if the data is spoiled by noise. Some methods allow merging of neighbouring regions if the shared border is weak (gradient magnitude is low). This situation is illustrated in figure 1 which shows gradient magnitude image of a CT head dataset slice – the corresponding watershed segmentation can be seen in figure 3 (red borders). The aim is to create segmentation in which the target object is labeled by a unique label. If this is not happening user intervention is usually required. For example, an interaction technique called marker-based watershed segmentation can be used [HP03], where a
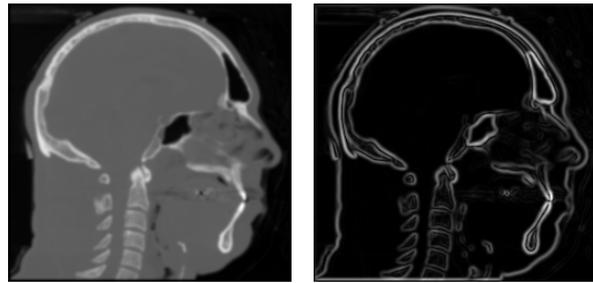

Figure 1: Slice from the Visible human male [Ack98] CT dataset with corresponding gradient magnitude image on right.

user specifies special include and exclude points in the data which prohibit merging of neighbouring regions if new region would contain markers of both types.

Watershed hierarchies [Beu94] were used in a technique based on interaction with slice views by Cates et al [CWJ05]. In such hierarchy, the order in which regions are merged defines a binary merge tree. Original regions form leaves and non-leaf nodes represent regions formed by merging of two regions – its children nodes. This tree may be used to segment an object from a data with a possibility to select large parts of the object by specification of high-positioned tree nodes and to refine the border by adding/removing low-positioned nodes.

In the paper by Armstrong et al [APB07] an extension of LiveWire or Intelligent Scissors called Live Surface was proposed. Data is presegmented into hierarchy of regions. Initial regions are computed using tobogganing [MB99] creating result equivalent to watershed segmentation (depends on the used watershed definition). For merging of the regions for higher levels in the hierarchy a special metric is used, which is based on the mean voxel intensity/colour and intensity/colour variance of a region. Segmentation is done by specifying two types of markers – inner and outer – which are then used to create a graph-cut in the region neighbourhood graph with minimal cost. Markers can be entered on an arbitrary cross section of the volume or directly in the 3D view allowing to add/remove parts to/from the border of an already segmented object.

3 THE SEGMENTATION TOOL

All of the segmentation approaches mentioned in the previous section were either proposed for two dimensional images or, if targeted to segmentation of 3D data, were used only for interaction in two dimensional space – on respective slices – or provided limited possibilities to modify the resulting segmentation directly in the 3D visualisation of the data. In our approach we let the user directly control the segmentation process by selecting fragments of the target object in 3D space. As manual segmentation by pixels/voxels is too cumbersome, data
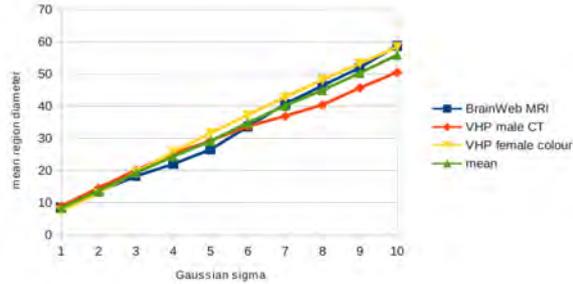
Figure 2: Measured dependency of mean region size on the Gaussian filter sigma. Three data sets were used – an MRI head, VHP male CT head and VHP female colour head dataset. Measured was diameter of minimal bounding sphere centered at the region's centre of mass.

is pre-segmented by the watershed transform as it creates homogeneous regions which can be used instead of single voxels.

3.1 Preprocessing

The watershed segmentation produces highly over-segmented results, because for each local minimum in the (usually gradient image of) original data a region is created. This can be a serious problem especially if noise is present in the data. To cope with this, data is usually first smoothed by the Gaussian filter. Smoothing, however, also shifts edges and removes weak edges, resulting in merging of regions which a user might desire in certain cases separated. Therefore, we perform the watershed segmentation on a sequence of derived data sets where each one is produced from the original by smoothing with a gradually increasing degree (increasing of the Gaussian sigma). As Koenderink showed [Koe84] this produces a set of images where on each image details smaller than certain size are increasingly ignored (the scale-space approach). As can be seen in figure 2, measurements on three different head datasets showed that dependency of mean bounding sphere diameter of regions on used Gaussian filter sigma can be approximated with function 1.

$$d(\sigma) = 5(\sigma - 1) + 10 \qquad (1)$$

To further reduce the starting number of regions, region merging based on mean region intensity or some other criterion can be used.

In order to correct position of the shifted edges, caused by smoothing, to the original position specified by the unsmoothed data or at least data at the lowest level of smoothing, an aligned region hierarchy is build. Spatial alignment of borders of corresponding regions on different hierarchy levels is achieved by the technique based on the maximum number of spatially overlapping voxels [SD02]. Thus, if later required, aligned regions
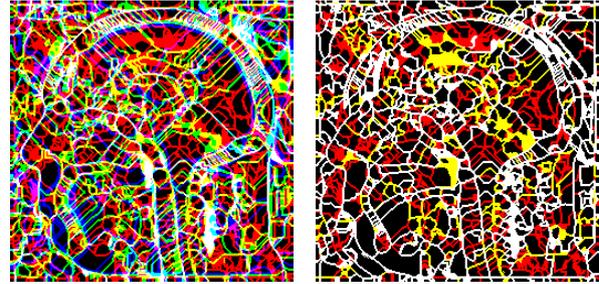

Figure 3: Three levels of watershed segmentation on visible human – male CT dataset. In the red channel are boundaries of regions for Gaussian smoothing with sigma 3, in green for sigma 5 and in blue for sigma 7. Overlapping boundaries are blended. On the left are the original watersheds, on the right the regions are aligned showing only red, yellow (red plus green) and white (all three levels) boundaries.

positioned higher in the hierarchy can be unambiguously decomposed to multiple smaller regions on some lower level.

As the last step of preprocessing the relevant neighbour information is extracted from the original voxel data and stored in a file. The described technique works equally well for scalar data (CT and MRI scans) as well as for multi-field data (dual energy CT, T1, T2 and PD MRI data etc). In the second case the watershed transform is performed on a gradient volume obtained as maximum of individual gradient fields.

3.2 GUI interaction

GUI of the segmentation application is shown in Figure 4. In this section all parts of the application as well as the segmentation workflow will be explained.

After the hierarchy is loaded into the application it is displayed in a tree widget allowing the user to navigate through it. Hovering or selecting a node highlights the corresponding region in a 3D visualisation window (Figure 5). This allows the user to choose an initial fragment of the region of interest. Depending on the target object and created hierarchy, regions on higher levels might consist of multiple target objects (e.g. various bones of skull). As pre-segmentation was performed at various scale levels, there is no need to repeat the process when the aforementioned problem arises – moving to lower levels in the hierarchy until the objects are separated is possible.

After the initial region is selected it is possible to display neighbouring regions which satisfy certain similarity criteria (Figure 6, also see section 3.3). The user can then select either one neighbour belonging to the target object by clicking into the 3D view or by selecting the neighbour in the list or can select all neighbors. Deselection of an undesired region is possible in a similar way, too. Once a neighbour is added/removed to/from
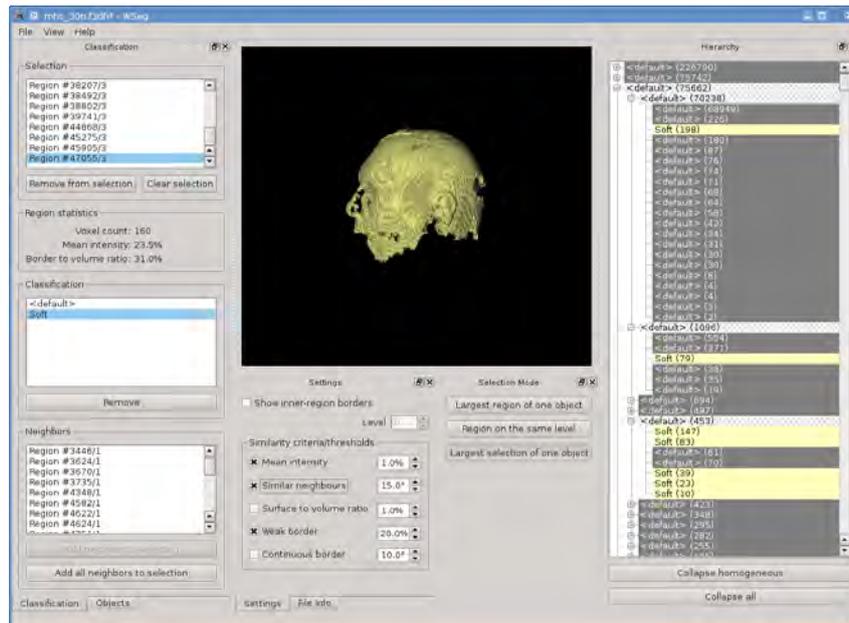
Figure 4: The segmentation tool: the region hierarchy is on the right. Left side from top to bottom: the list of selected regions, classification classes, the list similar neighbours of the regions in selection. At centre-bottom similarity criteria can be specified and above is the 3D visualisation and interaction window.
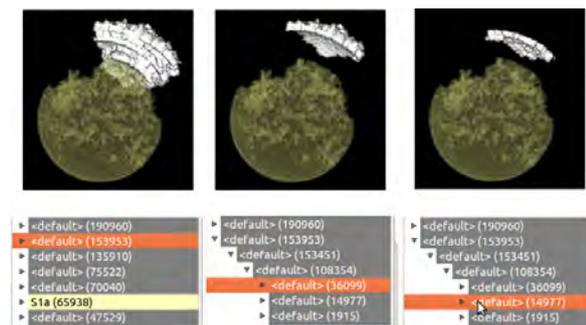

Figure 5: Selection of objects using the tree widget. The scene consist of concentric spheres/ellipsoids with different density and was intentionally spoiled by noise and superimposed low frequency density variation to make segmentation of the ellipsoids by thresholding impossible. Left: A region on the highest hierarchy level (white) was selected. One can see that it is composed of pieces of several ellipsoids. I we go down in the hierarchy (bottom row, in this case by two levels of the hierarchy), these pieces are separated and can be individually selected. The yellow region is a partially segmented and labeled part of the central sphere.

the selection, updated neighbours of the new selection are displayed. This allows traversal of the target object until all fragments are in the selection. Now classification class for the object can be created and all selected regions can be classified (Figure 6 right). Already classified regions are during selection refining displayed dimmed to provide context whereas the current selection is always highlighted.

3.3 Similarity criteria

To decide whether neighbouring regions are similar we implemented five different similarity criteria. Each criterion can be turned on or off and has a separate threshold value. When deciding whether two neighboring regions are similar all criteria which are turned on have to return a positive answer. Detailed description of the similarity criteria can be found below.

When searching for similar neighbours all neighbouring regions of the regions in the current selection are visited. As the selection might contain regions on different levels in the hierarchy we have to prevent multiple inclusion of same region – once directly and second time by including parent of the region. Naturally, regions positioned higher in the hierarchy are accepted in favor of the lower positioned child.

A second issue arises from the fact that we would like to move to a lower level of the hierarchy and to repeat the search for similar neighbors there, if we were not successful on a higher level. In this case, first, all regions in the selection are decomposed to regions on the lowest level – some lowest-level regions were directly present in the selection and some were selected indirectly by selection of its ancestor. Now we iterate through all pairs of the selected lowest-level regions and their neighbours (also at the lowest level). For both regions in the pair we ascend as many levels up in the hierarchy, until the original region in the selection which was decomposed to the currently processed lowest-level region is found. This leaves us with a list of pairs – (indirectly) selected region with its neighbour

– for each level not higher than level of the original selected region. Subsequently we start with the similarity tests. If a region and its neighbor on the higher level do not pass the selected similarity test we descent by one level and again perform the tests. By this higher positioned similar neighbours are found first.

A detailed description of the different similarity criteria follows:

Mean intensity Simple criterion comparing mean intensities of neighbouring regions. Difference in the intensities have to be in the specified interval for the test to pass. If the data contains multiple bands, the test must pass for all bands to be considered successful.

Similar neighbors In this criterion the prevalent directions of intensities in the region's neighborhood are compared. All neighbors of a region are visited and their center of mass is computed by averaging their geometric centroids weighted by their mean intensity. Subsequently, the direction vector, which points in the direction of growing intensity, is obtained by subtracting the center of mass from the geometric centroid of the region. Comparison of an angle between such vectors of two neighboring regions against a user defined threshold yields result of the similarity test.

Surface to volume ratio As working directly with the volume data would be slow and would increase memory requirements significantly, only derived information is used – mean intensity of a region, voxel count, etc. To compare region shapes ratio of the number of region's surface voxels to its total voxel count is used. The computed value is normalized to the $[0,1]$ range where 0 represents sphere-like objects and 1 string-like objects.

Weak borders For two regions to be similar in this criterion their border should have small mean gradient magnitude. Unlike the mean intensity criterion which compares mean values for whole regions, this criterion uses the original gradient data which were used during the creation of the most-detailed watershed segmentation. Intensity of the regions is n this criterion irrelevant.

Continuous border This criterion tries to find border in the data which spans multiple region boundaries. For two regions to pass this test they have to have a common neighbor. Both faces – first region with the common neighbor and second region with the common neighbor – have to be similar. For this, the angle between mean normal/gradient vectors of the faces is examined. Faces which are too small are ignored as their mean gradient vectors are based on too small set of values.

4 RESULTS

The application was implemented in C++ and uses OpenGL with GLSL. It was tested on the VHP dataset (the male head CT dataset). Individual bones of the skull, which are connected without a well defined border, were successfully segmented (figure 7 left). We also tested the application on an MRI scan of a human head. Figure 7 right shows segmentation of the brain cortex with certain gyri labeled as separate objects.

Preprocessing with three levels of watershed hierarchies and 3 additional levels of merging by density similarity took about 10 minutes. Both segmentations were produced in about 30 minutes.

In contrast to other methods, we intentionally omitted the possibility to interact through slices or cross sections to investigate the possibility for interaction only through the 3D view. If desired, display of the slices or cross sections can be easily added to the application, which may be then used for an initial selection.

If compared to traditional segmentation approaches, our method is most similar to region growing – assuming that mean intensity criterion is used. Neighbours to selected regions having similar intensity can be iteratively added to the selection until only regions with large difference in intensity remain. Because of the manual operation, user can omit regions which evaluate similar, but are not part of the target object. This can be essential when separating two or more objects of same or similar intensity, but different shape (as exemplified by segmentetion of the brain cortex in individual gyri). Automatic, or semi-automatic methods fail to separate these, if the interface is too weak or not present due to partial volume effect.

5 CONCLUSION

The presented segmentation and classification tool allows fast insight into data and fast segmentation of target structures while still leaving full control of the segmentation in user's hands. The amount of user interaction depends on data properties – resolution, presence of noise or other artefacts. Different similarity criteria were presented which should simplify and thus speed-up localisation of similar neighbouring regions in the data. Still, the user interaction is mainly done directly in the 3D visualisation window instead of slices. Operation in the visualisation window also gives direct visual feedback.

The concept was tested on the VHP dataset by segmentation of the bones of human skull and on MRI data by segmentation of respective gyri of the brain cortex.
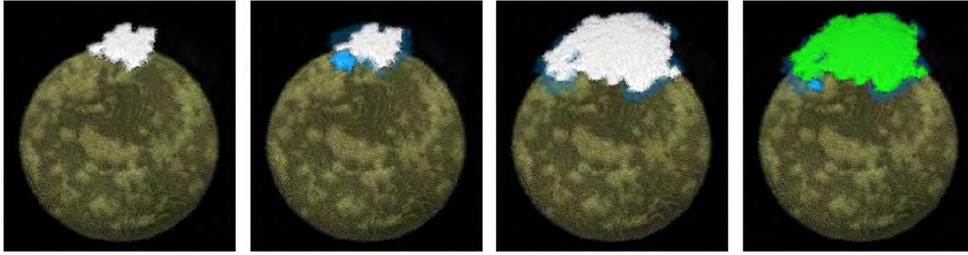
6 ACKNOWLEDGEMENTS

Figure 6: From left: selection of similar neighbors starting from an initial region. The blue regions are selection candidates (density similarity was used). White objects are already selected objects, the green color means that the object has already been labeled. The highlighted blue candidate object can be individually added to the selection or all candidates can be added at once.
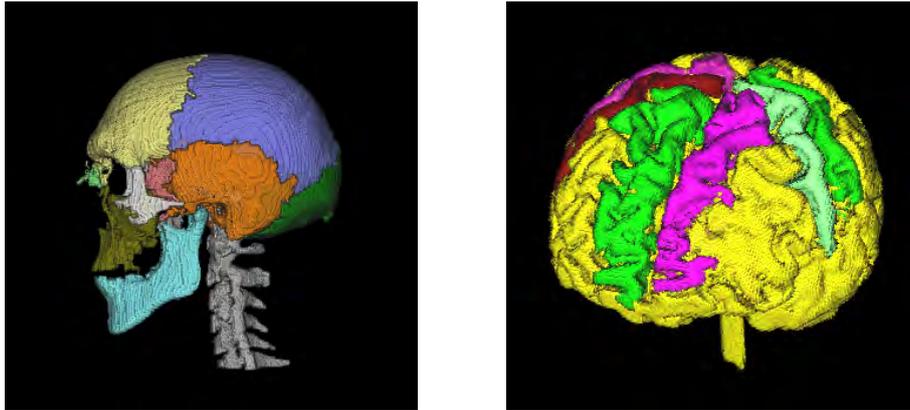

Figure 7: Left: segmentation of skull bones of the VHP male CT head dataset. Right: segmentation of brain and gyri in the MRI head dataset.

7 REFERENCES

[Ack98] M.J. Ackerman, *The visible human project*, Proceedings of the IEEE **86** (1998), no. 3, 504–511.

[APB07] Christopher J. Armstrong, Brian L. Price, and William A. Barrett, *Interactive segmentation of image volumes with live surface*, Comput. Graph. **31** (2007), no. 2, 212–229.

[Beu94] S. Beucher, *Watershed, hierarchical segmentation and waterfall algorithm*, Mathematical morphology and its applications to image processing (J. Serra and P. Soille, eds.), Kluwer Academic Publishers, 1994, pp. 69–76.

[CWJ05] Joshua E. Cates, Ross T. Whitaker, and Greg M. Jones, *Case study: an evaluation of user-assisted hierarchical watershed segmentation.*, Medical Image Analysis (2005), 566–578.

[HP03] Horst K. Hahn and Heinz-Otto Peitgen, *Iwt-interactive watershed transform: a hierarchical method for efficient interactive and automated segmentation of multidimensional gray-scale images*, vol. 5032, SPIE, 2003, pp. 643–653.

[Koe84] J.J. Koenderink, *The structure of images*, Biological Cybernetics **50** (1984), no. 5, 363–370.

[MB99] E.N. Mortensen and W.A. Barrett, *Toboggan-based intelligent scissors with a four parameter edge model*,

Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR '99), vol. II, 1999, pp. 452–458.

[MMBU92] Eric Mortensen, Bryan Morse, William Barrett, and Jayaram Udupa, *Adaptive boundary detection using Live-Wire two-dimensional dynamic programming*, IEEE Computers in Cardiology (1992), 635–638.

[PB07] B. Preim and D. Bartz, *Visualization in medicine: theory, algorithms, and applications*, The Morgan Kaufmann series in computer graphics, Morgan Kaufmann, 2007.

[RM00] Roerdink and Meijster, *The watershed transform: Definitions, algorithms and parallelization strategies*, FUNDINF: Fundamenta Informatica **41** (2000).

[SD02] M. Sramek and L. I. Dimitrov, *Segmentation of tomographic data by hierarchical watershed transform*, Journal of Medical Informatics and Technologies **3** (2002), 161–169.

[SOB] Zein Salah, Jasmina Orman, and Dirk Bartz, *Live-wire revisited*.

[SPoP00] Andrea Schenk, Guido Prause, and Heinz otto Peitgen, *Efficient semiautomatic segmentation of 3d objects in medical images*, In Proc. of Medical Image Computing and Computer-assisted Intervention (MICCAI, Springer, 2000, pp. 186–195.

[STBH92] T. Schiemann, U. Tiede, M. Bomans, and K. H. Höhne, *Interactive 3D-segmentation*, Visualization in Biomedical Computing 1992, Proc. SPIE 1808 (R. A. Robb, ed.), SPIE, Chapel Hill, NC, 1992, pp. 376–383.

[VS91] L. Vincent and P. Soille, *Watersheds in digital spaces: An efficient algorithm based on immersion simulations*, IEEE Transactions on Pattern Analysis and Machine Intelligence **13** (1991), 583–598.