Segmentation of SBFSEM Volume Data of Neural Tissue by Hierarchical Classification*

Björn Andres[1], Ullrich Köthe[1], Moritz Helmstaedter[2],
Winfried Denk[2], and Fred A. Hamprecht[1]

[1] Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg
[2] Max Planck Institute for Medical Research, Heidelberg, Germany

Abstract. Three-dimensional electron-microscopic image stacks with almost isotropic resolution allow, for the first time, to determine the complete connection matrix of parts of the brain. In spite of major advances in staining, correct segmentation of these stacks remains challenging, because very few local mistakes can lead to severe global errors. We propose a hierarchical segmentation procedure based on statistical learning and topology-preserving grouping. Edge probability maps are computed by a random forest classifier (trained on hand-labeled data) and partitioned into supervoxels by the watershed transform. Over-segmentation is then resolved by another random forest. Careful validation shows that the results of our algorithm are close to human labelings.

1 Introduction

Experimental methods used in neuroscience either provide detailed information about a small fraction of all neurons (e.g. single cell recordings, Golgi staining), or information averaged over relatively large areas (e.g. functional MRI, EEG). However, detailed knowledge of the *complete* connectivity pattern of all neurons would be of tremendous value for the understanding of neural computation [1]. This has now become a realistic objective because recent advances in electron microscopy make it possible to obtain 3D images of sufficiently large volumes (up to 1 mm^3) at sufficiently high isotropic resolution ($\approx$ 25 nm). The resulting data sets contain about 10^{12}-10^{13} voxels and can no longer be analyzed manually.

This paper deals with the automatic segmentation of neurons from serial block-face scanning electron microscopy (SBFSEM) [2]. This method collects image stacks by combining backscattering-contrast low-vacuum electron microscopy with a specially designed microtome: Every image in a stack is obtained by scanning the current surface (blockface) of the sample from which a slice is then cut off to expose the next plane. Sections can be as thin as 30 nm, and a pixel pitch of 10-20 nm is achievable in the lateral directions. Thus, the resolution of the volume obtained is sufficiently uniform to permit truly 3-dimensional image analysis. Moreover, alignment problems [3] between consecutive slices are avoided from the outset since the sample remains stationary.

* The authors gratefully acknowledge contributions by and fruitful discussions with Sebastian Seung, Viren Jain, Kevin Briggman, and Bjoern Menze.

G. Rigoll (Ed.): DAGM 2008, LNCS 5096, pp. 142–152, 2008.

The appearence of neurons and other brain tissue varies widely in these data, and we believe that adequate models must be learned rather than hand-crafted. To break the learning problem down into managable parts, we opt for a hierarchical strategy. First, the image structure around each voxel is encoded by a sufficiently rich set of linear and non-linear filter responses computed in the voxel's neighborhood. Next, these features are used as input for a non-linear classification of cell membrane vs. cell interior. Third, an over-segmentation into supervoxels is computed on the basis of the classification scores; finally, adjacent supervoxels are merged whenever a second classifier determines that their common surface arises from over-segmentation. Starting with an initial *over*-segmentation simplifies processing because one only needs to remove false boundaries, not add missing ones. In addition, the segmentation should have closed inter-neuron surfaces without gaps, since we are ultimately interested in the neurons, i.e. regions. Both requirements suggest the use of the watershed algorithm for initial segmentation. Finally, the reconstruction of neural connectivity requires the representation of both the geometry and spatial topology of the cells, and we chose the GeoMap [4] as our underlying data structure to facilitate this. We perform experiments showing that the accuracy of our method is comparable to the human performance on the test data set.

2 Related Work

The results reported here directly complement work by Jain et al. [5] who automatically segmented the same SBFSEM data set by means of a convolutional neural network (CNN) with more than 34,000 adjustable parameters. They show that this approach outperforms Markov random fields, conditional random fields, and anisotropic diffusion. The CNN does not require specification of a feature set and is, in this sense, less biased by prior expectations: Its main design parameter is its topology. On the other hand, training a CNN is quite expensive (in the order of weeks on a PC), whereas the strategy presented here achieves the same accuracy at much lower cost (section 3.1). In other work on similar data sets [6,7] axon tracking is suggested, i.e. a specialized model-based approach, which does not cover all cell types found in the data and may thus work only for selected areas and for certain types of processes.

The approach proposed here shares aspects with existing work on supervised learning for boundary estimation, and on hierarchical segmentation. The closest in spirit is perhaps [8], which introduced the term *superpixel* for the elements of an initial 2D over-segmentation, and demonstrated that hierarchical region merging on the basis of statistics learned from the superpixels can lead to a significant improvement of the initial segmentation. There are three major differences to our approach: First, their initial segmentation is determined by means of normalized cuts, which is much more expensive than watersheds because one has to solve a large eigenvalue problem. Second, learning is only applied to superpixel merging, not to initial segmentation. Third, the significance of boundaries was determined by a linear classifier, whereas non-linear classification is essential for our application. On the other hand, they make use of geometric cues which we have not yet incorporated in our approach.

Boundary computation from learned edge probabilities has recently seen considerable attention. Statistical edge detection was considered in [9,10,11] who compute a rich feature set on the basis of brightness, color, and texture (and Gestalt cues in [11]) and learn corresponding edge probabilities from manual segmentations by various statistical models (non-parametric, linear regression, and boosting respectively). Consistent improvements over traditional edge detectors were reported. The use of edge probabilities as elevation maps for watershed segmentation has been suggested only very recently by [12,13], who compute probabilities from k-nearest-neighbors and logistic regression, respectively.

In contrast, hierarchical methods for the reduction of the watershed algorithm's characteristic over-segmentation have been investigated intensively. A common example is multiscale watershed segmentation [14]. Unfortunately, the underlying assumption that exactly the relevant regions survive after smoothing, is not fulfilled in our application, because long thin regions (as are common in brain tissue) often disappear despite being significant. Another popular approach is based on the region adjacency graph (RAG) of the initial segmentation. The edge weights in the RAG reflect the dissimilarity of adjacent regions and/or the contrast of their common boundary. Usually, these weights are selected heuristically (e.g. by the watershed dynamics [15]) or by simple statistical criteria (e.g. the difference between the region's means). The RAG is then simplified by means of graph partitioning (e.g. normalized cut [16], graph-cut [17], fast region merging [18], and the waterfall algorithm [19]), which determines the regions to be merged. In contrast, we determine edge weights in a more elaborate way (supervised learning on a rich feature set with a random forest classifier), similar to [8]. This approach is justified by the fact that we can obtain quite satisfactory segmentations by simply thresholding on the edge weights.

3 Algorithm

Stacks of SBFSEM images are segmented in three steps. First, the probability of each voxel to represent a cellular membrane or not is computed by means of a random forest (RF) classifier [20] that was trained on a manually segmented subset of the data. Second, these probabilities are interpreted as an elevation map for the watershed algorithm, which produces an over-segmentation into supervoxels. Finally, supervoxel boundaries are classified as either "*salient*" (representing a true cellular membrane) or "*incorrect*" (being an artifact of the over-segmentation) by another RF classifier.

3.1 Voxel Classification

In order to decide for each voxel whether it represents a membrane, 63 features (cf. table 1) are computed and a RF is trained to generalize a manual labeling of a subset of the voxels. These labelings have been created by specially trained

Table 1. Features used for classification. *Left*: Local features for voxel classification. *Right*: Non-local features for the classification of faces. Derivatives were computed with Gaussian filters at $\sigma = 1$.

Local voxel features:	
I) Generic features	
1	original gray value
2	difference of Gaussian filters (σ_1=0.8, σ_2=3)
3	gradient magnitude
4–6	sorted eigenvalues of the structure tensor of 3x3x3 neighborhoods
7–9	as 4–6 but 5x5x5 neighborhoods
10–12	as 4–6 but 7x7x7 neighborhoods
13–15	sorted eigenvalues of the Hessian
II) Statistics over 3x3x3 neighborhoods	
a) Distribution of the gray value	
16	standard deviation
17	third moment
18	fourth moment
19	minimum
20	0.25-quantile
21	median
22	0.75-quantile
23	maximum
b) Distribution of the gradient magnitude	
24–31	cf. 16–23
III) Statistics over 5x5x5 neighborhoods	
32–47	cf. 16–31
IV) Statistics over 7x7x7 neighborhoods	
48–63	cf. 16–31

Non-local face features:	
I) Geometric features	
1	log of the number of voxels in the face
2	log of one plus the difference of the number of voxels in the two adjacent supervoxels
3	log of the sum of numbers of voxels in the two supervoxels
II) Distribution of the gray value (over the face)	
4	mean
5	standard deviation
6	third moment
7	fourth moment
8	minimum
9	0.25-quantile
10	median
11	0.75-quantile
12	maximum
III) Distrib. of the gradient magnitude	
13–21	cf. 4–12
IV) Distrib. of the greatest eigenvalue of Hessian	
22–30	cf. 4–12
V) Signal-based features	
31	absolute difference of the mean gray values of the two supervoxels
VI) Distrib. of the RF votes of voxel classification	
32–40	cf. 4–12

human tracers [21]. Prior to feature extraction, we interpolate the volume image by cubic splines to twice the resolution in each dimension (i.e. eight times as many voxels). As discussed in [22], non-linear operations such as the ones we apply during feature extraction increase the band-limit of the volume image and can lead to aliasing artifacts if the resolution is not increased accordingly. The trained RF, consisting of k decision trees, assigns a membrane probability to each voxel which equals the fraction of decision trees classifying the given voxel as membrane.

3.2 Watershed Segmentation

While the voxel classification accuracy of the above algorithm is in line with the variance between human labelers (see section 4.1), this is not necessarily sufficient for good segmentation: A one-voxel gap in a surface is insignificant in terms of voxel-count error, but can already lead to a false merger between distinct neurons. In order to avoid this, we detect boundaries by applying the *watershed algorithm* [23] (which guarantees closed boundaries) to the probability map $p(\boldsymbol{x})$

instead of simply comparing with a threshold probability. In our hierarchical setting, the resulting over-segmentation is an advantage, because the initial watersheds represent the *complete* boundary evidence. This is best demonstrated by the energy-minimization interpretation of the watershed transform [24]: Assuming that local minima of the boundary probability do not lie on the true boundary, one can define the watershed segmentation as the Voronoi tesselation of the plane with respect to these minima (the "region seeds"), such that every point is assigned to the seed with minimal *topographic distance*, where the topographic distance between two points is defined as the minimum of the functional $\int_C |\nabla p(\boldsymbol{x})| d\boldsymbol{x}$ over all paths C connecting the two points. Region boundaries (watersheds) are formed along ridges of $p(\boldsymbol{x})$, i.e. at points with identical topographic distance to two or more seeds. This condition also applies to ridges with low evidence, so that gaps cannot occur.

We denote the watershed regions as *supervoxels* emphasizing their role as smallest homogeneous regions derivable from the *content* of the data. Each supervoxel contains several sampling points. We can compute spatially adaptive statistics by integrating over these points. Supervoxels statistics are superior to spatially uniform integration as used in the initial voxel classification. Since similar statistics are needed for region surfaces, we use a region-growing variant of the watershed algorithm which keeps a one-voxel wide explicit boundary between all regions. We can thus collect boundary statistics directly from the sampling points in each surface. To give good statistics, supervoxels should not be too small. Therefore, we use the "marker-based" watershed algorithm, where all points with $p(\boldsymbol{x}) < 2\%$ are marked as seeds. In addition, we erased watershed regions smaller than 1000 voxels. We show in section 4.2 that the resulting regions are still an over-segmentation of the manual labeling.

3.3 Merging of Supervoxels

In the third processing step, adjacent supervoxels are merged if and only if they are separated by at least one face that does not represent a cellular membrane but is instead a result of the over-segmentation. This classification problem is again addressed by supervised learning; a second RF is trained to generalize from a training set of labeled faces and face features. The 40 features (cf. table 1) are computed from each face f and the two f-adjacent supervoxels s_1 and s_2. These features include statistics of distributions over f, s_1 and s_2 and hence exploit the data-dependent non-local nature of the supervoxels. In counting the numbers of voxels in these structures, we make rudimentary use of geometry. Labels (salient/incorrect) for a training set of faces are derived from the same labeling of voxels that has been used in the first step. For each face f and the two f-adjacent supervoxels s_1 and s_2, we compute the two connected components c_1 and c_2 of the manual labeling of intracellular space that have the largest overlap with s_1 and s_2, respectively. The face f is labeled as "incorrect" precisely if $c_1 = c_2$. With the RF trained, we obtain for each face the fraction of decision trees that classify this face as "salient".

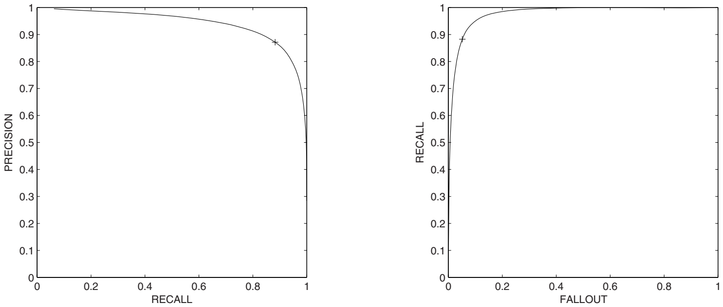

Fig. 1. Precision recall graph (left) and receiver operator characteristic (right) of the voxel classification ("membrane vs. rest"), for thresholds $t \in [0, 1]$ on the probability from the RF. The indicated points on the curves correspond to $t = 0.706$ where the error rate on the training set is minimal.

4 Application and Results

We have applied the proposed algorithm to a volume image of rabbit retina ($321 \times 187 \times 100$ voxels) that was acquired by SBFSEM at $26.2 \times 26.2 \times 50$ nm^3 resolution (cf. figure 3). As discussed in 3.1, we processed a block of $641 \times 373 \times 199$ voxels obtained from the raw data by cubic spline interpolation. Labels for intracellular space and membranes were obtained from a manual tracing of cells as described in 3.1. The labeled data were split into a training set ($\approx 5 \cdot 10^5$ voxels) and a disjoint test set ($\approx 1.3 \cdot 10^6$ voxels) exactly as in [5]. They served as a reference for very careful validation of our methods.

4.1 Voxel Classification

We trained a RF of 255 decision trees on 10^5 voxels sampled randomly from the stratified training set, such that both classes are represented equally. The obtained probability map is shown in figure 3. For binary classification a threshold t was chosen to minimize the fraction of misclassified voxels on the training set. The precision-recall curve and receiver operator characteristic for varying t are depicted in figure 1. At the optimal $t = 0.706$, 7.02% of the voxels in the training set and 8.00% in the test set are misclassified. The confusion matrices for this threshold are given in table 2. This error rate is comparable to the disagreement of different experts ($\approx$ 9%) and the performance of the CNN (also $\approx$ 9%) as reported in [5]. In order to estimate the variability of misclassification over the test set, we computed a subsampling statistic of the confusion: 10^3 samples of $5 \cdot 10^3$ voxels were drawn from the $1.3 \cdot 10^6$ voxel test set without replacement. Table 2 shows accumulated mean and standard deviation of the confusion. The fraction of misclassified voxels is $8.0\% \pm 0.4\%$.

Table 2. *Top*: Confusion of the voxel classifier on the training and test sets. CM and CI denote classification as membrane/intracellular tissue, TM and TI denote the true labeling. The error rate is 7.02% on the training set ($\approx 5 \cdot 10^5$ voxels) and 8.00% on the test set ($\approx 1.3 \cdot 10^6$ voxels). *Bottom*: Subsampling statistics of the confusion on 10^3 samples of $5 \cdot 10^3$ voxels drawn without replacement from the test set. The error rate is $8.0\% \pm 0.4\%$.

Training	CM		CI	
TM	**25,0%**	124402	**3,3%**	16556
TI	**3,7%**	18353	**68,0%**	337878

Testing	CM		CI	
TM	**14,1%**	195088	**3,5%**	48125
TI	**4,5%**	62720	**77,9%**	1079931

Subsampling	CM		CI	
TM	$14.08\% \pm 0.48\%$	704 ± 24	$3.46\% \pm 0.27\%$	173 ± 13
TI	$4.54\% \pm 0.30\%$	227 ± 15	$77.92\% \pm 0.57\%$	3896 ± 28

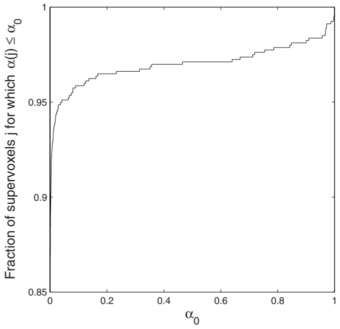

Fig. 2. Cumulative distribution of the under-segmentation index over the supervoxels that overlap with the manual tracing. Manual investigation of the upper 5% of supervoxels revealed that under-segmentation is either caused by very small volumes of less than 1000 voxels or by an incomplete manual tracing of objects at the borders of the tracing bounding box.

4.2 Watershed Segmentation

We ran the watershed algorithm (cf. 3.2) on the RF probabilities, yielding 2037 supervoxels separated by 12966 faces (cf. figure 3). As the watershed segmentation can only be coarsened in the third processing step, we need to make sure that no severe under-segmentation occurs. We therefore considered the supervoxels $\{s_1, \ldots, s_n\}$ from the watershed segmentation and the connected components $\{c_1, \ldots, c_m\}$ from the manual tracing of intracellular tissue and counted, for each s_j, the voxel overlap a_{jk} with each of the c_k. We considered only those supervoxels that overlap at least one of the connected components (only part of the volume was manually segmented into 219 connected components). For each of these supervoxels s_j, an under-segmentation index $\alpha(j)$ was computed as the second largest $a_{jk}/|c_k|$ with $|c_k|$ being the number of voxels of c_k. The

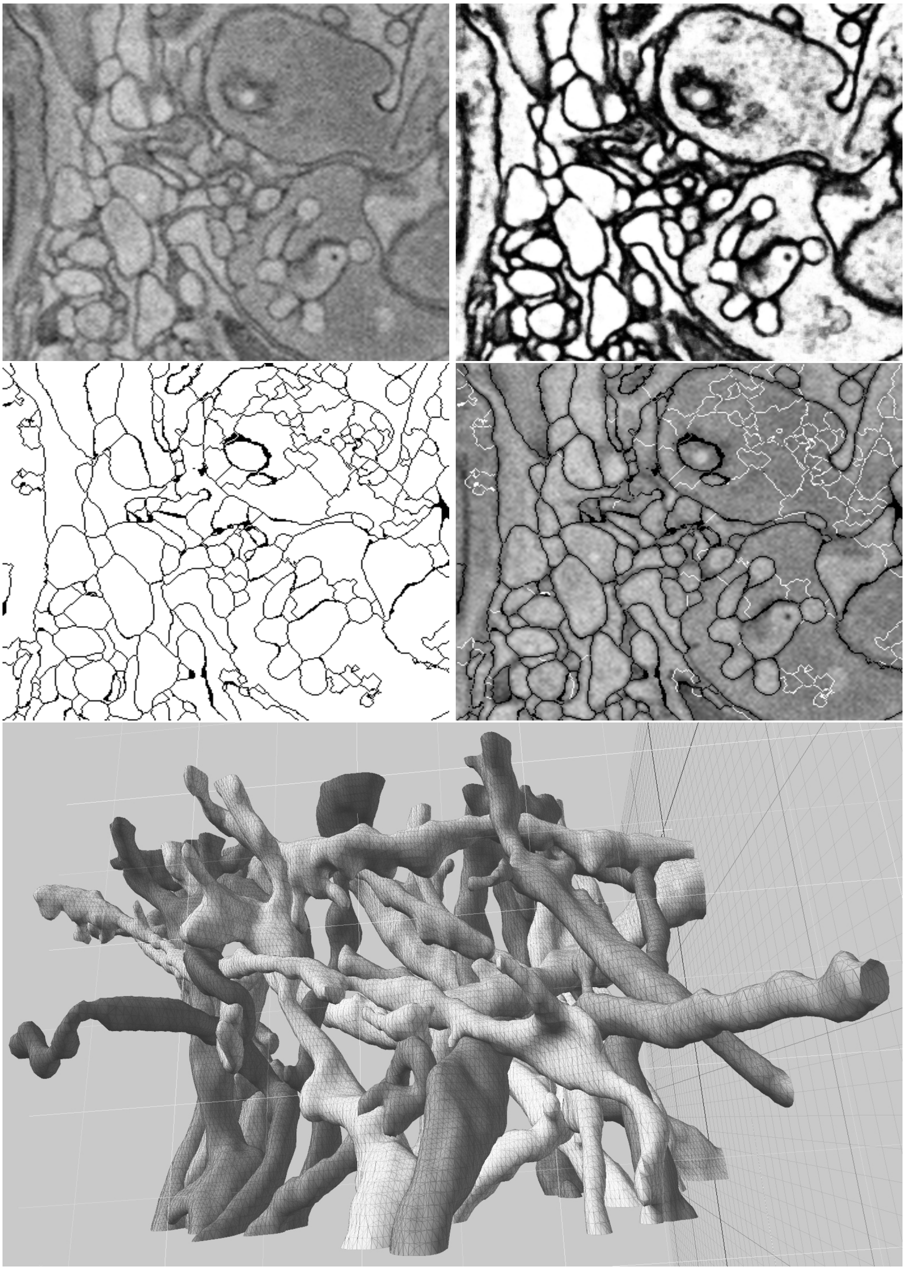

Fig. 3. Interpolated SBFSEM data from rat retina. Membrane probability according to the random forest used for single voxel classification (high probabilities are black). Supervoxels and segment faces from initial watershed segmentation. Classification of faces. Supervoxels that share an “incorrect” face–shown in white–are merged. 3D reconstruction of a fraction of the final segmentation.

Table 3. Confusion of the face classifier in a 5-fold stratified cross-validation. CS and CI denote classification as salient/incorrect, TS and TI denote the true labeling. The fraction of misclassified faces is 3.6% ± 1.6%.

	CS		CI	
TS	**48.0**% ± 0.6%	185.4 ± 2.4	**2.0**% ± 0.6%	7.6 ± 2.4
TI	**1.7**% ± 1.0%	6.4 ± 3.8	**48.3**% ± 1.0%	186.6 ± 3.8

under-segmentation index exceeds 25% for 27 (out of 2037) supervoxels. Figure 2 shows the cummulative distribution of α over the supervoxels considered.

Compared to the 219 connected components in the manual tracing, this number is significant, so we manually investigated where problems occur. We found that 23 out of these 27 segments merge one relatively large component of the manual tracing with a very small one. These errors may be due to the fact that the supervoxel size threshold has not yet been included in the optimization. This will be further investigated in the future. The remaining four supervoxels with an over-segmentation index above 25% are actually correct, but the connection is located outside the manually labeled region.

4.3 Merging of Supervoxels

We trained a second RF of 255 decision trees to classify faces as either "salient" or "incorrect". Labels for training sets of faces were defined as described in 3.3, yielding $n_s = 1361$ salient and $n_i = 969$ incorrect faces. We estimated the classifier performance by means of a 5-fold stratified cross-validation. To that end, we selected five subsets of labeled faces $T_1, \ldots, T_5$ by drawing, for $j \in \{1, \ldots, 5\}$, $n_c = 193$ salient and as many incorrect faces without ever replacing. For each of these sets T_j, we trained the classifier on the union of the remaining four sets ($8n_c$ faces) and computed its confusion on T_j. For an optimal probability threshold $t = 0.5$, 3.6% ± 1.6% of the faces are misclassified. Table 3 shows the confusion matrix. In order to classify the entire set of faces, a training set $T_{\max}$ of maximal size was obtained by taking all the n_i faces that are labeled as "incorrect" and drawing n_i of the n_s salient faces randomly without replacement. Results are depicted in figure 3.

5 Conclusion and Perspectives

We have shown that, given a rich set of (non-)linear features, classification-driven watershed segmentation can be used to obtain a supervoxel segmentation of SBFSEM volume images with only minor under-segmentation. In a second hierarchical step, non-local image statistics and non-linear classification reliably identify the over-segmenting faces of these supervoxels. Extensive experiments demonstrate performance on par with human labelers. Our results support the conclusion of [5] that the classification problem requires the use of non-linear classifiers with sufficient flexibility. In the future, we will improve the initial segmentation to ensure that supervoxels do not under-segment the data. This may be facilitated by anisotropic diffusion of the probability map. Furthermore, we will replace threshold-based supervoxel merging by more powerful inference using graphical models. This will likely require additional levels in the hierarchy, allowing us to include geometric cues as region shapes become more pronounced.

References

1. Briggman, K.L., Denk, W.: Towards neural circuit reconstruction with volume electron microscopy techniques. Curr. Opinion in Neurobiology 16, 562–570 (2006)
2. Denk, W., Horstmann, H.: Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. PLoS Biology 2, 1900–1909 (2004)
3. Kaynig, V., Fischer, B., Buhmann, J.M.: Probabilistic Image Registration and Anomaly Detection by Nonlinear Warping Technical Report. ETH Zürich (2008)
4. Meine, H., Köthe, U.: The GeoMap: A unified representation for topology and geometry. In: Brun, L., Vento, M. (eds.) GbRPR 2005. LNCS, vol. 3434, pp. 132–141. Springer, Heidelberg (2005)
5. Jain, V., Murray, J.F., Roth, F., Turaga, S., Zhigulin, V., Briggman, K.L., Helmstädter, M.N., Denk, W., Seung, H.S.: Supervised learning of image restoration with convolutional networks. In: Proceedings of the ICCV 2007 (2007)
6. Jurrus, E., Tasdizen, T., Koshevoy, P., Fletcher, P.T., Hardy, M., Chien, C.B., Denk, W., Whitaker, R.: Axon tracking in serial block-face scanning electron microscopy. WS Med. Image Comp. and Computer-Assisted Intervention (2006)
7. Macke, J.H., Maack, N., Gupta, R., Denk, W., Schölkopf, B., Borst, A.: Contour-propagation algorithms for semi-automated reconstruction of neural processes. Journal of Neuroscience Methods 167(2), 349–357 (2008)
8. Ren, X., Malik, J.: Learning a classification model for segmentation. In: Proceedings of the ICCV 2003, vol. 1, pp. 10–16 (2003)
9. Konishi, S., Yuille, A., Coughlan, J., Zhu, S.: Statistical edge detection: Learning and evaluating edge cues. PAMI 25(1), 57–74 (2003)
10. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI 26(5), 530–549 (2004)
11. Dollar, P., Tu, Z., Belongie, S.: Supervised learning of edges and object boundaries. In: Proceedings of the CVPR 2006, vol. 2, pp. 1964–1971 (2006)
12. Derivaux, S., Lefevre, S., Wemmert, C., Korczak, J.: On machine learning in watershed segmentation. IEEE WS Mach. Learning for Signal Proc., 187–192 (2007)
13. Levner, I., Zhang, H.: Classification-driven watershed segmentation. IEEE Transactions on Image Processing 16, 1437–1445 (2007)
14. Vanhamel, I., Pratikakis, I., Sahli, H.: Multiscale gradient watersheds of color images. IEEE Transactions on Image Processing 12, 617–626 (2003)
15. Grimaud, M.: New measure of contrast: the dynamics. In: Gader, P.D., Dougherty, E.R., Serra, J.C. (eds.) Proc. Image Algebra and Morphological Image Processing III, vol. 1769, pp. 292–305. SPIE (1992)
16. de Bock, J., de Smet, P., Philips, W.: Watersheds and normalized cuts as basic tools for perceptual grouping. In: Proc. ProRISC 2004 (2004)
17. Stawiaski, J., Decenciere, E.: Region merging via graph-cuts. In: 12^{th} International Congress for Stereology, ICS 2007 (2007)
18. Haris, K., Efstratiadis, S.N., Maglaveras, N., Katsaggelos, A.K.: Hybrid image segmentation using watersheds and fast region merging. IEEE Transactions on Image Processing 7, 1684–1699 (1998)

19. Marcotegui, B., Beucher, S.: Fast implementation of waterfall based on graphs. In: Ronse, C., Najman, L., Decencière, E. (eds.) Proc. ISMM 2005, pp. 177–186 (2005)
20. Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)
21. Helmstaedter, M., Denk, W.: Forthcoming paper (2008)
22. Köthe, U.: Edge and junction detection with an improved structure tensor. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 25–32. Springer, Heidelberg (2003)
23. Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. PAMI 13(6), 583–598 (1991)
24. Nguyen, H.T., Worring, M., van den Boomgaard, R.: Watersnakes: Energy-driven watershed segmentation. PAMI 25(3), 330–342 (2003)