Grayscale and Color Image Segmentation using Computational Topology

Márton Vaitkus*
Supervised by: Tamás Várady†

Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
Budapest / Hungary

Abstract

In this paper, we present image segmentation algorithms based on tools from computational algebraic topology and Morse theory. We build our implementations on a very general clustering algorithm [4], developed by Chazal et al., which has been adapted for both grayscale and color image segmentation. By building up a simplicial complex incrementally - filtering its simplices by values of a scalar function - we can assign a quantity called persistence to its topological features, measuring their "lifetime" in the construction. Combined with concepts from Morse theory, this allows us to construct and simplify a watershed-type segmentation of the complex using a Union-Find algorithm, guided by a single intuitive scalar parameter and supported by theoretical guarantees for topological consistency and robustness. In the case of grayscale images, the complex is an 8-connected pixel adjacency graph which is filtered by pixel values or the absolute value of the image gradient. For color images a point cloud in an appropriate color space is clustered by filtering a Vietoris-Rips neigbourhood graph via Gaussian density estimation and taking spatial proximity into account.

Keywords: Image Segmentation, Computational Topology, Clustering

1 Introduction

Segmenting an image to meaningful parts is a fundamental operation in image processing. Despite tremendous progress in recent decades, the problem remains challenging, with the quality and reliability of hand-made segmentations still unsurpassed by fully automatic methods and it is probably safe to assume that no simple, straightforward algorithm can realistically aspire to solve the problem in general.

General and algebraic topology are usually regarded as highly abstract branches of pure mathematics, see [20],[18] for accessible introductions; but the last decade or so saw the development of *computational topology*, which has made it possible to utilize powerful mathematical concepts. The field in its current form was initiated by Edelsbrunner, Letscher and Zomorodian in [16], which introduced the concept of *persistence* and Morse simplification of spaces. Since then, topological methods have found many applications, including but not limited to data analysis, image processing, Computer-Aided Geometric Design and reverse engineering, shape recognition, signal processing, machine learning and sensor networks.

In this paper we present segmentation algorithms for both grayscale and color images, based on the concept of topological persistence. In both cases we base our approach on a general point cloud clustering algorithm developed by Chazal et al. of INRIA Saclay [4]. On an abstract level this method takes as input any simplicial complex (e.g. a graph or mesh) with a scalar function defined on its vertices and produces a watershed-type segmentation of it. Simplification is carried out on-the-fly, controlled by a single intuitive scalar parameter called persistence.

For grayscale images an adjacency graph is segmented by the values of a gradient filter. The resulting method is similar to that of *Topological Watersheds*, introduced by Bertrand, Couprie et al. [9],[3],[10]; but it is formulated in a drastically different framework, has a more general appeal, and arguably leads to more straightforward implementations. Our approach also bears a resemblance to the *Maximally Stable Extremal Regions* (*MSER*) method of feature detection [24],[13], which assesses the importance of image regions based on their persistence along a range of intensity threshold levels.

For color images we apply the general algorithm as proposed by Chazal et al., on the neighbourhood graph of the point cloud representing the image in color space, which is filtered by Gaussian density estimation. The resulting algorithm has high computational complexity and requires careful tuning, but shows considerable potential from the viewpoint of general unsupervised machine learning.

*s.high.hopes@gmail.com
†varady@iit.bme.hu

2 Related work

Image segmentation is a long-standing problem with a massive literature, see [30],[22],[1] for thorough surveys on the topic. The methods presented in this paper can be interpreted as an online simplification scheme for the classic watershed transform [36],[31].

Our work is primarily based on the results of Chazal, Guibas, Oudot, and Skraba [4], who presented a very general point cloud clustering algorithm, suitable for data lying in general Riemannian manifolds and a wide class of point cloud density estimators and applied it for color image segmentation and data analysis.

The method we have implemented based on the previous article for color images is in the vein of mean-shift methods [6]. A similar algorithm has also been proposed earlier by Paris and Durant [29] in a less general framework.

Gu, Zheng and Tomasi [17] use Morse-Smale complexes simplified by persistence to segment video data.

The same approach has also been used for the purposes of mesh segmentation, based on curvature estimation [14],[35] and the simulation of heat diffusion [12],[32].

For grayscale segmentation, Letscher and Fritts [21] utilize persistent homology in a different way, by generating a sequence of alpha-complexes from an edge graph and identifying segments with persistent regions in the complements.

Parallels can be drawn between persistence and Total Variation methods [34]. This connection is elaborated from a signal processing viewpoint in [2].

3 Mathematical background

The theory of computational topology depends on a long chain of definitions and results, so most technical details are omitted here. For comprehensive, application-oriented treatments see [37],[15].

Topology is - informally speaking - study of the most general properties of spaces that are invariant with respect to deformations in some precise sense. Algebraic topology uses abstract algebraic structures (vector spaces, groups, rings, etc.) to represent this information in a quantitative way. Homotopy groups are the most natural such objects, consisting of equivalence classes of closed k-spheres (points, circles, spheres, for $k = 0, 1$ and 2 respectively) embedded in the space, with respect to continuous deformations. Although they are very strong and important invariants, their structure is extremely complicated even for simple spaces like the sphere.

By restricting ourselves to *simplicial complexes*[1], i.e. spaces defined as topologically consistent collections of *simplexes* - points, edges, triangles, tetrahedra, etc. (well known examples arising from applications are undirected graphs and triangular or tetrahedral meshes), we can construct the more practical *homology groups* as quotients of *cycles* - 'closed' collection of simplices of a given dimension - by the *boundaries* of sets of higher-dimensional simplices. Thus, homology groups contain equivalence classes of 'loops' whose 'difference' is a boundary of a higher dimensional subset, which is analogous to the intuitive notion of being able to deform them into each other. The rank of the k-dimensional homology group is called the k-th *Betti number* and is, informally speaking, a measure of the number of k-dimensional holes, i.e. topological features of the space. The well-known Euler number is equal to the alternating sum of Betti numbers.

Homology groups are powerful topological invariants [2], but their combinatorial nature also allows for practical algorithms for their computation. The k-th boundary map can be trivially represented as a matrix encoding the adjacency relation between k and $k-1$ simplices in the complex. We can infer on the structure of homology groups using basic linear algebra, by reducing the boundary matrix to the so-called Smith Normal Form.

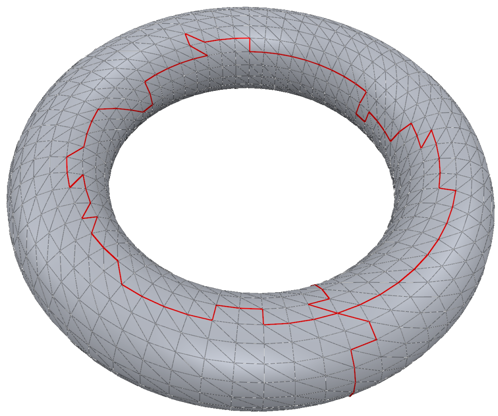


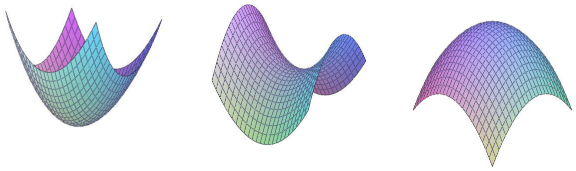
Figure 1: Simplicial torus with generators of the 1st homology group.

When our space has a smooth differentiable structure, the topology of the manifold is closely related to the sublevel-sets of scalar functions defined on it. *Morse theory* [25] makes this connection precise, by stating that by considering the sublevel sets for increasing threshold values, changes in topology can only occur at so-called *critical points*, where the gradient of the function vanishes. A critical point could correspond to a 'hill', a 'basin' or a 'saddle' depending on the definiteness of the Hessian, see Figure 2. The critical points and integral lines between them give a cell decomposition of the manifold called the *Morse-complex*.

The connection between Homology and Morse theory is given by *persistent homology*. Extending the results of

[1]There is a practically equivalent homology theory for cubical complexes.

[2]In general they are considerably weaker than homotopy groups. To see this, consider a loop around the central part of a two-holed torus: this is the boundary of both halves and thus null-homologous, but obviously cannot be shrinked to a point along the surface.



(a) (b) (c)

Figure 2: The 3 types of nondegenerate critical points on a 2-manifold: minima (a), 1-fold saddle (b), and maxima (c).

Morse theory to simplicial complexes, i.e. to a piecewise-linear context is relatively straightforward, and in this case the critical points correspond exactly to changes in the structure of the homology groups. It turns out that only two kinds of change could happen at each step, as we build up our complex: a new homology class (topological feature) can be created, or an existing class can be destroyed (can be made homologous to zero) by adding a new simplex. This allows us to define the persistence of a topological feature as the difference (e.g. in function value or ordering index) between its death and its birth in the construction, and also to organize the critical points in pairs of creators and destructors of topological features. The history of changes in the homology classes thus give a compact multiscale description of the topology, e.g. as a barcode or a so-called *persistence diagram*.

Besides having attractive theoretical properties like stability [5], persistence can also be computed very efficiently. Even in the most general case, only a simple matrix elimination procedure is required to compute persistence information induced by a given simplex ordering, i.e. it can be computed (in worst case[3]) in time cubic[4] in the size of the problem, but empirically the running time is found to be only slightly supralinear for non-artificial data. If we are only interested in connectivity information (by symmetry arguments, this applies to both the 0- and 2-dimensional homology of a 2-dimensional simplex), we could use a Union-Find data structure [11], reducing computational complexity to $O(\alpha(n)n)$, where α is the inverse of the Ackerman function and is less than 5 for any conceivable problem size.

4 Watershed Segmentation with Persistence-based merging

The watershed transformation is a classical tool for image segmentation. The basic idea is very intuitive: consider a porous landscape getting filled by water from the bottom. Lakes start to form at basins, i.e. local height minima, and as the water level rises, two adjacent lakes might get merged when their water reaches a saddle point. By elevating dams along the lines where adjacent lakes 'meet', we get a network of 'watershed lines', and thus a segmentation of the landscape. It can be shown that the lines are (excluding some degenerate cases) always integral lines of the height gradient and connect height maxima with saddles, and each region corresponds to a local height minima/basin. A naive implementation of this idea is highly sensitive to noise for obvious reasons, so oversegmentation is inevitable, making simplification necessary.

By comparing the description of the watershed transform and that of Morse theory and persistence the analogies are quite straightforward. To utilize this connection we adopt the following variant of the abstract algorithm from [4], originally devised as a point-cloud clustering method for the purposes of constructing a simplified watershed segmentation:

1. First, a simplicial complex (e.g. a graph, or a triangular or tetrahedral mesh) is generated from our data.

2. Next, a scalar function is defined on the complex, e.g. at each vertex.

3. We compute a watershed transform sequentially, by processing the vertices by increasing function values [5].

4. If the vertex is a local function minima (every neighbouring vertex has larger function value) it represents a new segment.

5. If a vertex is not a local function minima, it gets added to the component of the neighbour with the smallest function value.

6. If a vertex is adjacent to several components, i.e. it is a saddle, a merging step is performed: if the difference in function value between the saddle and a representative local minima is smaller than some prescribed threshold τ, we merge given segment with the one with the smallest representative local minima.

This algorithm is provably correct, and defines a strict hierarchy of segmentations, guided by a simple merging threshold. Due to the lack of space, detailed pseudocodes and in-depth algorithmic analysis are omitted here; the interested reader shall consult [4] for reference.

We use the *Union-Find* data structure to store and manipulate the hierarchy: the segments are represented as (non-binary) trees, with the ability to find the root of the tree to which a node belongs, and to merge two such trees by connect one's root to the other's. There are two widely used optimizations to the naive implementation: *path compression*, i.e. connecting each node we traverse to the root,

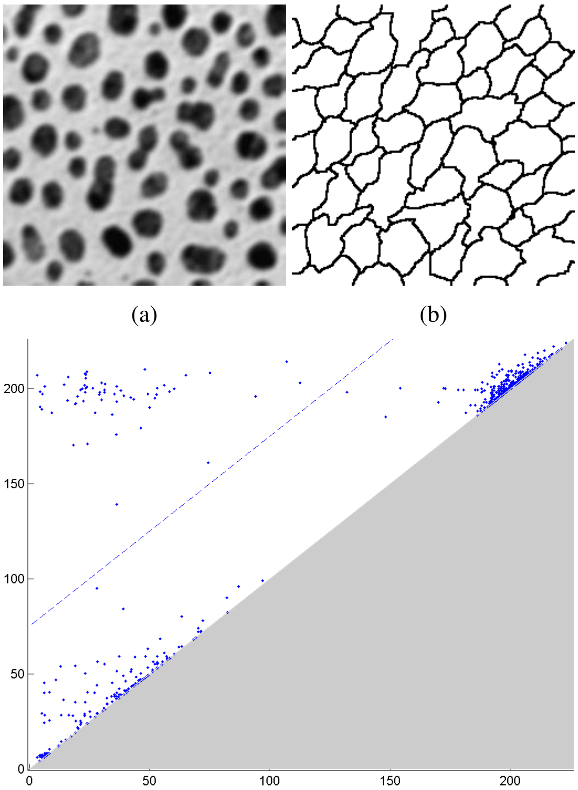
[3]Although this bound is known to be sharp in theory, it is currently known to be achieved only in case of artificial degeneracies. [26]

[4]The exact theoretical bound is that of matrix multiplication, i.e. $O(n^{2.376})$ by current knowledge [7], but the algorithm realizing this asymptotic growth rate is highly impractical.

[5]If segments correspond to local maxima, they are processed by decreasing values and the rest of the algorithm should be modified accordingly.

and *union-by-rank*: always merging the smaller tree into the larger one. Of these, we implement only the former, as our merges shall be carried out in a strict order, which results in a theoretical complexity of $O(n\log(n))$ [8].

By setting the persistence threshold to infinity (or any sufficiently large value, where every segment get merged together) and plotting the time of birth of a segment on the horizontal and its time of death on the vertical axes, we get a scatterplot called the persistence diagram of the image, see Figure 3.



(c)

Figure 3: (a): Grayscale image, (c): persistence diagram, (b): segmentation with persistence threshold $\tau = 75$. Dashed line separates features with shorter or longer lifetimes than the persistence threshold.

5 Grayscale image segmentation

In the context of grayscale images, we consider the 8-connected pixel adjacency graph as a simplicial complex. It is obvious that using the pixel values as a Morse-function gives valid segmentations only in special cases, as it is only capable of isolating darker regions separated by lighter boundaries (or vice versa), see Figure 3. Thus, as it is common with watershed techniques, we use the absolute value of the image gradient instead, computed with a robust variant [19] of the well-known *Sobel-filter*:

$$F_x = \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \\ -2 & -4 & 0 & 4 & 2 \\ -1 & -2 & 0 & 2 & 1 \end{bmatrix}$$

$$F_y = F_x^T.$$

6 Color image segmentation

For color images we have implemented the method of the original article [4], inspired by the so-called mean-shift methods. We first build a nearest neighbour graph from the point cloud representing the image in color space. It is widely known, that Euclidean distances in common color spaces like *RGB* do not correspond well with perceived color differences, so, the image is first transformed into the more suitable *L*a*b** color space. Then, the so-called *Vietoris-Rips graph* is generated by connecting points that are closer to each other than some given parameter δ. The segmentation is produced by clustering the point cloud, i.e. by finding the basins of attraction of its significant local density maxima. Density approximation of point clouds is a well-developed, active research field in statistics and data analysis, with a huge variety of methods available, but for our purposes a simple *truncated Gaussian* estimator was sufficient:

$$f(x) = \sum_{(d(x,p)<h)} e^{\frac{-d^2(x,p)}{2h}}$$

where $d(\cdot,\cdot)$ is the metric defined on the space and $h \in \mathbb{R}$ is called the *bandwidth* of the estimator. Given the graph and the density estimation, the vertices are processed in descending order by the algorithm. As of yet we have considered color information only, which is obviously insufficient to produce meaningful segmentations in a general case. Following the advice of [4] we take spatial proximity into account in the graph construction phase: only those points will be connected that are sufficiently close as pixels in the image. The size of the neighbourhood considered around each pixel gives another degree of freedom to the algorithm.

Graph generation and density estimation both require the computation of pairwise distances and nearest neighbour searches for the point cloud. As these operations are prohibitively expensive to carry out directly, we store the point cloud in a *k-d tree* data structure [27].

7 Implementation and Results

We have implemented the algorithms in C++. For the handling of images and for the nearest neighbour searches in color space we have used the CImg [33] and the ANN [28] libraries respectively.

In the grayscale case, results for selected images in the *Berkeley Segmentation Database* [23] are given in Figure 5, Figure 6 and Figure 7. Segments with cardinality smaller than a given threshold (set to 100 for our experiments, based on trial-and-error) are candidates for further merging, and are painted black.

Figure 8 demonstrates the noise robustness of the grayscale algorithm. Figure 4 illustrates how the segmentation depends on the choice of persistence threshold τ. Runtime results are given for a wide range of image sizes in Table 1.

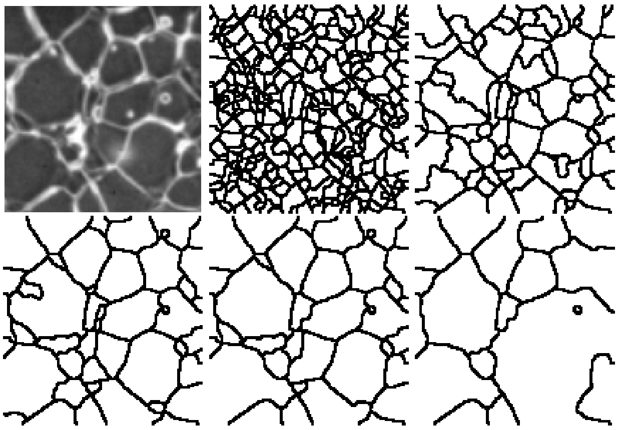
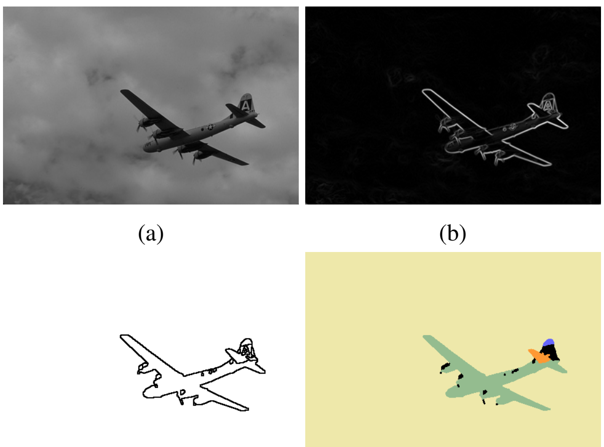
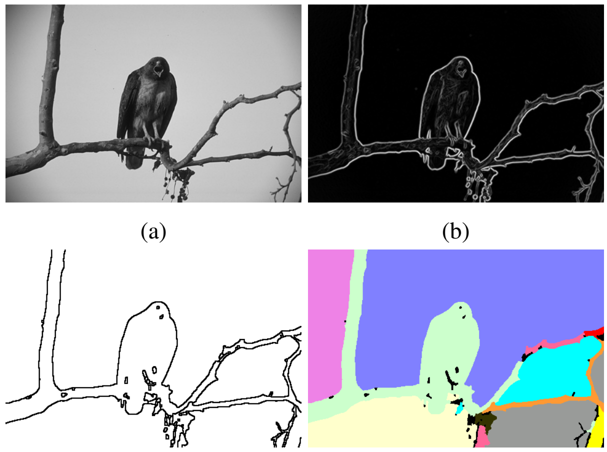


Figure 4: From left to right, and top to bottom: Uranium oxide picture and segmentations for $\tau = 1, 6, 13, 19, 36$.



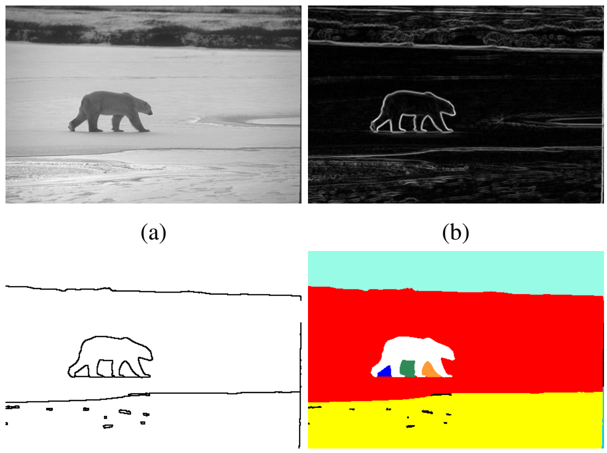
(a) (b) (c) (d)

Figure 5: (a): Image 4096 of the BSD, (b): gradient, (c)(d): segment lines and segment coloring for $\tau = 24$.



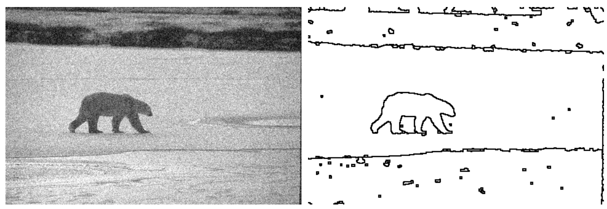
(a) (b) (c) (d)

Figure 6: (a): Image 42049 of the BSD, (b): gradient, (c)(d): segment lines and segment coloring for $\tau = 32$.



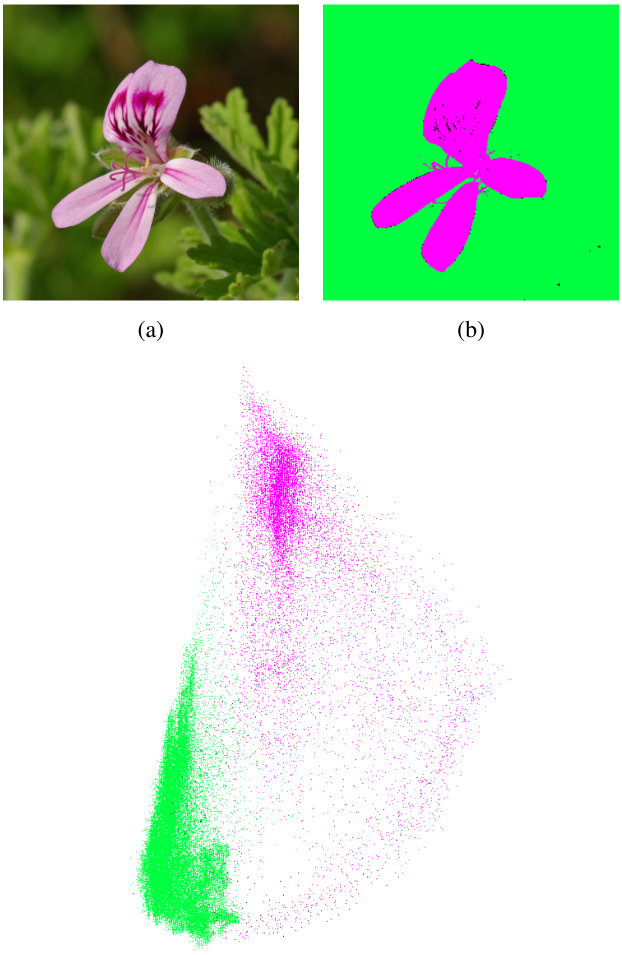
(a) (b) (c) (d)

Figure 7: (a): Image 100007 of the BSD, (b): gradient, (c)(d): segment lines and segment coloring for $\tau = 37$.



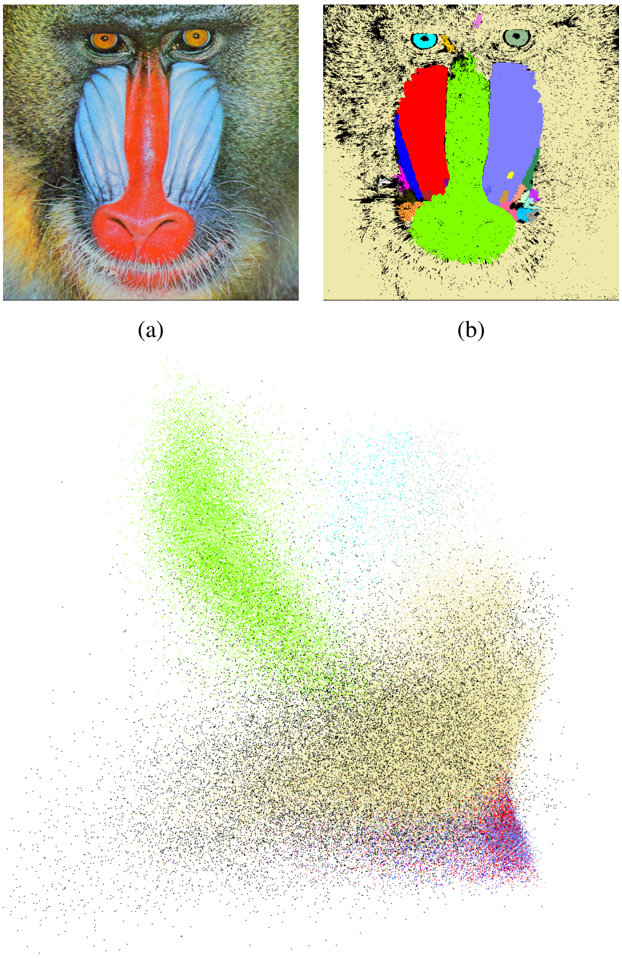
(a) (b)

Figure 8: (a): Image 100007 of the BSD with Gaussian noise of zero mean, 0.01 normalized variance, (b): segmentation with $\tau = 36$. Compare with Figure 7.



(a) (b) (c)

Figure 9: (a): Geranium image, (b): segmentation with $\delta = 8$, $\tau = 4500$, and 7×7 local patch size, (c): clustered point cloud in $L^*a^*b^*$ color space.



(a) (b) (c)

Figure 10: (a): Mandrill image, (b): segmentation with $\delta = 8$, $\tau = 6000$, and 5×5 local patch size, (c): clustered point cloud in $L^*a^*b^*$ color space.

Results for the color method on some common benchmark images are given in Figure 9 and Figure 10.

We note that in our implementations, the persistence diagrams turned out to be of limited use for inferring on the optimal value of the persistence threshold, thus it is necessary to resort to trial-and-error in most cases. In the grayscale case the main reason for this might be that with the very simple gradient filter that we use for edge detection, it is hard to discriminate between significant details and fine texture or noise, which results in very narrow noise margin. It might be possible to overcome this limitation and also improve the overall performance of the algorithm by combining our method with a more intelligent edge detection scheme, e.g the ones considered in [1]. In the color case, the generally ill-defined nature of the image segmentation problem could be be blamed for the lack of an unambiguous separation between signal and noise (this is in contrast with the unsupervised learning and data analysis problems the method have been devised for originally).

The color method is highly experimental in its current stage. The density estimation step could take considerable time (well up to several hours for larger images) and the parameters of the algorithm (persistence threshold, Gaussian bandwidth, graph parameter, local neighbourhood size, etc.) almost always require fine tuning by heuristics.[6]

8 Conclusions

We have presented segmentation algorithms for both grayscale and color images based on the watershed transform and concepts from computational topology.

In the grayscale case the algorithm produces useful results for simpler images even in the very straightforward way we have implemented it, although the threshold parameter requires tuning by trial-and-error for each im-

[6] In our experiments we fixed the Gaussian bandwidth to 25, following the advice of the original article.

Image	Size	Time (ms)
Uranium	141 x 148	13
Blob	256 x 254	31
BSD-42049	481 x 321	73
Lena	512 x 512	145
Emblem	1080 x 1080	440
PET	1800 x 1741	1041
Truck	3753 x 2775	6805

Table 1: Grayscale segmentation timings on an Intel Core2Duo E8400 3.00 GHz, 2GB DDR2 RAM machine, excluding image I/O and gradient computation.

age. We conjecture that much better performance might be achieved by more sophisticated pre- and post-processing.

For color images, persistence-based clustering yields acceptable results if there is a strong correlation between color information and semantics, but requires computationally expensive pre-processing and non-trivial fine tuning of its parameters. It should be noted that the algorithm has been devised as a general point cloud clustering or mode-analysis tool, thus it might be of considerable interest for more general data analysis or machine learning applications.

Acknowledgements

The author is immensely grateful to his advisor, Tamás Várady, for the support and many apt observations. Thanks also to Péter Salvi for his useful suggestions. This work was supported by the Hungarian Scientific Research Fund (No. 101845).

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.

[2] U. Bauer, C. Schönlieb, and M. Wardetzky. Total variation meets topological persistence: A first encounter. In *AIP Conference Proceedings*, volume 1281, page 1022, 2010.

[3] G. Bertrand. On topological watersheds. *Journal of Mathematical Imaging and Vision*, 22(2):217–230, 2005.

[4] F. Chazal, L.J. Guibas, S.Y. Oudot, and P. Skraba. Persistence-based clustering in riemannian manifolds. Technical report, 2009. `http://hal.inria.fr/docs/00/38/93/90/PDF/RR-6968.pdf`.

[5] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[6] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[7] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of symbolic computation*, 9(3):251–280, 1990.

[8] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. The MIT Press, 2001.

[9] M. Couprie, G. Bertrand, et al. Topological grayscale watershed transformation. In *SPIE Vision Geometry V Proceedings*, volume 3168, pages 136–146, 1997.

[10] M. Couprie, L. Najman, and G. Bertrand. Quasi-linear algorithms for the topological watershed. *Journal of Mathematical Imaging and Vision*, 22(2):231–249, 2005.

[11] C.J.A. Delfinado and H. Edelsbrunner. An incremental algorithm for betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12(7):771–784, 1995.

[12] T.K. Dey, K. Li, C. Luo, P. Ranjan, I. Safa, and Y. Wang. Persistent heat signature for pose-oblivious matching of incomplete models. In *Computer Graphics Forum*, volume 29, pages 1545–1554. Wiley-Blackwell, 2010.

[13] Michael Donoser and Horst Bischof. Efficient maximally stable extremal region (mser) tracking. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 553–560. IEEE, 2006.

[14] H. Edelsbrunner. Surface tiling with differential topology. In *Eurographics Symposium on Geometry Processing 2005*, pages 9–11. The Eurographics Association, 2005.

[15] H. Edelsbrunner and J. Harer. *Computational topology: an introduction*. American Mathematical Society, 2010.

[16] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28(4):511–533, 2002.

[17] S. Gu, Y. Zheng, and C. Tomasi. Hierarchical video segmentation using morse-smale complexes. 2009.

[18] M. Henle. *A combinatorial introduction to topology*. Dover publications, 1994.

[19] P. Holoborodko. Noise robust gradient operators. http://www.holoborodko.com/pavel/image-processing/edge-detection/, 2009.

[20] K. Jänich. *Topology*. Springer Verlag, 1984.

[21] D. Letscher and J. Fritts. Image segmentation using topological persistence. In *Computer Analysis of Images and Patterns*, pages 587–595. Springer, 2007.

[22] L. Luccheseyz and S.K. Mitray. Color image segmentation: A state-of-the-art survey. *Proceedings of the Indian National Science Academy (INSA-A). Delhi, Indian: Natl Sci Acad*, 67(2):207–221, 2001.

[23] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[24] Jiri Matas, Ondrej Chum, Martin Urban, and Tomáš Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[25] J.W. Milnor. *Morse theory*, volume 51. Princeton Univ Pr, 1963.

[26] D. Morozov. Persistence algorithm takes cubic time in worst case. In *BioGeometry News, Dept. Comput. Sci., Duke Univ*, 2005.

[27] D. Mount. ANN programming manual: http://www.cs.umd.edu/~mount/ANN/Files/1.1.2/ANNmanual_1.1.pdf. Technical report, 2010.

[28] D. Mount and S. Arya. ANN: A library for approximate nearest neighbor searching: http://www.cs.umd.edu/~mount/ANN/.

[29] S. Paris and F. Durand. A topological approach to hierarchical segmentation using mean shift. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[30] D. Pham, C. Xu, and J.L. Prince. Current methods in medical image segmentation 1. *Annual review of biomedical engineering*, 2(1):315–337, 2000.

[31] J.B.T.M. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41(1-2):187–228, 2000.

[32] P. Skraba, M. Ovsjanikov, F. Chazal, and L. Guibas. Persistence-based segmentation of deformable shapes. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 45–52. IEEE, 2010.

[33] D Tschumperlé. The CImg library: http://cimg.sourceforge.net. *The C++ Template Image Processing Library*.

[34] M. Unger, T. Pock, W. Trobin, D. Cremers, and H. Bischof. Tvseg-interactive total variation based image segmentation. In *British Machine Vision Conference*, volume 2, 2008.

[35] T. Várady, M. Facello, and Zs. Terék. Automatic extraction of surface structures in digital shape reconstruction. *Computer-Aided Design*, 39(5):379–388, 2007.

[36] 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(6):583–598, 1991.

[37] A. Zomorodian. *Topology for Computing*. Cambridge University Press, New York, NY, 2005.