Mathematical Morphology in Computer Graphics, Scientific Visualization and Visual Exploration

Jos B.T.M. Roerdink

Johann Bernoulli Institute for Mathematics and Computer Science
University of Groningen, P.O. Box 407, 9700 AK Groningen, The Netherlands
j.b.t.m.roerdink@rug.nl
http://www.cs.rug.nl/svcg

Abstract. Historically, mathematical morphology has primarily focused on the processing and analysis of two-dimensional image data. In this paper, we survey a number of other areas where mathematical morphology finds fruitful application, such as computer graphics and solid modeling; path planning; filtering, segmentation and visualization of volume data; or visual exploration of high-dimensional data. We also mention techniques for accelerating morphological computations by using graphics hardware (GPU computing).

Keywords: Constructive solid geometry, computational geometry, group morphology, volume processing and visualization, high-dimensional data exploration, GPU computing.

1 Introduction

Historically, mathematical morphology originated as a methodology for processing and analyzing two-dimensional image data. However, its scope turned out to be much wider, being applicable to general multi-dimensional data. In this paper we review a number of such areas which go beyond the image analysis domain. In constructive solid geometry, Minkowski operators are used for modelling and visualization of 3D objects. Group morphology is applicable to path planning and configuration space analysis. Morphological operators have been used for transfer function design in volume rendering. Morphological pyramids and connected morphological operators find application in multiresolution visualization and filtering of (medical) volume data, while for volumetric segmentation morphological active surface models have been proposed.

More recently, mathematical morphology has been applied in visual exploration of high-dimensional data. For example, the watershed transform has been adapted for fast reconstruction and visualization of brain networks; connected filters are used for finding relevant subspaces in high-dimensional scientific data, or for filtering tensor fields such as diffusion tensor imaging data. We also briefly discuss recent techniques for accelerating morphological computations by using graphics hardware (GPU computing).

2 Computer Graphics & Computational Geometry

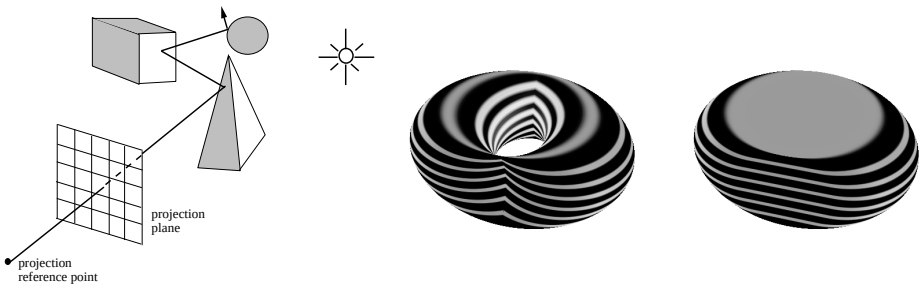

Fig. 1. (a): Set-up for ray tracing of a 3D scene. Visualization of the Minkowski sum of: (b) two perpendicular circles, a quartic surface defined by the equation $(x^2 - y^2 + z^2 + r_z^2 - r_y^2)^2 = 4x^2(r_z^2 - y^2)$; (c) two perpendicular flat discs [35].

2.1 Computer Graphics

In computer graphics various techniques are used to synthesize realistic images of real-world scenes. In Constructive Solid Geometry (CSG), composite solid objects are formed by applying set operations (union, intersection and difference) to simple solids, such as spheres, blocks, or cylinders. Visualization of these objects is possible by incorporating the CSG operations in the ray tracing technique, which visualizes 3D objects by simulating the physical processes of ray propagation, reflection and transmission [11, 13].

Various authors have found the Minkowski addition and subtraction operators to be suitable tools in CSG [30, 40]. Other uses of Minkowski operations in shape description and solid modeling were presented by Ghosh [12]. In special cases the Minkowski addition reduces to the sweep representation (a special case of translation surface) in solid modeling [11, 13].

A different approach was followed in [35]. Using a basic set of elementary shapes, a decomposition was derived of a multiple Minkowski sum of any number of objects, chosen from the basic set, into a union of standard primitives. This union can then subsequently be visualized by standard CSG combined with ray tracing; see Figure 1 for an example. The advantages of this method are efficiency (once the decomposition has been carried out, the ray tracing process is comparatively fast) and compactness of the representation.

2.2 Path planning and configuration space analysis

In path planning the problem is to find a path for an object, say a robot or a car, moving in a space with obstacles. The problem falls apart into two distinct

subproblems [15]. First, the *empty-space problem*: find the allowed states of the robot (moving object); second, the *find-path problem*: find a trajectory in the empty space, subject to certain constraints.

Morphological operations can be used to address the empty-space problem. For robots with only translational degrees of freedom, one can find the allowed positions of the (arbitrarily chosen) center of the robot by a standard erosion of the space outside the obstacles, where the structuring element B is the robot itself. Equivalently, one may perform the dilation by the reflected set $\check{B}$ of the set of obstacles to find the *forbidden* positions of the center of the robot. This is more efficient when the obstacle space is smaller than the space outside the obstacles. If the robot has rotational degrees of freedom or rotating joints, the framework of *group morphology* is appropriate.

Group morphology. The theory of group morphology deals with the construction of morphological operators on an homogeneous space $(\mathbb{T}, \mathcal{X})$, where $\mathbb{T}$ is a group acting transitively on $\mathcal{X}$. For background, see Heijmans and Ronse [17,38] for the case of *abelian* symmetry groups, and Roerdink [31,34] for the case of arbitrary (abelian and non-abelian) symmetry groups; see also [19]. For example, when $\mathcal{X} = \mathbb{R}^d$ any appropriate group $\mathbb{T}$ may be chosen, such as the translation group, the motion group, the affine group, or the projective group, where in each case the morphological operations of interest are invariant under the corresponding group $\mathbb{T}$.

As shown in [32], translation-rotation morphology, where $\mathbb{T}$ is the motion group, is applicable to the empty space problem for robots with rotational degrees of freedom. Another application is the *tailor problem*, which concerns the fitting of sets without overlap within a larger set [33].

An interesting approach to configuration space analysis and similar problems was presented recently by Lysenko *et al.*, who reformulated the framework of group morphology in terms of *group convolution algebras* [27].

2.3 Shape comparison and symmetry detection

Shape comparison is one of the fundamental problems of machine vision. For the case of convex polygons and convex polyhedra shape similarity measures have been studied based on Minkowski addition and inequalities related to the Brunn-Minkowski theory [18,47]. The same theory can be applied for symmetry detection of convex polyhedra, see for example [48]. Also, group convolution algebras have been applied for this purpose [27].

If one considers measures which are not only translation-invariant, but also invariant under the group of orthogonal transformations, the direct computation of similarity measures in the 3D case becomes very time consuming. In principle, optimization should be performed for all possible positions of rotation axes and rotation angles. As shown in [47], for certain measures based on (mixed) volume, it is sufficient to consider only a finite number of "critical" rotations; see also [1]. By using geometric inequalities in the slope diagrams of the polyhedra, the set

of relative orientations to be considered can be narrowed down, so that the time complexity of $O(n^6)$ is reduced to $O(n^{4.5})$ [37].

3 Volume processing and visualization

Volume visualization, or volume rendering, is a technique which produces two-dimensional image representations of three-dimensional data from different viewpoints, using computer graphics techniques such as illumination, shading and colour [16]. Two types of rendering are distinguished: (i) *surface rendering*, where the volume is reduced to one or more isosurfaces $S(c) : f(x, y, z) = c$ of a density function f representing the boundary between materials; (ii) *direct volume rendering*, which maps the volume data directly on the screen, with semi-transparent effects (see Figure 2). Two volume rendering methods which are widely used in

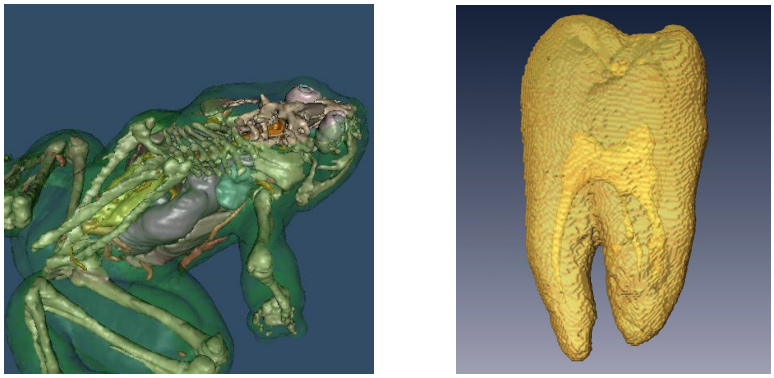

Fig. 2. Left: surface rendering of a frog data set with partially transparent surfaces corresponding to different tissues (data source: the VTK distribution [42]). Right: direct volume rendering of tooth data (data source: the Volume Library `http://www9.informatik.uni-erlangen.de/External/vollib`).

medical imaging are *X-ray rendering* and *maximum intensity projection* (MIP). Here one generates, for each pixel of the view plane, a ray through the data parallel to the line of sight (i.e., perpendicular to the view plane), and assigns either the average or the maximum data value encountered along this ray to the pixel. Because of its computational simplicity and effectiveness, MIP is widely used in the display of magnetic resonance angiography (MRA) and ultrasound data.

Especially when interactive rendering rates are required (i.e., there must be a fast response of the rendering system to actions of the user), volume rendering is a very demanding problem when the sizes of the volume data are large. Two general approaches are available for accelerating the involved computations.

- *Special hardware.* In this category we will briefly discuss general purpose computation on graphics hardware.
- *Special data structures.* Hierarchical structures like wavelets or morphological pyramids are of special interest here.

3.1 General purpose computation on graphics hardware

A recent development which has a major impact on interactive data processing and visualization, is new programmable graphics hardware. The Graphics Processing Unit (GPU), originally used for graphical tasks only, has evolved into a so-called General Purpose GPU (GPGPU). The computing power of GPUs is currently increasing at a faster pace than that of CPUs, so that the GPU is now a major computational device for diverse applications, such as physics simulations, neural networks, image processing and computer vision, graphics and visualization, and even database sorting.

Initially, GPGPU applications, even though not concerned with graphics rendering, did have to use the rendering paradigm, involving the use of textures. As an example we mention the GPU-acceleration of elementary morphological operators as pioneered in the nineties by Hopf and Ertl [20]. However, in 2006 NVidia introduced a programming environment called CUDA [25], which allows the GPU to be programmed through more traditional means. At this moment a dedicated programming effort is still required to develop algorithms that perform efficiently on GPU hardware, but efforts are underway for automatic transformation of CPU programs into GPU counterparts [24].

3.2 Morphological operators for transfer function design

In volume rendering, one of the most difficult tasks is the process of *classification*, that is, to determine for each voxel to what type of material, tissue, etc., it belongs. Classification is usually done by using a *transfer function* that defines the colour and opacity values of each voxel. These are then used when the values of voxels along viewing rays are combined into a single pixel color. Finding a suitable transfer function is often done by an interactive process which can be very complicated and time consuming [16, Ch. 9]. To make this process more automatic, Lürig and Ertl [26] proposed multiscale morphological operators which incorporate spatial neighbourhood information, as an alternative to traditional transfer functions.

3.3 Morphological pyramids for multiresolution visualization

For very large data sets, a multiresolution approach is an obvious choice, which allows a quick visualization of reduced versions of the data that can be progressively refined if needed. For maximum intensity projection (MIP), the transform is nonlinear, so the standard linear multiresolution models based on wavelets

(see, e.g., [50]) are not applicable. Instead, the framework of morphological pyramids as developed by Goutsias and Heijmans [14] can be used as the basis for developing multiresolution algorithms for MIP; see [36] for a survey.

The multiresolution MIP algorithm can be summarized as follows. In the *pyramid analysis* phase, which is a preprocessing step, a 3D morphological pyramid of approximation and detail coefficients is computed by repeated morphological filtering followed by downsampling. The original data at each level of the pyramid can be recovered by successive upsampling and morphological filtering of higher-level data; this process is called *pyramid synthesis*. Particularly suitable for multiresolution MIP are *dilation pyramids*, where the filter operation during synthesis is a dilation. For such pyramids, the MIP operation and the pyramid synthesis work nicely together, in the sense that the maxima along the line of sight can be computed first from pyramid data on a coarse level (where the size of the data is reduced), after which a fast 2D morphological synthesis operator is used to perform reconstruction of the projection image to full grid resolution [36].

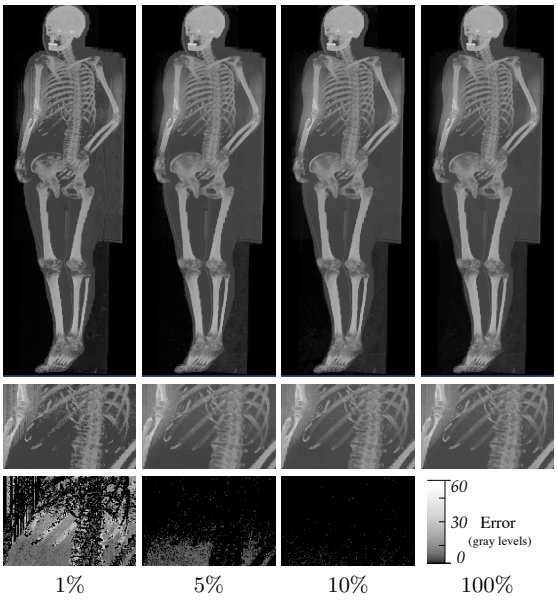

Fig. 3. Streaming MIP-splatting of the complete Visible Woman dataset (source: `http://www.nlm.nih.gov/research/visible`) in a 800 by 2000 window. The rendering is shown at various quality settings (given as percentages of the total number of detail coefficients). The second row shows a detail image for each quality setting, and the third row shows the difference image in gray levels [23].

Several approaches based on pyramid schemes (all of the dilation-pyramid type) were compared in [36]. The most efficient approach was found to be *stream-*

ing MIP-splatting. In this method, detail coefficients from all levels are jointly resorted with respect to decreasing magnitude of a suitable error measure. In the rendering phase, all resorted coefficients are projected successively, until a desired accuracy is obtained. As shown in [36], streaming MIP-splatting outperforms earlier methods based on morphological pyramids, both with respect to image quality with a fixed amount of detail data, and in terms of a flexible trade-off between approximation error and computing time.

Streaming MIP-splatting on the GPU. A GPU implementation of the streaming MIP algorithm was studied in [23]. The load and the dataset can be spread over multiple graphics cards in a straightforward way, thereby achieving support for large volume data with an almost optimal speedup. An example is given in Figure 3. The method achieves interactive frame rates, ranging from 20-50 frames per second, depending on the allowed error.

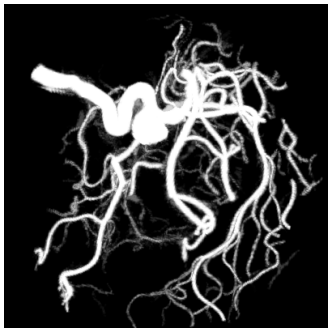

Fig. 4. MIP rendering of the aneurysm data set (rotational b-plane X-ray scan, courtesy Philips Research, Hamburg, Germany; `http://volvis.org`), filtered according to a non-compactness criterion with value $\lambda = 2.0$ [51].

3.4 Connected operators for combined filtering and visualization

Connected filters are based upon an axiomatic definition of connectivity within a complete lattice framework [19, 43]. They are used to perform filtering based on various shape and size attributes. A key property of connected filters is their edge preserving nature. Connected filters can be efficiently computed by the Max-tree data structure, in which the nodes represent connected components for all threshold levels in a data set [41]. The basic Max-tree data structure can be augmented by extensions that allow (i) direct volume rendering, (ii) representation of the Max-tree on graphics hardware, and (iii) fast active cell selection for isosurface generation. In all three cases, the Max-tree representation can be used to change filter parameters interactively and visualize the result at interactive rates [51]; see Figure 4 for an example.

3.5 Segmentation and visualization by active surface models

Segmenting images using active contour models (snakes) involves the evolution of a curve or surface (*i.e.*, an interface), subject to constraints derived from a given input image or volume. State-of-the-art active contours are based on the level set framework, which is able to handle complicated topologies of the underlying shapes [46]. The evolving curve $C(s) : [0,1] \to \mathbb{R}^2$ is given by the zero level-

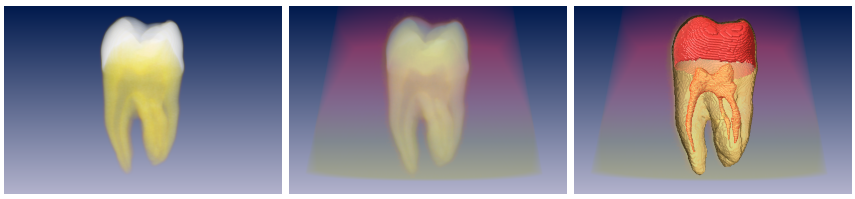

Fig. 5. Segmentation of multiple nested objects. *Left, center*: volume renderings of the tooth data set (cf. Figure 2); *right*: the segmented constituent parts: the enamel, dentin and root canal [21].

set at time t of a function $\phi(x, y, t)$ that satisfies an evolution equation of the form $\frac{\partial \phi}{\partial t} = F\,||\nabla \phi||$. In the context of image segmentation, various formulations for the speed function F have been proposed. Traditionally F is set to some function of the gradient image [6, 28], such that the active contour stops its evolution whenever important edges in the input image are encountered. Chan and Vese [7] used a minimum-variance criterion of the segmented regions. Their active contour model leads to the evolution equation:

$$\frac{\partial \phi}{\partial t} = \left\{\mu \cdot \kappa - \nu + \lambda \left[(I - c_2)^2 - (I - c_1)^2\right]\right\} ||\nabla \phi||\,, \tag{1}$$

which has to be solved for ϕ, with I the input image, κ the level set curvature, $c_1 = \text{average}(I)$ in $\{\phi \geq 0\}$ and $c_2 = \text{average}(I)$ in $\{\phi < 0\}$. The first term in Eq. (1) represents the curvature flow and minimizes the length of the curve, the second term represents inwards motion at constant speed and minimizes the area of the region, whereas the last term represents region competition by the minimum-variance criterion.

As shown in [21], Chan and Vese's minimum-variance model can be reformulated within the context of discrete *multi-scale morphology* [29] as follows:

$$u^{k+1} = \text{sgn}\left(u^k * \chi_{B_p} + \text{sgn}(f^k)\,(|B_p| - 1)\right), \tag{2}$$

where k is the discrete time (scale) parameter, $\text{sgn}(x) = 1$ if $x > 0$ and -1 otherwise, B_p is the unit ball w.r.t. the p-norm, χ_{B_p} the characteristic function of B_p, $|B_p|$ the number of elements of B_p, and '*' denotes linear convolution. The 'speed function' f^k is given by

$$f^k = \lambda\left((I - c_2^k)^2 - (I - c_1^k)^2\right) + \alpha \cdot \text{sgn}(u^k * \chi_{B_p} + \beta), \tag{3}$$

where $\lambda \geq 0$, $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{Z}$.

By varying the free parameters in the definition of f^k, various filters are obtained, such as median filtering, local dilations and erosions, or open-close and close-open filters. Both the PDE-based model and the discrete model were implemented on the GPU in [21]. Experiments showed that the discrete model produces results which are comparable to those of the continuous PDE model based on level sets, while being almost two orders of magnitude faster. An example is given in Figure 5, showing the segmentation of multiple nested objects.

4 Visual exploration of high-dimensional data

4.1 Reconstruction and visualization of brain networks

Electroencephalography (EEG) is a method to measure the electrical activity of the brain by means of electrodes attached to the scalp at multiple locations. Synchronous electrical activity in different brain regions is generally assumed to imply functional relationships between these regions. A measure for this synchrony is EEG coherence, calculated between pairs of electrode signals as a function of frequency.

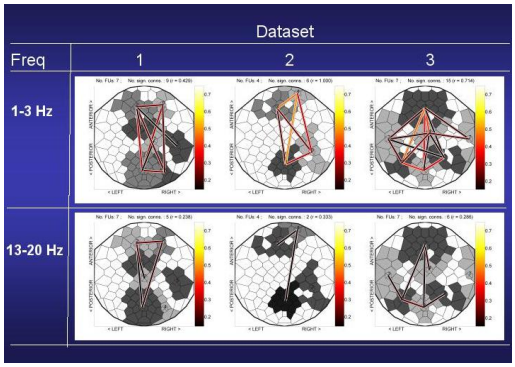

Fig. 6. *Functional Unit maps for multichannel EEG coherence visualization.* Brain responses were collected from three subjects using an EEG cap with 119 scalp electrodes. During a so-called P300 experiment, each participant was instructed to count target tones of 2000 Hz (probability 0.15), alternated with standard tones of 1000 Hz (probability 0.85) which were to be ignored. After the experiment, the participant had to report the number of perceived target tones. To each electrode a Voronoi cell in a graph layout is associated and all cells belonging to a functional unit (FU) have a corresponding color. Lines connect FU centers if the inter-FU coherence exceeds a significance threshold. The color of a line depends on the inter-FU coherence. Shown are FU maps for target stimuli data, with FUs larger than 5 cells, for the 1-3Hz EEG frequency band (top row) and for 13-20Hz (bottom row), for three datasets [5].

A typical data-driven visualization of electroencephalography (EEG) coherence is a graph layout, with vertices representing electrodes and edges representing significant coherences between electrode signals. A drawback of this layout is its visual clutter when the number of electrodes is large. To reduce clutter, ten Caat *et al.* [5] defined a so-called *functional unit* (FU) as a data-driven region of interest (ROI). An FU is a spatially connected set of electrodes recording pairwise significantly coherent signals, represented in the coherence graph by a spatially connected clique. Computing such cliques is very time-consuming: its time complexity is $O(3^{n/3})$, with n the number of vertices.

As an alternative, a modified watershed method (time complexity $O(n^2 \log n)$) was developed, which merges basins representing FUs during the segmentation process if they are spatially connected and if their union is a clique [5]. The modified watershed method produces FU maps which are comparable to the clique-based method, and is up to a factor of 10^5 faster for a typical setting with 128 EEG channels, thus making interactive visualization possible; see Figure 6 for an example. The method can also be extended to find averaged maps for data-driven group analysis. The method was applied to mental fatigue [4] and neurodegenerative disease [8].

4.2 Filtering and visualization of diffusion tensor imaging data

Processing and visualization of tensor fields has become very important over the last decade [49]. A prime application area is medical imaging, where magnetic resonance diffusion tensor imaging (DTI) enables the *in vivo* exploration of the structural organization of fibrous tissue, such as the brain or the heart. An

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

Fig. 7. Illustrative visualization of DTI fiber tracts. (a): Initial set. (b)-(d): Three example stages of filtering, using the fractional anisotropy (FA) value of a DTI fiber tract data set. With the growing filtering threshold for FA, more of the internal structure of the data set is revealed [9].

interesting feature of DTI is its ability to derive local information, such as the amount of anisotropy in a single brain voxel; this can be visualized by tensor glyphs [22], which represent iso-probability surfaces of the diffusion process (in ordinary DTI these are ellipsoids oriented along the main fiber direction). In addition, one can track fiber bundles from a selected brain area. This allows the

determination and visualization of *structural connectivity* between brain regions; see Fig. 7 for an example [9].

For connectivity-based morphological filtering and visualization of tensor fields, new developments in (hyper)connectivity, constrained and partial connectivity are of current interest [3, 39, 44, 45, 52].

4.3 Finding relevant subspaces in high-dimensional data

Data sets in many scientific areas are growing to enormous sizes with high dimensionality. Exploration of such large data spaces poses a huge challenge. *Subspace clustering* is one among several approaches which have been proposed for this purpose in recent years. This method detects subspaces of a high-dimensional space which are potentially "relevant" or "interesting", as defined by specific criteria. The actual clustering can then be limited to the relevant subspaces. Ferdosi *et al.* [10] recently proposed a subspace finding method based on con-

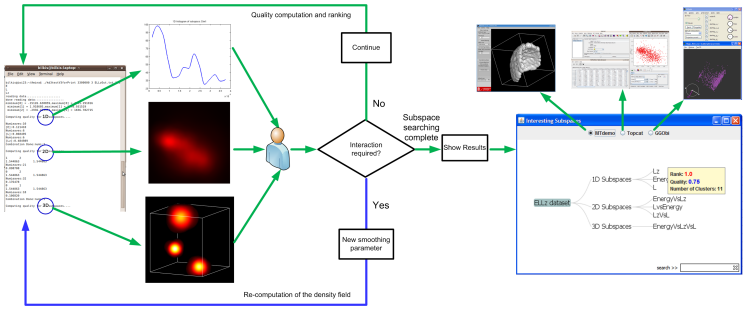

Fig. 8. Schematic diagram of the interactive search and exploration system of high-dimensional data spaces [10].

nected morphological operators. First a transformation is performed from the high-dimensional parametric space to discrete image space where the data are represented by a grid-based density field. Then connected operators are applied on this density field that provides visual support for the analysis of the important subspaces. The importance of a cluster is measured by a quality criterion based upon the notion of *dynamics* [2]. The search for modes/local maxima is done on the Max-tree representation of the density image [41]. For subspaces of dimension higher than three, principal component analysis (PCA) is applied and the first three principal components are used for subspace ranking.

During computation the user can interact with the system to improve the results. In the result stage, three visualization toolkits are used that are linked within a graphical user interface for in-depth exploration of the ranked subspaces; see Figure 8, where the system is used in an astronomical application. Current work involves the extension of this approach to large touch-sensitive displays, which support collaborative research.

5 Conclusions

As is apparent from this brief survey, mathematical morphology is a very versatile methodology, with applications ranging from image processing and computer graphics to data visualization. Built on a solid mathematical foundation, it continues to find new theoretical directions such as (hyper)connected filters, as well as important applications, including tensor imaging and high-dimensional data exploration.

References

1. Bekker, H., Brink, A.A., Roerdink, J.B.T.M.: Reducing the time complexity and identifying ill-posed problem instances of Minkowski sum based similarity calculations. International Journal of Computational Geometry and Applications 19(5), 441–456 (October 2009)
2. Bertrand, G.: On the dynamics. Image Vision Comput. 25(4), 447–454 (2007)
3. Braga-Neto, U., Goutsias, J.: A theoretical tour of connectivity in image processing and analysis. J. Math. Imag. Vision 19, 5–31 (2003)
4. ten Caat, M., Lorist, M.M., Bezdan, E., Roerdink, J.B.T.M., Maurits, N.M.: High-density EEG coherence analysis using functional units applied to mental fatigue. J. Neuroscience Methods 171(2), 271–278 (2008), doi:10.1016/j.jneumeth.2008.03.022
5. ten Caat, M., Maurits, N.M., Roerdink, J.B.T.M.: Data-driven visualization and group analysis of multichannel EEG coherence with functional units. IEEE Trans. Visualization and Computer Graphics 14(4), 756–771 (2008)
6. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proc. 5th Int. Conf. Computer Vision. pp. 694–699 (1995)
7. Chan, T., Vese, L.: Active contours without edges. IEEE Trans. Image Processing 10, 266–277 (2001)
8. Crippa, A., Maurits, N.M., Roerdink, J.B.T.M.: Graph averaging as a means to compare multichannel EEG coherence networks and its application to the study of mental fatigue and neurodegenerative disease. Computers & Graphics 35(2), 265–274 (2011)
9. Everts, M.H., Bekker, H., Roerdink, J.B.T.M., Isenberg, T.: Depth-Dependent Halos: Illustrative Rendering of Dense Line Data. IEEE Transactions on Visualization and Computer Graphics 15(6), 1299–1306 (Nov/Dec 2009)
10. Ferdosi, B.J., Buddelmeijer, H., Trager, S., Wilkinson, M.H.F., Roerdink, J.B.T.M.: Finding and visualizing relevant subspaces for clustering high-dimensional astronomical data using connected morphological operators. In: Proceedings of IEEE Conference on Visual Analytics Science and Technology (IEEE VAST), October 2010. pp. 35–42 (2010)
11. Foley, J.D., Dam, A.V., Feiner, S.K.: Computer Graphics : Principles and Practice. Addison-Wesley, Reading, MA (1990)
12. Ghosh, P.K.: A mathematical model for shape description using Minkowski operators. Comp. Vis. Graph. Im. Proc. 44, 239–269 (1988)
13. Glassner, A.S. (ed.): An Introduction to Ray Tracing. Academic Press, New York (1989)
14. Goutsias, J., Heijmans, H.J.A.M.: Multiresolution signal decomposition schemes. Part 1: Linear and morphological pyramids. IEEE Trans. Image Processing 9(11), 1862–1876 (2000)

15. Gouzènes, L.: Strategies for solving collision-free trajectories problems for mobile and manipulator robots. Intern. J. Robotics Res. 3, 51–65 (1984)
16. Hansen, C.D., Johnson, C.R. (eds.): The Visualization Handbook. Elsevier Butterworth-Heinemann (2005)
17. Heijmans, H.J.A.M., Ronse, C.: The algebraic basis of mathematical morphology. Part I: dilations and erosions. Comp. Vis. Graph. Im. Proc. 50, 245–295 (1989)
18. Heijmans, H.J.A.M., Tuzikov, A.: Similarity and symmetry measures for convex shapes using Minkowski addition. IEEE Trans. Patt. Anal. Mach. Intell. 20(9), 980–993 (1998)
19. Heijmans, H.J.A.M.: Morphological Image Operators, Advances in Electronics and Electron Physics, Supplement, vol. 25. Academic Press, New York (1994)
20. Hopf, M., Ertl, T.: Accelerating Morphological Analysis with Graphics Hardware. In: Workshop on Vision, Modelling, and Visualization VMV '00. pp. 337–345 (2000)
21. Jalba, A.C., Roerdink, J.B.T.M.: An efficient morphological active surface model for volumetric image segmentation. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) Proc. 9th International Symposium on Mathematical Morphology and its Application to Signal and Image processing, Groningen. pp. 193–204 (Aug 24-27 2009)
22. Kindlmann, G.L., Westin, C.F.: Diffusion tensor visualization with glyph packing. IEEE Trans. Vis. Comput. Graph. 12(5), 1329–1336 (2006)
23. van der Laan, W.J., Jalba, A.C., Roerdink, J.B.T.M.: Multiresolution MIP rendering of large volumetric data accelerated on graphics hardware. In: Proc. Eurographics/IEEE VGTC Symposium on Visualization (EuroVis). pp. 243–250 (2007)
24. Leung, A., Lhoták, O., Lashari, G.: Automatic parallelization for graphics processing units. In: Proceedings of the 7th International Conference on Principles and Practice of Programming in Java. pp. 91–100. PPPJ '09, ACM, New York, NY, USA (2009)
25. Lindholm, E., Nickolls, J., Oberman, S., Montrym, J.: NVIDIA Tesla: A unified graphics and computing architecture. IEEE Micro 28(2), 39–55 (2008)
26. Lürig, C., Ertl, T.: Hierarchical volume analysis and visualization based on morphological operators. In: Proc. IEEE Visualization '98. pp. 335–341. IEEE Computer Society Press (1998)
27. Lysenko, M., Nelaturi, S., Shapiro, V.: Group morphology with convolution algebras. In: Proceedings of the 14th ACM Symposium on Solid and Physical Modeling. pp. 11–22. SPM '10, ACM, New York, NY, USA (2010)
28. Malladi, R., Sethian, J.A., Vemuri, B.C.: Shape modeling with front propagation: A level set approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 17, 158–175 (1995)
29. Maragos, P.: Differential morphology and image processing. IEEE Transactions on Image Processing 5(6), 922–937 (1996)
30. Menon, J.P., Marisa, R.J., Zagajac, J.: More powerful solid modeling through ray representations. IEEE Computer Graphics and Applications 14(3), 22–35 (1994)
31. Roerdink, J.B.T.M.: Mathematical morphology with non-commutative symmetry groups. In: Dougherty, E.R. (ed.) Mathematical Morphology in Image Processing, chap. 7, pp. 205–254. Marcel Dekker, New York, NY (1993)
32. Roerdink, J.B.T.M.: Solving the empty space problem in robot path planning by mathematical morphology. In: Serra, J., Salembier, P. (eds.) Proc. Workshop 'Mathematical Morphology and its Applications to Signal Processing', Barcelona, Spain, May 12-14. pp. 216–221 (1993)

33. Roerdink, J.B.T.M.: The generalized tailor problem. In: Maragos, P., Shafer, R.W., Butt, M.A. (eds.) Mathematical Morphology and its Applications to Image and Signal Processing. pp. 57–64. Kluwer Acad. Publ., Dordrecht (1996)
34. Roerdink, J.B.T.M.: Group morphology. Pattern Recognition 33(6), 877–895 (2000)
35. Roerdink, J.B.T.M., Blaauwgeers, G.S.M.: Visualization of Minkowski operations by computer graphics techniques. In: Serra, J., Soille, P. (eds.) Mathematical Morphology and its Applications to Image Processing. pp. 289–296. Kluwer Acad. Publ., Dordrecht (1994)
36. Roerdink, J.B.T.M.: Morphological pyramids in multiresolution MIP rendering of large volume data: Survey and new results. J. Math. Imag. Vision 22(2/3), 143–157 (2005)
37. Roerdink, J.B.T.M., Bekker, H.: Similarity measure computation of convex polyhedra revisited. In: Bertrand, G., Imiya, A., Klette, R. (eds.) Digital and Image Geometry. Lecture Notes in Computer Science, vol. 2243, pp. 389–399. Springer, Wien, New York (2001)
38. Ronse, C., Heijmans, H.J.A.M.: The algebraic basis of mathematical morphology. Part II: openings and closings. Comp. Vis. Graph. Im. Proc.: Image Understanding 54, 74–97 (1991)
39. Ronse, C.: Partial partitions, partial connections and connective segmentation. Journal of Mathematical Imaging and Vision 32(2), 97–125 (2008)
40. Rossignac, J.R., Requicha, A.A.G.: Offsetting operations in solid modeling. Computer Aided Geometric Design 3, 129–148 (1986)
41. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE Transactions on Image Processing 7, 555–570 (1998)
42. Schroeder, W., Martin, K., Lorensen, B.: The Visualization Toolkit: an Object Oriented Approach to 3D Graphics (4th ed.). Kitware, Inc. (2006)
43. Serra, J. (ed.): Image Analysis and Mathematical Morphology. II: Theoretical Advances. Academic Press, New York (1988)
44. Serra, J.: Connectivity on complete lattices. J. Math. Imag. Vision 9(3), 231–251 (1998)
45. Soille, P.: Constrained connectivity for hierarchical image decomposition and simplification. IEEE Trans. Pattern Anal. Mach. Intell. 30(7), 1132–1145 (2008)
46. Suri, J., Liu, K., Singh, S., Laxminarayan, S., Zeng, X., Reden, L.: Shape recovery algorithms using level sets in 2-D/3-D medical imagery: A state of the art review. IEEE Trans. on Inf. Tech. in Biomed. 6, 8–28 (2002)
47. Tuzikov, A.V., Roerdink, J.B.T.M., Heijmans, H.J.A.M.: Similarity measures for convex polyhedra based on Minkowski addition. Pattern Recognition 33(6), 979–995 (2000)
48. Tuzikov, A.V., Sheynin, S.: Symmetry measure computation for convex polyhedra. Journal of Mathematical Imaging and Vision 16(1), 41–56 (2002)
49. Weickert, J., Hagen, H. (eds.): Visualization and Processing of Tensor Fields. Springer, Berlin (2006)
50. Westenberg, M.A., Roerdink, J.B.T.M.: Frequency domain volume rendering by the wavelet X-ray transform. IEEE Trans. Image Processing 9(7), 1249–1261 (2000)
51. Westenberg, M.A., Roerdink, J.B.T.M., Wilkinson, M.H.F.: Volumetric attribute filtering and interactive visualization using the Max-tree representation. IEEE Trans. Image Processing 16(12), 2943–2952 (2007)
52. Wilkinson, M.H.F.: Attribute-space connectivity and connected filters. Image and Vision Computing 25, 426–435 (2007)