

A Multilevel Segmentation Method

M. Droske, T. Preußner, M. Rumpf

Institut für Angewandte Mathematik, Universität Bonn
Wegelerstr. 10, 53115 Bonn
[droske | tpreuss | rumpf]@iam.uni-bonn.de

Abstract

Segmentation is an essential ingredient in a wide range of image processing tasks and a building block of many visualization environments. Many known segmentation techniques suffer from being computationally exhaustive and thus decreasing interactivity, especially when considering volume data sets. Multilevel methods have proved to be a powerful machinery to speed up applications which incorporate some hierarchical structure. So does segmentation when considered on quadtree respectively octree data sets. Here we present a new approach which combines a discrete and a continuous multilevel segmentation model.

In figure 1 four different grid segments are depicted which resulted from the multilevel segmentation by using combinations of different boundary indicators which will be described later in detail. At first, the discrete method enables a fast segmentation depending on possibly multiple parameters describing the segment boundary and on selected seed points inside a segment. In an interactive process the user is able to adjust seed points which steer the automatic discrete segmentation process. Furthermore fast multilevel splatting techniques simultaneously enable interactive frame rates in the visualization to validate the obtained results. Thus, the user is effectively supported in the selection of appropriate parameters for the segmentation. Once an acceptable voxel discrete approximation is found a second segmentation and smoothing method based on a continuous model comes into play. It can be regarded



Figure 1: Segmentation Results by greyscale and gradient magnitude thresholding and the exclusion indicator $\sigma = -\chi_{G^c}$ to prevent the algorithm to grow into Lena's hat. Efficiency: 0.152.

as a suitable postprocessing step. Hence, solving an appropriate diffusion problem the boundary approximation of the already obtained segment is improved including a suitable tangential smoothing.

1 Introduction

Multilevel techniques in scientific computing [14, 2, 3], computer vision [1, 31] and visualization [17, 25, 13, 20, 12] have become successful approaches to overcome computational capacity problems and to achieve fast processing times up to interactive performance. Especially when considering large

three-dimensional data sets, adaptivity and multiresolution techniques can drastically increase the interactivity of many applications [5, 7].

One of the fundamental tasks in computer vision is to automatically detect connected visually perceptible objects – segments – underlying some mathematical segmentation model. Afterwards, based on the segmentation either an appropriate visualization or quantitative respectively qualitative measurements can be considered and the knowledge on segments can be used as an essential input for physical process simulation.

The underlying model for our segmentation is to find from a given set A the region of all points in a 2D or 3D domain Ω that can be connected continuously by a curve to A along which a given homogeneity condition is true. This condition is determined by particularly thresholding a general function on the spatial domain which indicates the boundary. We denote this *boundary indicator* by $\sigma : \Omega \rightarrow \mathbb{R}$. Whenever σ becomes non-negative homogeneity is lost. Typically such boundary indicators represent derived quantities from given functions on the domain, which can be physical quantities or simply some grayscale ($d = 1$) or color valued ($d > 1$) image intensity $I : \Omega \rightarrow \mathbb{R}^d$. The homogeneity need not necessarily be considered as low variation of the intensities of the data. The number of connected components of the segmentation is less or equal the number of connected components of A . By defining the starting set A , not necessarily a single seed point, we can permit our algorithm to grow in several disjunct regions simultaneously which may or may not join during the segmentation process or to consider some coarser inner approximation of the true segment as starting set. A boundary indicator can be interpreted as an indicator of regions where homogeneity is lost. Lateron, we will discuss how to combine different homogeneity criteria. At first we will developed a hybrid formulation, allowing arbitrary combinations of different homogeneity criteria.

There are several ways of defining homogeneity, many of them are based on the detection of image edges, that represent the boundaries of the segment, calculated for example by evaluating the image gradient or the zero-crossings of the Laplacian [22] or the local maximum of the gradient along its own direction. The latter technique is known as Canny-edge-detection [9]. As most edge-detection techniques are very sensitive to image noise, the segmentation results can be improved significantly by an effective noise reduction filter. In the context of anisotropic diffusion, new robust noise filters have been presented recently, which produce very good results in terms of edge preservation and coherence promotion [31, 6, 26].

In what follows let us be more precise in the formulation of the problem setting and introduce some basic notation. We define an abstract segment $\mathcal{S}(A, \sigma)$ with starting set A depending on the indicator function $\sigma : \Omega \rightarrow \mathbb{R}$ by

$$\mathcal{S}(A, \sigma) := \left\{ y \in \Omega \mid \exists C^0 \text{ curve } \gamma : [0, 1] \rightarrow \Omega, \right. \\ \left. \gamma(0) \in A, \quad \gamma(1) = y, \right. \\ \left. \sigma(\gamma(t)) < 0 \quad \forall t \in [0, 1] \right\}.$$

Given a scalar valued image intensity I , important examples of boundary indicators σ are the following:

- $\sigma = -\chi_{[e_-, e_+]} \circ I$ (image magnitude interval)
- $\sigma = \|\nabla I\| - M$ (image gradient threshold)
- $\sigma = -\chi_{\mathcal{S}(\tilde{A}, \tilde{\sigma})^c}$ (other segment boundaries)
- $\sigma = \lambda_{\text{crit}} - (K \circ I)(\cdot)$ (local filter results, for instance wavelet filters, which identify local frequencies of a certain intensity)

These indicator functions may be composed to build new boundary indicator functions. E.g. this can be done by

- linear combination $\sigma = \sum_{i=1}^d \omega_i \sigma_i$ steered by weights ω_i , or
- by the intersection of two segments which corresponds to the choice $\sigma = \max\{\sigma_1, \sigma_2\}$.

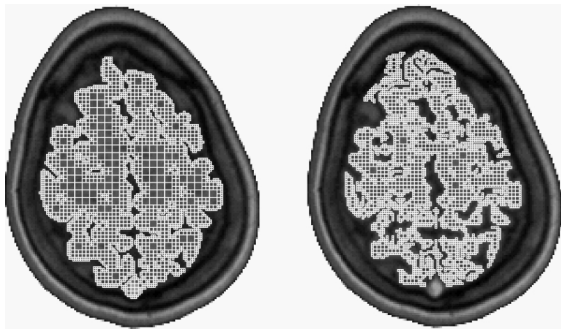


Figure 2: A comparison of a grey value indicator (left) and a gradient threshold indicator (right) on a 2D slice of an MR-scan.

So far our segmentation model considers continuous underlying images or data sets. Let us now focus on a discrete data base. Thus, we interpret our data as piecewise constant of bilinear, respectively trilinear on a regular cell subdivision of our domain. Then a segmentation algorithm turns out to be quite simple, when we only operate on the finest grid, whose nodes represent the pixels or voxels of the image. From the starting set A we just have to apply a standard recursive filling algorithm by testing at each voxel or pixel (cell) the boundary indicator. When we reach a cell with $\sigma \geq 0$ we know that we have reached the boundary of the segment and stop growing from this point on. We recognize that the filling algorithm has to traverse through every cell contained in the segment and therefore has an execution time of $\mathcal{O}(n^d)$ if A is a single cell and $\mathcal{S}(A, \sigma)$ is a “thick” set of diameter n , measured in cells. With respect to interactive response for the user’s segmentation queries with varying segmentation parameters this is much too expensive.

We will present a discrete algorithm on a quadtree, respectively an octree which significantly increased performance. In a pre-computing step we once evaluate error indicators corresponding to the physical functions on which the definition of the boundary indicator is based. Afterwards, in the user guided process of seed point selection and parameter adjustment for the selected single or combined boundary indicator we can ensure interactive

response times. A further advantage of our method is that we can adjust the level of detail, respectively the depth of the hierarchical grid on which we want to extract an approximation of the actual discrete segment first.

All query results obtained so far are sets of voxels on a certain grid level. Cells are either completely inside or completely outside a segment. This leads to non smooth segment boundaries and requires an improvement to guarantee a proper visual perception and still to respect the given boundary indicator. Thus, we consider the characteristic function of the obtained, proper discrete segment as initial value for a diffusion process proposed first by Weickert [31], which incorporates tangential smoothing on a continuous segment boundary. We apply a finite element discretization based on highly graded meshes for the efficient solution of this diffusion problem. Finally this second method can be regarded as a postprocessing of the previously selected discrete segments and it delivers a new segment which is now nicely shaped by a smooth level set, without destroying its correspondence to the actual segment.

2 Review of related work

In computer vision literature various methods dealing with segmentation and feature extraction are discussed. The well known technique of the morphological watershed transform [23] creates a tessellation of the image domain Ω in several small regions by considering the image values as intensity niveaus in a topographical landscape. By simulating rainfall, the domain is grouped in catchment basins, regions in which the water drains from all points to the same local intensity minimum. Naturally this method is very sensitive to small variations of the image magnitude and consequently the number of generated regions is undesirably large. To overcome this problem of identifying exhaustively many segments there have been efforts in recent years to reduce the complexity of the tessellation by region merging based on homogeneity criteria [15] or studying the

evolution of the catchment basins in Gaussian scalespace [10]. Such techniques can generate unpredictable results and depend to a large extent on user interaction and the quality of the initial partition. Although improvements have been made [30], the creation of the watersheds is still computationally demanding.

An entirely different and popular approach to visual shape analysis is related to so called *active contour models* and *snakes* [31, 21, 32, 8]. It is based on a curve respectively surface evolution, starting from some initial curve or surface which is propagated to achieve a proper approximation of the segment boundary. Active contour models incorporate a wide range of driving forces. Many of them are based on minimization of combined energy functionals controlling the fairness of the resulting curve on one hand and the attraction to areas of interest such as object boundaries on the other hand. Weighting parameters have to be carefully chosen to find a good balance between these terms. In early works *explicit snakes* with a standard parametric curve representation were used. The key disadvantage of this method is a topological constraint: the curve can not split to approximate boundaries of not simply connected segments. Such problems have been solved by introducing *implicit snakes* models [8, 21], in which the initial curve is interpreted as the zero level curve of a function $\Phi(t, \cdot) : \Omega \rightarrow \mathbb{R}$. The evolution of these snakes is controlled by a PDE. An external term is considered to include information about the initial image. Although contours are able to split in this formulation there remains the problem that the result of the segmentation relies significantly on a good initialization. Furthermore many models have difficulties in progressing into boundary concavities. Addressing these particular problems a new class of external forces has been proposed by deriving from the original image a *gradient vector flow* field in a variational framework [32]. Sensitivity to initialization has been drastically reduced and contours have a more sensible behaviour in the regions of concavities.

Furthermore a general variational framework for Mumford-Shah and Geman type functionals [24, 11] has been introduced [16]. Edge boundaries are represented by continuous function, yielded by the minimization of an energy functional.

3 Discrete multilevel segmentation

Let us suppose an intensity function, respectively a derived boundary indicator to be given on a $m \times n$ grid. Furthermore let us assume we can interpret our discrete domain as a hierarchical grid, i.e. a quadtree or an octree respectively. At least it can be embedded in some regular $(2_{\max}^l)^d$ grid, e.g. filling the superficial cells with some background intensity. Instead of some process solely on the finest grid level which successively visits all fine grid cells inside the segment, our aim is to detect coarse elements inside the segment in the hierarchy of nested grids. By adding such a elements we implicitly add all its child cells at the same time and thus increase the algorithms performance. Let us denote by $\mathcal{M} := \{\mathcal{M}^l\}_{l=1, \dots, l_{\max}}$ the family of nested grids, each consisting of elements or cells E such that $\bigcup_{E \in \mathcal{M}^l} E = \Omega$ for all $l = 1, \dots, l_{\max}$. They are supposed to be nested in the sense that for all $E^{l+1} \in \mathcal{M}^{l+1}$ there exists an $E^l \in \mathcal{M}^l$ such that $E^{l+1} \subset E^l$. Instead of checking the segment boundary criterion σ for each of the child elements contained in the element $E^l \in \mathcal{M}^l$, we introduce a levelwise *saturated* segment indicating function

$$\sigma_h : \bigcup_{l=1, \dots, l_{\max}} \mathcal{M}^l \rightarrow \mathbb{R} \quad ,$$

which replaces the continuous σ on every cell in the hierarchy. Here we proceed in analogy to the saturation approach in [25] based on grid nodes. Instead of error indicators on nodes we now deal with values on grid cells. Hence saturation of σ_h means, that $\sigma_h(E) \geq \sup_E \sigma$ for each $E \in \mathcal{M}$ in order to construct a robust indicator.

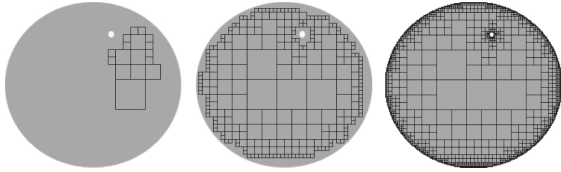


Figure 3: As illustrated, saturation implies robustness, thus small objects will be detected.

To define an analogue for our segment formulation we need the following notation. For each element $E \in \mathcal{M}$ we define $\mathcal{P}(E)$ and $\mathcal{C}(E)$ to be the parent element respectively the set of child elements of E . We denote by $\text{adj}(E)$ the set of adjacent cells of E . We now focus on σ_h as the only function and define a discrete segment as

$$\begin{aligned} \mathcal{S}_h(A, \sigma_h) := & \{ E \in \mathcal{M} \mid \exists (E_i)_{i=1 \dots m} \\ & E_1 \in A, \quad E_m = E, \quad E_{i+1} \in \text{adj}(E_i), \\ & \sigma_h(E_i) < 0, \text{ for } i = 1, \dots, m \}. \end{aligned}$$

In order to construct an efficient growing algorithm we shall collect cells as coarse as possible to minimize the number of adjacent cells needed to build a path from a seed cell to the segment boundary. We measure efficiency by the ratio of the cells visited on the grid hierarchy compared to the number of pixels/voxels which belong to segment. We now have to give a constructive definition for the discrete indicator function σ_h given the continuous σ . As an obvious choice we obtain $\sigma_h(E) := \sup_E \sigma$, which is naturally saturated by the definition above, but it is clear in this case, that we have no knowledge about the difference of grid levels of neighboring cells in our hierarchy.

To ensure that our grid hierarchy has only one-level transitions between neighboring cells - which turns out to be preferable concerning the algorithm - we require the following properties:

$$\begin{aligned} \sigma_h(E) &\leq \sigma_h(\mathcal{P}(E)) \quad \text{for all } E \in \mathcal{M}, \\ \sigma_h(E) &\leq \sigma_h(\mathcal{P}(\tilde{E})) \quad \text{for all } \tilde{E} \in \text{adj}(E). \end{aligned}$$

Thus we begin with the following pre-roll step to obtain an upwards saturated indicator $\bar{\sigma}_h$

(see below how to proceed in case of parameter dependent multiple segment indicator functions):

```

for each  $E \in \mathcal{M}^{l_{\max}}$  do  $\sigma_h(E) := \sup_E \sigma$ 
for  $l = l_{\max}-1$  to  $0$  step  $-1$  do
  for each element  $E$  of  $\mathcal{M}^l$  do
     $A := \mathcal{C}(E) \cup \text{adj}(\mathcal{C}(E))$ 
     $\bar{\sigma}_h(E) := \max(\sigma_h(E), \max_{\tilde{E} \in A} \sigma_h(\tilde{E}))$ 

```

Analogously we introduce for later usage a downwards saturated indicator $\underline{\sigma}_h$.

So far we have only considered a single parameter-independent indicator function. We observe, that combinations of saturated indicators are also saturated, therefore we can combine our precalculated discrete indicators in an efficient way. Hence we only need to saturate each single indicator once on startup and can interactively change parameters and choose different combinations on the fly. Let us study this in more detail for two significant cases:

- To incorporate the grey value indicator $-\chi_{[\varrho_-, \varrho_+]} \circ I$ we need an upwards saturated indicator $\bar{\sigma}_h^I$ and a downwards saturated indicator $\underline{\sigma}_h^I$ both initialized with discrete approximations of the original image intensities I . Now we can check the intersections for the element E by testing

$$(\bar{\sigma}_h^I(E) \leq \varrho_+) \wedge (\underline{\sigma}_h^I(E) \geq \varrho_-).$$

- The image gradient indicator $\|\nabla I\| - M$ is incorporated by an upwards saturated $\bar{\sigma}_h^{\text{grad}}$ which is initialized on the finest grid with some approximation of the gradient magnitude and checked by

$$\bar{\sigma}_h^{\text{grad}}(E) \leq M.$$

- In case of weighted combinations of segment indicators the saturated indicator is obviously the weighted combination of the corresponding primal indicators in case of positive weights.
- Finally we have to assemble these indicators into one saturated negative characteristic function $\sigma_h : \mathcal{M} \rightarrow \{-1, 0\}$, by

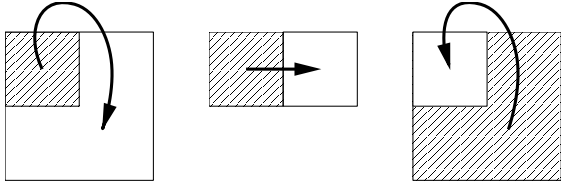


Figure 4: Three possible local configurations which show up in the algorithm. From left to right: element coarsening, shifting, refining. The candidate cell is shaded.

asking for any desired intersection and combination of the indicators described above. This function is used in the algorithm to test the grid cells during the segment growing process. As explained above, this function can be easily adjusted without requiring another saturation.

Thus, based on the required precomputation of saturated segment indicators we can afterwards in the interactive segmentation process reuse these saturated values on the cells for robust and fast response while selecting suitable parameters.

Starting with the finest grid $\mathcal{M}^{l_{\max}}$ we can now recursively coarsen elements $E \in \mathcal{M}^{l_{\max}}$ by checking the indicator $\sigma_h(E) < 0$ for each element. The result is an adaptive grid, which is fine along boundaries and coarse in areas of candidate inner segment cells, where the indicator function is < 0 . However, in the concrete implementation we do not need *a-priori* knowledge of the entire grid during the segmentation algorithm, but grow in all directions locally from the already segmented area starting with the set of seed cells A . Thus only parent or neighboring elements are evaluated in each step. The one-level transitions ensure that only three cases are relevant by growing from one element to an adjacent cell, namely as illustrated in figure 4, a neighboring element may be coarsened, have the same level or be refined. The priority is to coarsen elements as soon as possible.

We will now present the formulation of the algorithm, with a given a saturated boolean indicator σ_h as described above.

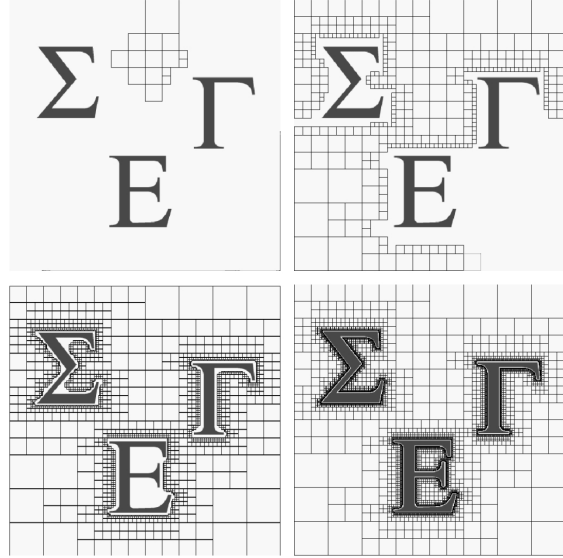


Figure 5: For thin, delicate structures, we can get a coarse approximation, by starting a segmentation process of the complement. Efficiencies from left to right: 0.0049, 0.0109, 0.027953, 0.058.

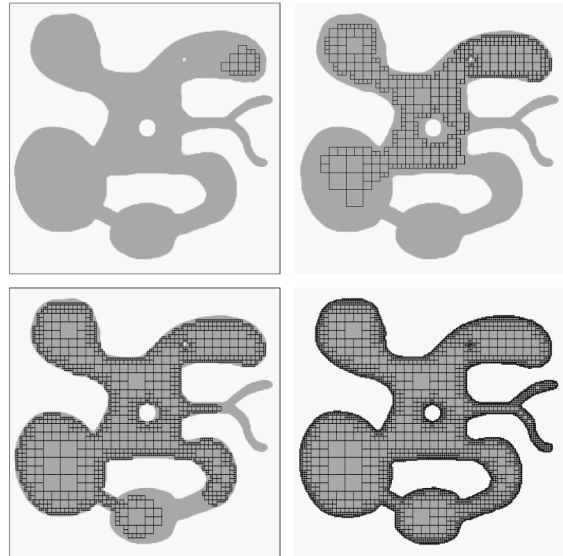


Figure 6: In this series of successive segmentation steps it can be clearly observed, that priority lies on growing in areas of coarse grid cells. At the final stage, the small detailed structures are processed.

```

 $\mathcal{F} := A; \mathcal{S} = \emptyset;$ 
do {
  take coarsest  $E$  from  $\mathcal{F}$ 
  if  $\sigma_h(E) < 0$  {
     $\mathcal{S} = \mathcal{S} \cup E$ 
     $\mathcal{F} = (\mathcal{F} \cup \mathcal{P}(E) \cup \text{adj}(E)) \setminus \mathcal{S};$ 
  } else {
     $\mathcal{F} = (\mathcal{F} \cup \{\hat{E} \in \mathcal{C}(E) \mid \hat{E} \cap \mathcal{S} \neq \emptyset\}) \setminus \mathcal{S};$ 
  }
} while ( $\mathcal{F} \neq \emptyset$ );

```

where \mathcal{F} is the set of candidate cells, \mathcal{S} the union of segmented cells. The selection of the coarsest element out of \mathcal{F} can be efficiently implemented by separating \mathcal{F} levelwise, i.e. we have $\mathcal{F}^l, l = 0, \dots, l_{\max}$, so one of the coarsest elements can be quickly extracted by searching in increasing order the sets \mathcal{F}^l and pick one element as soon as $\mathcal{F}^l \neq \emptyset$. The segment \mathcal{S} can be represented as a levelwise bit field. During the filling algorithm at level l we might eventually collide with a cell which already belongs to the segment or at least has already been checked, so we do not have to grow anymore in this direction. Because we have one-level transitions, we just have to check for neighbors on the three different grid levels $l - 1, l, l + 1$.

4 Continuous segmentation and post processing by anisotropic diffusion

In comparison to the discrete segmentation algorithm, we will briefly outline a continuous watershed algorithm. The region growing process of the segment is simulated by linear diffusion, with a diffusion coefficient that depends on local image properties similar to the boundary indicator criteria. A diffusive watershed algorithm consists of solving the following diffusion problem:

$$\begin{aligned}
\partial_t \rho - \operatorname{div}(A(\sigma) \nabla \rho) &= f(\rho) && \text{in } \mathbb{R}^+ \times \Omega, \\
\rho(0) &= \rho_0 && \text{on } \Omega, \\
\partial_\nu \rho &= 0 && \text{on } \mathbb{R}^+ \times \partial\Omega,
\end{aligned}$$

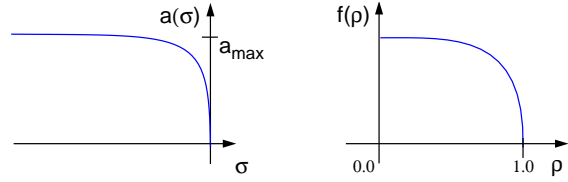


Figure 7: The diffusion coefficient and the right hand side in the continuous segmentation algorithm.

where ρ_0 is the initial distribution, a positive function with compact support in the set A . Thereby the diffusion coefficient $A(\sigma)$ is positive inside the segment and vanishes at the segment boundary. Now the segmentation function can be considered as a continuous function – no longer piecewise constant on fine grid cells as in the case of σ_h above. Considering in addition an anisotropic diffusion model [31, 27], tangential smoothing along the segment boundary can be incorporated.

We solve this problem numerically by applying a bilinear, respectively trilinear conforming finite element discretization on an adaptive quadrilateral, respectively hexahedral grid. In time a semi-implicit second order Euler scheme is used. As it has become standard [4, 19] the scheme is semi-implicit with respect to the evaluation of the nonlinear diffusion coefficient g and the right hand side. In the i th timestep we have to solve the linear system

$$(M + \tau L) \vec{\rho}^i = M \vec{\rho}^{i-1} + F,$$

where $\vec{\rho}^i$ is the corresponding solution vector consisting of the nodal values, τ the current timestep, M is the lumped mass matrix, L the weighted stiffness matrix and F the vector representation of the right hand side. The stiffness matrix and the right hand side are computed by applying the midpoint quadrature rule.

The above linear system is solved by a multigrid method suitable for problems with jumping coefficients.

5 Interactive visualisation aspects

During the interactive stage of experimenting with different control parameters and weights of the indicating functions the visualization has to be fast and meaningful enough to give a good impression of the final result. The segment is visualized by drawing the outer elements, which still will be $O(n^2)$. Instead of drawing all boundary faces of each boundary cell, the number of graphic primitives to be drawn can be significantly decreased by using *texture-splats*. Here we precomputed a $m \times m$ -texture for some fixed m representing a ball shaded under an arbitrary lighting model. Then drawing it for each cell at the smallest possible radius such that the cell is entirely included in this ball, the shading of the ball leads to an approximative shading of the entire surface. Once the user is satisfied with his choice of parameters the segment may be stored as a characteristic function for later continuous segmentation improvement and tangential smoothing, or to extract iso-surfaces.

Figure 8 shows the results of the segmentation of a ball having radius $1/2$ and holes along the coordinate axes. The results of the succeeding tangential smoothing and isosurface extraction is shown in figure 9.

In figure 10 we have segmented the cortex of a human brain from a 3D MRI data set. The segmentation criterion was solely gray value based.

6 Conclusions

We have presented a multilevel segmentation algorithm which ensures interactive response times in a user guided inspection especially of large 3D data sets. Based on a priori saturated indicators for image intensities a robust estimation of derived boundary indicator functions on a underlying hierarchical grid is ensured. This allows to classify preferably coarser cells of the hierarchical grid as being

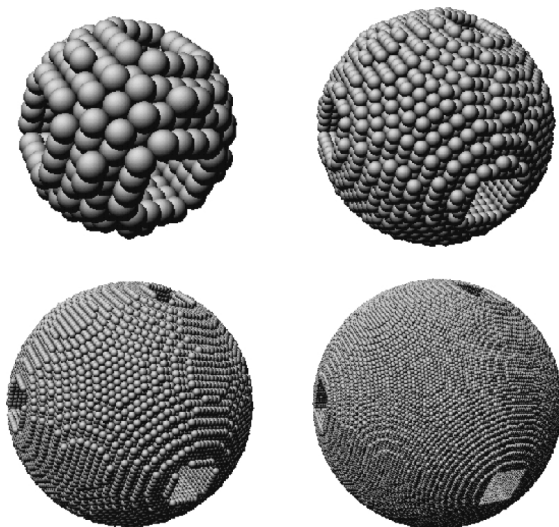


Figure 8: Segmentation efficiencies at different levels. From top left to bottom right: 0.0002, 0.001, 0.0042, 0.017. Final efficiency: 0.05 (see figure 9).

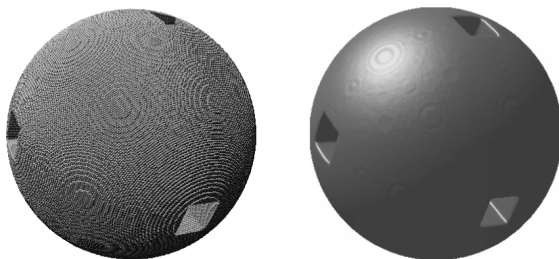


Figure 9: Final segmentation on the finest level (left) and tangential smoothing (right) of the object segmented in figure 8.

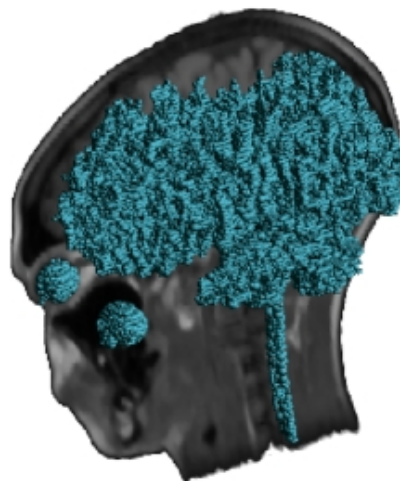


Figure 10: A final segmentation of the brain and the eyes together with a slice through an MRI volume data set.

contained in the segment set without traversing all its finer child cells. Thus, an interactive adjustment of seed points and indicator parameters in the search for appropriate parameters characterizing interesting segment sets is possible. Multilevel splatting techniques for a fast preview support the actual segmentation algorithm. Once a suitable set is extracted its approximation properties and the smoothness of the boundary shape are improved solving a diffusion problem which especially incorporates tangential smoothing on the segment boundary surface. The algorithms were tested on typical test data sets as well as on real images. Future research perspectives are

- to derive useful combined indicator functions, with an intuitive meaning for the user,
- to consider more than one image intensity overlaying 3D images of the same data set but obtained by different methods, e.g. CT or MRT, and
- to carry over the same multilevel methodology to the solution of generalized distance map problems by the level set method, which delivers additional useful information on distances.

References

- [1] S. T. Acton. Multigrid anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3), 1998.
- [2] R. E. Bank, T. Dupont, and H. Yserentant. The hierarchical basis multigrid method. *Numer. Math.*, 52:427–458, 1988.
- [3] D. C. Banks. Illumination in Diverse Codimensions. *Computer Graphics Annual Conference Series*, pages 327–334, 1994.
- [4] E. Bänsch and K. Mikula. A coarsening finite element strategy in image selective smoothing. *Preprint, Universität Freiburg*, 18, 1996.
- [5] P. Bastian, K. Birken, K. Johannsen, S. Lang, N. Neuss, H. Rentz-Reichert, and C. Wieners. Ug - a flexible software toolbox for solving partial differential equations. *Comput. Visual. Sci.*, 1:27–40, 1997.
- [6] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3), 1998.
- [7] Bornemann, F. and Erdmann, B. and Kornhuber, R. Adaptive multilevel methods in three space dimensions. *Int. J. Numer. Methods Eng.*, 36, No.18:3187–3203, 1993.
- [8] V. Caselles, F. Catté, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numer. Math.*, 66, 1993.
- [9] R. Deriche. Using canny’s criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1:167–187, 1987.
- [10] J. M. Gauch. Image segmentation and analysis via multiscale gradient watershed hierarchies. *IEEE Transactions on Image Processing*, 8(1), 1999.
- [11] D. Geman, S. Geman, C. Graffigne, and P. Dong. Boundary detection by constrained optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 609–628, 1990.
- [12] T. Gerstner and M. Rumpf. Multiresolutional parallel isosurface extraction based on tetrahedral bisection. In *Workshop Volume Visualization ’99, submitted*, 1999.
- [13] R. Grosso, C. Luerig, and T. Ertl. The multilevel finite element method for adaptive mesh optimization and visualization of volume data. In *Proceedings Visualization*, 1997.
- [14] W. Hackbusch. *Multi-grid methods and applications*. Springer, Berlin/Heidelberg, 1985.
- [15] K. Haris, S. N. Efstratiadis, N. Maglaveras, and A. K. Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Transactions on Image Processing*, 7(12), 1998.
- [16] G. A. Hewer, C. Kenney, and B. S. Man-

- junathg. Variational image segmentation using boundary functions. *IEEE Transactions on Image Processing*, 7(9), 1998.
- [17] H. Hoppe. Progressive meshes. In *SIGGRAPH 96 Conference Proceedings*, pages 99–108, 1996.
- [18] T. Kapur, W. Grimsol, W. WellsIII, and R. Kikinis. Segmentation of brain tissue from magnetic resonance image. *Medical Image Analysis*, 1(2), 1997.
- [19] J. Kačur and K. Mikula. Solution of nonlinear diffusion appearing in image smoothing and edge detection. *Appl. Numer. Math.*, 17:47–59, 1995.
- [20] D. Laur and P. Hanrahan. Hierarchical splatting: A progressive refinement algorithm for volume rendering. In T. W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 285–288, July 1991.
- [21] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modelling with front propagation. *IEEE Trans. Pattern Anal. Machine Intell.*, 17, 1995.
- [22] D. Marr and E. Hildreth. Theory of edge detection. In *Proc. Royal Soc. Lond.*, volume B 207, pages 187–217, 1980.
- [23] F. Meyer and S. Beucher. Morphological segmentation. *J. Vis. Commun. Image Represent.*, 1:21–46, 1990.
- [24] D. Mumford and J. Shah. Boundary detection by minimizing functionals. pages 22–26, 1985.
- [25] M. Oehlberger and M. Rumpf. Adaptive projection operators in multiresolutional scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 4 (4), 1998.
- [26] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. 1987.
- [27] T. Preußner and M. Rumpf. An adaptive finite element method for large scale image processing. *Journal of Visual Comm. and Image Repres.*, to appear, 1999.
- [28] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, and R. Kikinis. Three-dimensional multiscale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, 2(2), 1998.
- [29] K. Siddiqi, Y. B. Lauzière, A. Tannenbaum, and S. W. Zucker. Area and length minimizing flows for shape segmentation. *IEEE Transactions on Image Processing*, 7(3), 1998.
- [30] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Machine Intell.*, 13, 1991.
- [31] J. Weickert. *Anisotropic Diffusion in Image Processing*. Springer, 1998.
- [32] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3), 1998.