Eccentricity based Topological Feature Extraction*

Adrian Ion[1], Samuel Peltier[2], Sylvie Alayrangues[3] and Walter G. Kropatsch[1]

[1] Pattern Recognition and Image Processing Group, Vienna University of Technology, Austria
ion,krw@prip.tuwien.ac.at
[2] IFP, Rueil-Malmaison, France
samuel.peltier@gmail.com
[3] Laboratoire XLIM-SIC, UMR CNRS 6172, Université de Poitiers, France
sylvie.alayrangues@sic.univ-poitiers.fr

Abstract The eccentricity transform associates to each point of a shape the geodesic distance to the points farthest away. This can be used to decompose shapes into parts based on the connectivity of the isoheight lines. The adjacency graph of the obtained decomposition is similar to a Reeb graph and characterizes the topology of the object. Graph pyramids are used to efficiently compute the decomposition and its adjacency graph. Initial experimental results and discussion for 2D and 3D are given.

Keywords eccentricity transform, topology, Reeb graphs, graph pyramids

1 Introduction

Topological features play an important role in structural pattern recognition due to their ability to describe classes of objects robust with respect to changes in the geometry produced by usual deformations or noise. Application domains include general object categorization, medical image processing, study of different structures (alloys, bones, etc.), and study of the behavior over time by considering an embedding in a higher dimensional space (e.g. a beating heart in 3D gives a 4D object).

A special kind of graphs introduced by G. Reeb in [14] is often used in this context. It is computed on an object with respect to a particular function depending on the application and produces a kind of topological skeleton of the object. The accuracy of the representation deeply depends on the chosen function.

The *eccentricity transform* (named from the eccentricity of a vertex in graph theory) associates to each point of a shape the length of the shortest geodesic path(s) connecting it to the point farthest away i.e the **longest** of the shortest geodesics. The eccentricity transform is part of a class of image transforms also containing the distance transform [15], which associates to each point the length of the **shortest** path to the border, and the Poisson equation [5], which can be used to associate to each point the **average** time to reach the border by a random path (an average of the random shortest paths).

We propose to extract topological features of a shape, based on observing the evolution of the isolines of the eccentricity transform through a graph similar to the Reeb graph. Using a measure based on geodesic distances for obtaining a Reeb graph is also proposed in [1], where the focus is on average rather than longest distances.

Initial ideas regarding the presented decomposition are given in [6], but only focusing on 2D shapes without holes and no direct interest regarding the topology of the shape. Also the connection eccentricity - Reeb graphs is novel.

*Partially supported by the Austrian Science Fund under grants S9103-N13 and P18716-N13.

Section 2 recalls Reeb graphs, eccentricity and graph pyramids which can be used to improve the computation. Section 3 focuses on the computation of the proposed graph. Section 4 considers the extraction of topological features, properties and open points. Conclusion follows in Section 5

2 Recall

In this section basic definitions and properties of the eccentricity transform, graph pyramids, and Reeb graphs are given.

2.1 Eccentricity Transform

The following definitions and properties follow [7]. Let the shape $\mathcal{S}$ be a closed set in $\mathbb{R}^n$ and $\partial\mathcal{S}$ be its border[1]. A (geodesic) path π is a continuous mapping from the interval $[0,1]$ to $\mathcal{S}$. Let $\Pi(\mathbf{p_1},\mathbf{p_2})$ be the set of all paths between two points $\mathbf{p_1},\mathbf{p_2} \in \mathcal{S}$ within the set $\mathcal{S}$. The geodesic distance $d(\mathbf{p_1},\mathbf{p_2})$ between two points $\mathbf{p_1},\mathbf{p_2} \in \mathcal{S}$ is defined as the length $\lambda(\pi)$ of the shortest path $\pi \in \Pi(\mathbf{p_1},\mathbf{p_2})$ between $\mathbf{p_1}$ and $\mathbf{p_2}$:

$$d(\mathbf{p_1},\mathbf{p_2}) = \min\{\lambda(\pi(\mathbf{p_1},\mathbf{p_2}))|\pi \in \Pi\} \text{ where } \lambda(\pi(\mathbf{p_1},\mathbf{p_2})) = \int_0^1 |\dot{\pi}(t)|dt \tag{1}$$

where $\pi(t)$ is a parametrization of the path from $\mathbf{p_1} = \pi(0)$ to $\mathbf{p_2} = \pi(1)$.

The eccentricity transform of $\mathcal{S}$ can be defined as, $\forall \mathbf{p} \in \mathcal{S}$

$$ECC(\mathcal{S},\mathbf{p}) = \max\{d(\mathbf{p},\mathbf{q}) : \mathbf{q} \in \mathcal{S}\} \tag{2}$$

i.e. to each point $\mathbf{p}$ it assigns the length of the shortest geodesic to the points farthest away from it. In [10] it is shown that this transformation is quasi-invariant to articulated motion and robust against salt and pepper noise (which creates holes in the shape).

This paper considers the class of shapes $\mathcal{S}$ defined by points on a square grid $\mathbb{Z}^2$ that are 4-connected. Paths need to be contained in the area $\mathbb{S} \subseteq \mathbb{R}^2$ defined by the union of the support squares/cubes for the pixels/voxels of $\mathcal{S}$. The distance between any two pixels whose connecting segment[2] is contained in $\mathcal{S}$ is computed using the L_2-*norm*. In a graph theoretical framework, a vertex is associated to every point of $\mathcal{S}$. Every two vertices, whose associated points can be connected by a line segment included in $\mathbb{S}$, are connected by an edge weighted with the length of the line segment. The geodesic distance $d(\mathbf{p},\mathbf{q})$ is the length of the shortest path connecting the vertices associated to $\mathbf{p}$ and $\mathbf{q}$.

Terminology:
An *eccentric point* of a shape $\mathcal{S}$ is a point $\mathbf{e} \in \mathcal{S}$ that is farthest away in $\mathcal{S}$ from at least one other point $\mathbf{p} \in \mathcal{S}$ i.e. $\exists \mathbf{p} \in \mathcal{S}$ s.t. $ECC(\mathcal{S},\mathbf{p}) = d(\mathbf{p},\mathbf{e})$.

The *center* $C \subseteq \mathcal{S}$ of a shape $\mathcal{S}$ is the set of points $\mathbf{c} \in C$ with the smallest eccentricity i.e. $\mathbf{c} \in C \iff ECC(\mathcal{S},\mathbf{c}) = min\{ECC(\mathcal{S},\mathbf{p}), \forall \mathbf{p} \in \mathcal{S}\}$. If the shape $\mathcal{S}$ is a simply connected shape[3] in $\mathbb{R}^2$, the center C is a single point. If $\mathcal{S}$ is a simply connected discrete shape in $\mathbb{Z}^2$, C is a connected set of less than 5 pixels. Otherwise it can be a set of arbitrary size and connectivity (e.g. for $\mathcal{S}$ made out of the points on a circle, all points are eccentric and they all make up the center).

The smallest eccentricity is called the *radius* of the shape, and the highest one is called the *diameter*.

[1] This definition can be generalized to any dimension, continuous and discrete objects.
[2] Straight line segment connecting the two points.
[3] No holes.

Properties:
The variation of geodesic distances is bounded under articulated deformation to the width of the 'joints' [11]. The eccentricity transform uses geodesic distances and is bounded in the same way.

The transform is very robust with respect to noise, and the positions of eccentric points and the center are stable [10]. They change only if all supporting pixels for a certain eccentric point are removed or if the diameter changes i.e. the shape is modified around the points with the highest eccentricity.

See Figure 2(a) and (b) for an example of a shape and its eccentricity transform.

2.2 Irregular Graph Pyramids

A *graph pyramid* is a stack of successively reduced graphs $P = \{G_0, \ldots, G_t\}$. Each level $G_k = (V_k, E_k)$, $1 \leq k \leq t$, is obtained by *contracting* and *removing* edges in the level below. Successive levels reduce the size of the data by $\lambda > 1$. Edges and vertices of G_k can be attributed. The *reduction window* relates a vertex at a level G_k with a set of vertices in the level directly below (G_{k-1}). Higher level descriptions are related to the original input data. This is done by the *receptive field* (RF) of a given vertex $v \in G_k$, which aggregates all vertices in the base level (G_0) of which v is an ancestor.

Each level represents a partition of the base level into connected subgraphs i.e. *connected subsets of pixels* in our case. The union of neighboring vertices on level $k-1$ (children) to a vertex on level k (parent) is controlled by contraction kernels (CK) [8], a spanning forest which relates two successive levels of a pyramid. Contraction kernels are chosen based on the content and the structure of the data[4]. Every parent computes its values independently of other on the same level. Thus local independent (and parallel) processes propagate information up and down and laterally in the pyramid [9].

2.3 Reeb graphs

Reeb graphs are used to partially encode the topology of manifolds and to provide a compact representation of them. More precisely, they provide a kind of skeleton of the object on which they are computed. They are for instance used in $3D$ shape retrieval methods together with graph matching algorithms. The construction of such graphs is deeply related to the choice of the function, f.

A mathematical definition of Reeb graphs of manifolds can actually be expressed as follows [3]. Let $f : M \longrightarrow R$ be continuous, and call a component of a level set a contour. Two points x, $y \in M$ are equivalent if they belong to the same component of $f^{-1}(t)$ with $t = f(x) = f(y)$. The Reeb graph of f is the quotient space defined by this equivalence relation.

Most applications use functions with particular properties such as Morse functions. A Morse function is a function whose critical points (i.e. points where the derivative is null) are non degenerated (i.e. the second derivative at these points is different from zero).

If using such functions, each node of the obtained graph corresponds to a critical point of the function f, in other words there is one node per level set of f. Edges between nodes somehow represent the connectivity between level sets. Moreover the preservation of the topology can be more easily quantified with such functions. For instance, when dealing with $2D$-orientable manifolds, it is known that the number of loops of the graph is exactly the genus of manifold [3].

The choice of f is not straightforward [2]. It often depends on the properties of the objects on which the matching shall focus. Ideally such functions should also be invariant under classical transformations such as translation, rotation, scaling. Moreover they should not induce expensive computational cost. The simplest function often used is the height function: it associates to each point of the manifold its altitude in a given direction (See Figure 1). Such a function is obviously not invariant under rotation as if the direction changes the height function also does. It is hence particularly suited for applications where a given orientation can naturally be associated to the

[4]For example, for connected component labeling, edges connecting vertices with the same label are contracted.

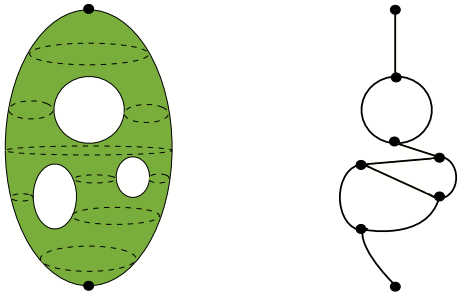

Figure 1: An object with 3 tunnels and its associated Reeb graph obtained with the elevation function

object. But it is not adapted to the classification of any set of images. Many other functions can also be used based for instance on the distance from the barycenter, or the computation of curvature extrema.

3 2D Shape Decomposition based on ECC isoheight Lines

The *level set* of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ corresponding to a real value h is the set of points $\{\mathbf{p} \in \mathbb{R}^n : f(\mathbf{p}) = h\}$ [16].

A level set of the eccentricity transform of $\mathcal{S} \in \mathbb{R}^n$ is the set:

$$LS(e) = \{\mathbf{q} \in \mathcal{S} : ECC(\mathcal{S}, \mathbf{q}) = e\}, \tag{3}$$

with $e \in [\min\{ECC(\mathcal{S}, \mathbf{p})\}, \max\{ECC(\mathcal{S}, \mathbf{p})\}]$ an eccentricity value. If $S \in \mathbb{R}^2$, $LS(e)$ can be a single closed curve (cycle) or a set of disconnected open curves. The connected components of $LS(e)$ are called *isoheight lines*, $IL \subseteq LS(e)$, IL connected.

Given a shape $\mathcal{S}$, a *decomposition of* $\mathcal{S}$ *into simply connected regions* is the set $\{R_1, \ldots, R_n\}$ s.t. $\bigcup R_k = \mathcal{S}, k = 1, \ldots, n$; $R_i \bigcap R_j = \emptyset, \forall i, j \in \{1, \ldots, n\}$; and R_i is a simply connected region.

$HD(\mathcal{S}) = \{R_1, \ldots, R_n\}$ is a *a decomposition of* $\mathcal{S} \in \mathbb{R}^2$ *based on the connectivity of the ECC isoheight lines* if HD is a decomposition into simply connected regions (as defined above), $\forall R_i$ and $\forall e \in [\min\{ECC(\mathcal{S}, \mathbf{p})\}, \max\{ECC(\mathcal{S}, \mathbf{p})\}] \Rightarrow R_i \bigcap LS(e)$ is a connected region, all IL part of the same R_i are closed or all open (same topology), and n, the number of regions, is **minimal**. $HD(\mathcal{S})$ exists for any connected shape $\mathcal{S}$.

Given a discrete shape $\mathcal{S}$ and its eccentricity transform $ECC(\mathcal{S})$, $HD(\mathcal{S})$ can be computed by:

1. **in a graph theoretical framework:** Algorithm 1 creates a graph pyramid s.t. the top level G_h is an oriented region adjacency graph describing the topology of the decomposition $HD(\mathcal{S})$. Edges of G_h are oriented from regions with lower eccentricity to regions with higher eccentricity. Each vertex contains the length of the longest isoheight line in its receptive field.

2. **sequential approach:** 'follow' the isoheight lines from the minimum eccentricity to the maximum eccentricity. Whenever an isoheight line gets disconnected, or merged, new regions are started for the formed isoheight line part(s). This way of presenting the algorithm is similar to building a Reeb graph, and equivalent with Algorithm 1. (The sequential approach is more intuitive, but needs building the adjacency graph for the decomposition 'over' it. In addition, it is lacking the fast access advantages when searching for the pixel with a known coordinate.).

We will use *decomposition graph* to denote the adjacency graph of the decomposition, as obtained in the top level of the pyramid produced by Algorithm 1. Figure 2(c) shows an example decomposition based on the connectivity of isoheight lines.

Algorithm 1 *HD* - Decomposition of $\mathcal{S} \in \mathbb{R}^2$ based on the connectivity of ECC isoheight lines

Input: Discrete shape $\mathcal{S}$.

1: $iECC = \lfloor ECC(\mathcal{S}) \rfloor$ /*compute ECC, round = at least 8 connected isoheight lines*/
2: $G_0 \leftarrow$ oriented neighborhood graph of $iECC$ /*ensure proper connectivity of isoheight lines while keeping G_0 planar, orient from small to high eccentricity*/
3: $k \leftarrow 0$
4: **for all** $v \in V_k$ **do** $v.maxlength \leftarrow 1, v.ecc \leftarrow [ECC(v), ECC(v)]$, /*init max length of isoheight lines and eccentricity interval for each vertex*/
5: **repeat**
6: $A \leftarrow \{e = (v_i, v_j) \in E_k : v_i.ecc = v_j.ecc\}$ /*merge isoheight segment parts*/
7: $A \leftarrow A \bigcup \{e = (v_i, v_j)$:deg(v_i),deg(v_j) ≤ 2 and closed(v_i)=closed(v_j)$\}$ /*same region, closed(v)=true $\iff$ receptive field of v contains only closed isoheight lines*/
8: **if** $|A| > 0$ **then**
9: $K \leftarrow$ contraction kernels as subset of A /*use MIS or MIES [9] to optimally break A into valid contraction kernels*/
10: $G_{k+1} \leftarrow$ contract(G_k, K) /*contract edges in K and simplify*/
11: **for all** $v \in V_{k+1}$ **do** compute $v.maxlength, v.ecc$ from G_k /*use reduction window*/
12: $k \leftarrow k + 1$
13: **end if**
14: **until** $|A| = 0$
15: $h \leftarrow k$

Output: Graph Pyramid $G_0, \ldots, G_h$.

4 Eccentricity based Reeb

The graph produced in Section 3 as a result of the decomposition, is similar in concept with the Reeb graph, if considering the eccentricity transform as the function f (see Section 2.3).

For the shape in Figure 2(a), the produced graph contains 53 nodes. Figure 2(d) shows the graph. Note that for visualization purposes, 21 leaves corresponding to tiny regions with 1-3 pixels have been removed. The two cycles identifying the two holes are shown in Figure 2(e), and the region corresponding to the vertex with the highest degree (6) is given in Figure 2(f). Both figures can be easily obtained from Figure 2(d) by using edge contraction and removal, and the concept of receptive field.

To directly obtain the cycles in the decomposition graph, one can use the approach proposed in [4] and contract all edges which are not selfloops. The obtained graph will contain a self loop for each hole. Also, the graph in Figure 2(e) can be transformed to one like in Figure 1 (i.e. without any loops enclosed in other loops), with a continuous deformation as described in [13].

The following main questions arise from these concepts:

Correctness: the fact that the decomposition graph contains exactly as many selfloops as holes is an intuition and is proposed as future study. Note that a similar decomposition can be done in the same way, for other transforms also (e.g. the single source shape bounded distance transform, also called geodesic distance function). We use the eccentricity transform because its center is stable [10] and there is no need to give a starting point. On the other side, proving that the obtained graph has the number of cycles equal to the number of holes in the input object could be easier for the geodesic distance function.

Going to higher dimensions: the algorithm in Section 3 uses connectivity and whether an isoline is closed or not to create the decomposition for 2D shapes. Only level sets LS having the same topology (intuitively, number of connected components and of holes, if the 2D shape is embedded in 2D) are grouped together. If going to 3D, level sets become surfaces, which have properties regarding connectedness, 1D and 2D holes (tunnels & cavities). One might consider

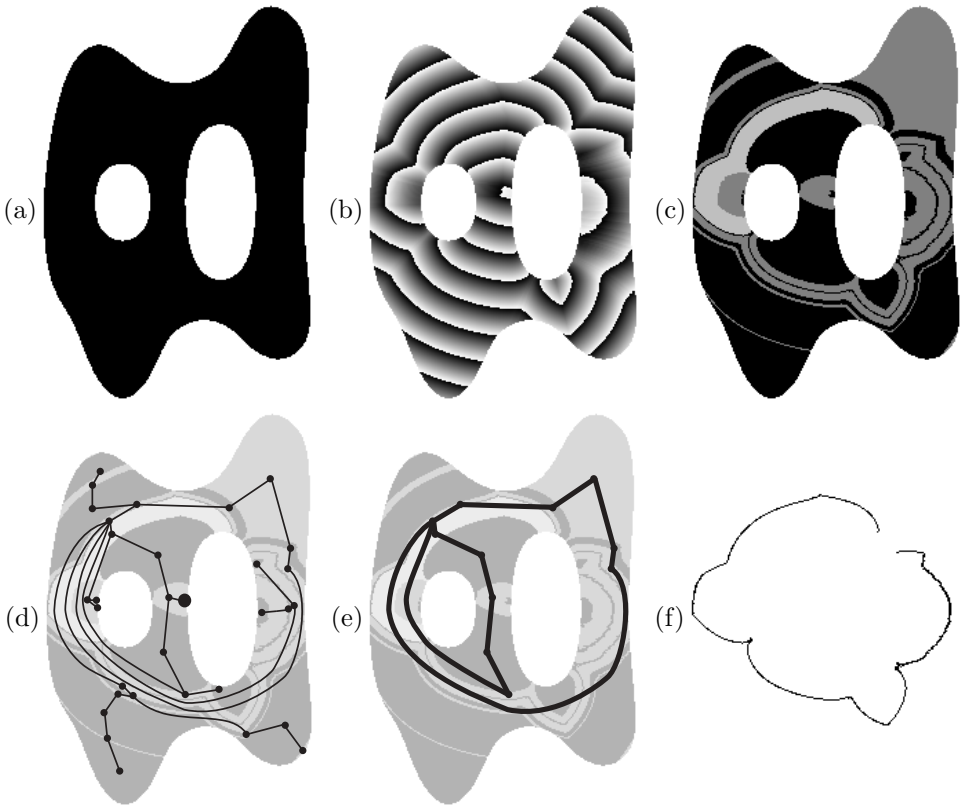

Figure 2: (a) A 2D Shape with 2 holes, (b) its eccentricity transform (modulo 20), (c) decomposition (connected & same gray value means same region), (d) decomposition graph (leaves belonging to 1-3 pixel regions have been left out for drawing clarity), (e) the two cycles in the graph, (f) image region corresponding to vertex with highest degree (6) (the small width is just a coincidence).

putting all neighboring connected components of level sets together, if they share the same number of tunnels and cavities. This would produce the correct result for the object in Figure 1, but ambiguities can appear in the cases of cavities. Like Reeb graphs, the obtained graphs will certainly not be able to distinguish between cavities and tunnels (e.g. a plain sphere i.e. a solid with one cavity, could result in the same graph as the torus). To overcome this difficulty, a solution could be to compare this invariant to other topological invariants such as the homology groups that provide an accurate representation of holes in any dimension [12]. One could also imagine applying the same algorithm again, for each of the 2D eccentricity level sets obtained for a 3D object. But the question of putting together the results from each level is still open.

Relations between features: what are the relations between the eccentricity transform and the holes in an object? (e.g. a disconnected center $\implies$ at least one hole).

5 Conclusion

A concept for using isolines of the eccentricity transform to extract topological features is presented. The isolines are used to decompose a 2D shape into parts. The adjacency graph of the decomposition is similar to the Reeb graph and can be used to characterize the holes. A discussion regarding properties and extension to 3D is given.

References

[1] Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. Partitioning of 3d meshes using reeb graphs. In *18th International Conference on Pattern Recognition (ICPR)*, pages 19–22. IEEE Computer Society, 2006.

[2] Silvia Biasotti, Simone Marini, Michela Mortara, and Giuseppe Patanè. An overview on properties and efficacy of topological skeletons in shape modelling. In *Shape Modeling International*, pages 245–256, 297. IEEE Computer Society, 2003.

[3] K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Loops in reeb graphs of 2-manifolds. *Discrete & Computational Geometry*, 32(2):231–244, 2004.

[4] Rocio Gonzalez-Diaz, Mabel Iglesias-Ham, Adrian Ion, and Walter G. Kropatsch. Irregular graph pyramids, integral operators and representative cocycles of cohomology generators. In *submitted to the Workshop on Computational Topology in Image Context, CTIC*, Poitiers, France, June 2008.

[5] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt. Shape representation and classification using the poisson equation. In *CVPR (2)*, pages 61–67, 2004.

[6] Adrian Ion and Walter G. Kropatsch. Mapping a coordinate system to a non-rigid shape. In *32nd OAGM/AAPR*, Linz, Austria, May 2008. OCG.

[7] Adrian Ion, Walter G. Kropatsch, and Eric Andres. Euclidean eccentricity transform by discrete arc paving. In *14th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI)*, Lecture Notes in Computer Science, Lyon, France, April 2008. Springer.

[8] Walter G. Kropatsch. Building irregular pyramids by dual graph contraction. *IEE-Proc. Vision, Image and Signal Processing*, 142(6):366–374, December 1995.

[9] Walter G. Kropatsch, Yll Haxhimusa, Zygmunt Pizlo, and Georg Langs. Vision pyramids that do not grow too high. *Pattern Recognition Letters*, 26(3):319–337, 2005.

[10] Walter G. Kropatsch, Adrian Ion, Yll Haxhimusa, and Thomas Flanitzer. The eccentricity transform (of a digital shape). In *13th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI)*, Lecture Notes in Computer Science, pages 437–448, Hungary, October 2006. Springer.

[11] Haibin Ling and David W. Jacobs. Shape classification using the inner-distance. *IEEE Transactions Pattern Analysis Machine Intelligence*, 29(2):286–299, 2007.

[12] S. Peltier, S. Alayrangues, L. Fuchs, and J.-O. Lachaud. Computation of homology groups and generators.. *Computer & Graphics*, 30(1):62–69, February 2006.

[13] Samuel Peltier, Adrian Ion, Walter G. Kropatsch, Guillaume Damiand, and Yll Haxhimusa. Directly computing the generators of image homology using graph pyramids. In *submitted to Image and Vision Computing (extended version of GBR2007).*

[14] G. Reeb. Sur les points singuliers d'une forme de pfaff complément intégrable ou d'une fonction numérique. *Annales de l'institut Fourier, 14 no. 1*, 14(1):37–42, 1964.

[15] A. Rosenfeld. A note on 'geometric transforms' of digital sets. *Pattern Recognition Letters*, 1(4):223–225, 1983.

[16] J.A. Sethian. *Level Set Methods and Fast Marching Methods.* Cambridge Univ. Press, 2nd edition, 1999.