Level Sets and Voronoi based Feature Extraction from any Imagery

Ojaswa Sharma
Dept. of Computer Science
IIT Mumbay
Mumbai, India
Email: ojaswa@gmail.com

François Anton
Dept. of Informatics and Mathematical Modelling
Technical University of Denmark
Kongens Lyngby, Denmark
Email: fa@imm.dtu.dk

Darka Mioc
National Space Institute
Technical University of Denmark
Kongens Lyngby, Denmark
Email: mioc@space.dtu.dk

***Abstract*—*Polygon features are of interest in many GEOProcessing applications like shoreline mapping, boundary delineation, change detection, etc. This paper presents a unique new GPU-based methodology to automate feature extraction combining level sets, or mean shift based segmentation together with Voronoi skeletonization, that guarantees the extracted features to be topologically correct. The features thus extracted as object centerlines can be stored as vector maps in a Geographic Information System after labeling and editing. We show application examples on different sources: paper maps, digital satellite imagery, and 2D/3D acoustic images (from hydrographic surveys). The application involving satellite imagery shown in this paper is coastline detection, but the methodology can be easily applied to feature extraction on any king of imagery. A prototype application that is developed as part of this research work.*

***Keywords*-*Feature Extraction; Imagery; Segmentation; Level sets; Voronoi; Mean Shift.*

I. INTRODUCTION

Polygon features are of interest in applications like shoreline mapping, boundary delineation, change detection, etc. This paper presents a unique new GPU-based methodology to automate feature extraction on any imagery (not only raster images) by deformation of level sets or mean shift based segmentation / Voronoi skeletonization that guarantees the extracted features to be topologically correct. The features thus extracted as object centerlines can be stored as vector maps in a Geographic Information System after labeling and editing. We show application examples on different sources: scanned paper maps, digital satellite imagery, and 2D/3D acoustic images (from hydrographic surveys). This paper presents new results closely related to previous work of the authors on Voronoi [26] and level set evolution [25] based feature extraction.

The application involving satellite imagery shown in this paper is coastline detection, but the methodology can be easily applied to feature extraction on any king of imagery. For example, a coastline is defined as the boundary between land and water. Coastline mapping is important for coastal activity monitoring, resource mapping, navigation, etc. A lot of work on coastline extraction from SAR (Synthetic Aperture Radar) and multi-spectral imagery has been done. A technique for coastline extraction from remotely sensed images using texture analysis is described in [3]. The delineation of the complete coastline of Antarctica using SAR imagery is shown in [17]. A morphological segmentation based automated approach for coastline extraction has been suggested in [2]. Di et al. use the image segmentation algorithm by [8] to segment an image and detect the shoreline [9]. Our work in progress is an extension to the work by Gold and Snoeyink [12]. Our boundary (crust) extraction algorithm converts detected image features into connected sets of vectors that are topologically equivalent to the segmented objects. We claim topological equivalence of the extracted features since these are obtained as subsets of the Voronoi diagram or the Delaunay triangulation. This method can be applied on imageries, that have a high signal to noise ratio (scanned maps, aerial photographs, optical satellite imagery). For lower signal to noise imagery, we need level sets or continuous deformations.

The application involving acoustic images shown in this paper is 3D reconstruction of fishes from large acoustic datasets. Level set based methods have been shown to successfully restore noisy images [24]. Malladi and Sethian [18] have shown image smoothing and enhancement based on curvature flow interpretation of the geometric heat equation. In a more recent approach to use level set methods for acoustic image segmentation, Lianantonakis and Petillot [16] provide an acoustic image segmentation framework using the region based active contour model of Chan and Vese [4]. Here we focus on using the level set methods for simultaneous suppression of noise and 3D reconstruction of relevant features. We limit features of interest to fishes from acoustic images and provide a level set based framework for acoustic image segmentation. Image restoration techniques based on level set evolution are generally oriented to segment the image or to remove noise from it. Work by Lianantonakis and Petillot [16] is closest to our approach since they use active contours using Mumford-shah functional for seabed classification, but together with extraction of Haralick feature set for textural analysis.

Since acoustic data resulting from marine surveys can result in gigabytes of information, we employ GPU (Graphics Processing Unit) based computations for 3D reconstruction. The GPU is not very suitable for data intensive applications due to unavailability of large memory on commodity hardware. A number of publications suggest schemes to circumvent this

 ISBN: 978-1-61208-178-6

situation by performing computations in a streaming manner [14], but most of the implementations process 2D sections to generate a 3D reconstruction. We present a new Level Set method implementation with computations performed entirely in 3D using the 3D textures (read only) available to the new CUDA (Compute Unified Device Architecture) 2.0 framework.

This paper is organized as follows. Section II presents the new methodology for image segmentation and feature extraction. Section III presents the results obtained with the new methodology presented in Section II. Finally, Section IV concludes this paper.

II. IMAGE SEGMENTATION AND FEATURE EXTRACTION

Edge detection produces global edges in an image. This means that there is no object definition attached to the edges. Therefore, it is required to somehow define the objects first and then obtain edges from them. This can be achieved by using *image segmentation*. The main goal of image segmentation is to divide an image into parts that have a strong correlation with objects or areas of the real world depicted in the image [28, chap. 5]. Thus, image segmentation divides the whole image into homogeneous regions based on color information. The regions can be loosely defined as representatives of objects present in the image. While edge detection is very sensitive to noise, level sets based segmentation tolerates noise quite well.

A. Mean Shift Algorithm Segmentation

The segmentation method adopted here is the one provided by [7] which is based on feature space analysis. Feature space analysis is used extensively in image understanding tasks. [7] provide a comparatively new and efficient segmentation algorithm that is based on feature space analysis and relies on the *mean-shift algorithm* to robustly determine the cluster means. A *feature space* is a space of feature vectors. These features can be object descriptors or patterns in case of an image. As an example, if we consider a color image having three bands (red, green, and blue) then the image we see as intensity values plotted in Euclidean XY space is said to be in *image space*. Consider a three dimensional space with the axes being the three bands of the image. Each color vector corresponding to a pixel from the image can be represented as point in the feature space.

Given n data points x_i, $i = 1, \ldots, n$ in the d-dimensional space $\mathbb{R}^d$, a *flat kernel* of a location x that is the characteristic function of a λ-ball in $\mathbb{R}^d$ is defined as

$$K(x) = \begin{cases} 1 \text{ if } \|x\| \leq \lambda \\ 0 \text{ if } \|x\| > \lambda \end{cases} . \quad (1)$$

The *mean shift* vector at x is defined using the kernel of radius r as

$$M_\lambda(x) = \frac{\sum_{r \in \mathbb{R}^d} xK(r-x)}{\sum_{r \in \mathbb{R}^d} K(r-x)} - x \quad (2)$$

Cheng [6] shows that the mean shift vector, the vector of difference between the local mean and the center of the window $K(x)$, is proportional to the gradient of the probability density at x [6]. Thus mean shift is the steepest ascent with a varying step size that is the magnitude of the gradient. Further, Comaniciu and Meer use mean shift vector in seeking the mode of a density by shifting the kernel window by the magnitude of the mean shift vector repeatedly [8]. The authors also prove that the mean shift vector converges to zero and eventually reaches the basin of attraction of that mode.

In their research work, Comaniciu and Meer state a simple, adaptive steepest ascent mode seeking algorithm [7].

1) Choose the radius r of the search window (i.e, radius of the kernel).
2) Choose the initial location of the window.
3) Compute the mean shift vector and translate the search window by that amount.
4) Repeat until convergence.

The mean shift algorithm gives a general technique of clustering multi-dimensional data and is applied here in color image segmentation. The fundamental use of mean shift is in seeking the modes that give regions of high density in any data.

The method described in [7] provides an autonomous segmentation technique with only the type of segmentation to be specified by the user. This method emphasizes the importance of utilizing the image space along with the feature space to efficiently perform the task of segmentation. The segmentation has three characteristic input parameters:

- Radius of the search window, r,
- Smallest number of elements required for a significant color, N_{min}, and
- Smallest number of connected pixels necessary for a significant image region, N_{con}.

The size of the search window determines the resolution of the segmentation, smaller values corresponding to higher resolutions. The authors use square root of the trace of global covariance matrix of the image, σ, as a measure of the visual activity in the image. The radius r is taken proportional to σ. Later, Comaniciu and Meer provide an improvement [8] over this segmentation algorithm by merging the image domain and the feature (range) space into a joint spatial-range domain of dimension $d = p + 2$, where p is the dimension of the range domain. This gives an added advantage of considering both the spaces together and gives good results in cases where non-uniform illumination produces false contours when the previous segmentation algorithm is used. Therefore, the new algorithm is particularly useful to segment natural images with man-made objects. An added computational overhead to process higher dimensional space is inevitable here. The simple mean shift based segmentation algorithm provides satisfactory results in the case of scanned maps as shown in the results section.

Segmentation provides us with definite boundaries of objects that are used to extract sampling points around an object. This methodology does not allow one to guarantee that the topology of the segmented objects matches the topology of the imaged objects.

 ISBN: 978-1-61208-178-6

B. Feature extraction after mean shift algorithm

Anton et *al.* [1] suggest a new algorithm for skeleton extraction. This is based on the concept of *Gabriel Graph* [10]. A Gabriel graph G (highlighted in Figure 1) is a connected subset of the Delaunay graph $\mathscr{D}$ of points in set S such that two points p_i and p_j in S are connected by an edge of the Gabriel graph, if and only if, the circle with diameter p_ip_j does not contain any other point of S in its interior. In other words, the edges in G are those edges from $\mathscr{D}$ whose dual Voronoi edges intersect with them.

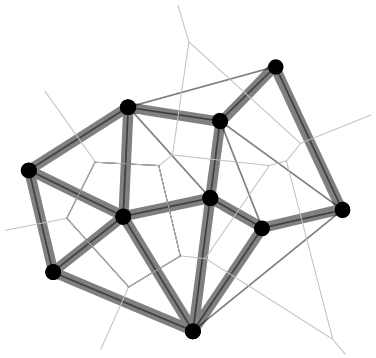


Fig. 1. Gabriel graph highlighted in a Delaunay triangulation.

Given the Delaunay triangulation $\mathscr{D}$ and the Voronoi diagram V of sample points S from the boundary of an object, the algorithm for centerline extraction in [1] proceeds by selecting all the Gabriel edges in graph G. Each dual Voronoi edge v of the Gabriel edge g from G is inserted in the skeleton K if the following condition is met:

$$\begin{gathered} g.Origin.Colour \neq g.Destination.Colour \\ \text{Or} \\ g.Origin.Colour \neq v.Origin.Colour \\ \text{Or} \\ g.Origin.Colour \neq v.Destination.Colour \\ \text{And} \\ \|g.Origin.Colour - g.Dest.Colour\| \geq \\ \|v.Origin.Colour - v.Dest.Colour\| \end{gathered} \tag{3}$$

Here, $Origin.Colour$ and $Destination.Colour$ are color values from the gray scale image corresponding to the location of the origin and the destination of an edge respectively.

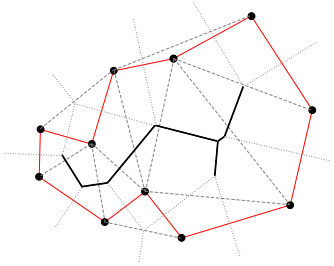


Fig. 2. Anti-crust from the crust.

1) Obtaining Anti-crust from the Voronoi diagram: The anti-crust of an object, as described above, forms a tree like structure that contains the skeleton. Once all the Delaunay edges belonging to the border set or the crust are identified using the condition given by Gold [11], it is easy to identify the Voronoi edges belonging to the anti-crust. In Figure 2, consider the Delaunay triangulation (dashed edges), the corresponding Voronoi diagram (dotted edges) and the crust edges (solid red edges).

Navigation from a Delaunay edge to its dual Voronoi edge can be achieved by using the $Rot()$ operator in the quad-edge data structure [15]. A Voronoi edge $e.Rot()$ of the dual Delaunay edge e is marked as an edge belonging to the anti-crust if the following conditions are satisfied:

1) $e \notin Crust$
2) $e.Rot().Origin \in I$
3) $e.Rot().Destination \in I$,

where $e.Rot().Origin$ is the origin of the edge $e.Rot()$, $e.Rot().Destination$ is the destination of the edge $e.Rot()$ and I is the selected object. This marks all the Voronoi edges belonging to the anti-crust that fall inside the selected object. Negating conditions (2) and (3) so that the points do not fall inside the object will give us the exterior skeleton or the *exoskeleton*. Once the anti-crust is identified, an appropriate pruning method can be applied to get rid of the unwanted edges.

2) Pruning: Gold [11] also discusses the "hairs" around the skeleton that result due to the presence of three adjacent sample points whose circumcircle does not contain any other sample point - either near the end of a main skeleton branch or at locations on the boundary where there is minor perturbation because of raster sampling. Gold and Thibault [13] suggest a skeleton retraction scheme in order to remove the hairs that also results in smoothing of the boundary of the object. Ogniewicz [20] presents an elaborate skeleton pruning scheme based on various residual functions. Thus, a hierarchic skeleton is created which is good for multi-scale representation. Sharma et *al.* [27] suggest the use of ratio based pruning in order to simplify a network of skeletons for extracting linear features from satellite imagery.

The problem of identifying skeleton edges now reduces to reasonably prune the anti-crust. We next present an optimal criterion for pruning by successively removing leaf edges from the anti-crust.

3) Pruning by Removing Leaf Edges: Gold and Thibault [13] present a retraction scheme for the leaf nodes in the anti-crust. The skeleton is simplified by retracting the leaf nodes of the skeleton to their parent nodes. Gold and Thibault [13] recommend performing the retraction operation repeatedly until no further changes take place. An observation reveals that an unwanted branch in a skeleton may be composed of more than one edge (see Figure 3). Therefore, single retraction may not be sufficient to provide an acceptable skeleton.

A similar simplification can be achieved by pruning the leaf edges instead of retracting the leaf nodes. Leaf edge pruning produces satisfactory results and requires only two or three

 ISBN: 978-1-61208-178-6

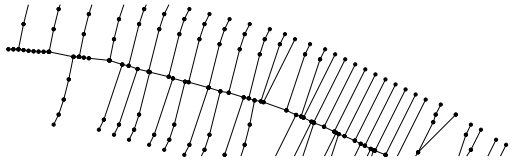


Fig. 3. Hair around the skeleton composed of multiple edges.

levels of pruning. Before pruning the leaf edges, they must be identified in the anti-crust. An edge e from a tree of edges $T \in V$, where V is the Voronoi diagram, is marked as a leaf edge if the following condition is satisfied:

$$\begin{array}{c} e.Oprev() \notin T \text{ And } e.Onext() \notin T \\ \text{Or} \\ e.Sym().Oprev() \notin T \text{ And } e.Sym().Onext() \notin T \end{array} . \quad (4)$$

This condition essentially selects all the Voronoi edges belonging to the anti-crust that have at least one end point free (i.e., connected to an edge not belonging to the anti-crust). This condition is used to locate leaf edges followed by their removal from the skeleton. Experiments show that removing leaf edges two to three times simplifies the skeleton to a major extent for linear features. We find an optimal criterion for removal of extraneous hair from the skeleton by pruning the leaf edges (see results in next section).

C. Level sets based segmentation

Let an image $I(x, y)$ be defined on a bounded open subset $\Omega : \{(x, y)|0 \leq x, y \leq 1\}$ of $\mathbb{R}^2$, with $\partial\Omega$ as its boundary. I takes discrete values between 0 and $(2^n - 1)$ where n is the number of bits used to store intensity. The basic idea in active contour model is to evolve a curve $C(s) : [0, 1] \rightarrow \mathbb{R}^2$ by minimizing the following energy functional [21]:

$$E(C) = \alpha \int_0^1 |C'|^2 \, ds + \beta \int_0 |C''| \, ds - \lambda \int_0^1 |\nabla I(C)|^2 \, ds,$$

where α, β, and λ are positive parameters, and ∇I denotes the gradient of I. In the above energy functional, the evolution of curve C is controlled by the internal energy (first two terms that define the smoothness of the curve) and the external energy (the last term that depends on the edges present in the image). More intuitively, the curve evolves to minimize the differences of intensity between the points in the interior of the curve, and at the same time maximizes the differences between the points in the interior and the points in the exterior of the curve, thus, clustering the image. The curve C can be represented by an implicit function ϕ, $C = \{(x, y)|\phi(x, y) = 0\}$, where the evolution of C is given by the zero level curve at any time t of the function $\phi(x, y, t)$.

With this formulation, an edge detector is defined as a positive decreasing function $g(\nabla I)$ based on the gradient of image [23] such that

$$\lim_{|\nabla I| \rightarrow \infty} g(\nabla I) = 0$$

Therefore, the zero level curve evolves in the normal direction and stops at the desired boundary where g vanishes.

Evolving the curve C in normal direction amounts to solving the partial differential equation (PDE) [22]

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| F \quad (5)$$

with the initial condition $\phi(x, y, 0) = \phi_0(x, y)$, where $\phi_0(x, y)$ is the initial contour. Motion by mean curvature allows for cusps, curvature and automatic topological changes [22], [5]. This results in the speed function $F = div\left(\frac{\nabla \phi}{\|\nabla \phi\|}\right)$ in terms of the curvature of ϕ

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| div\left(\frac{\nabla \phi}{|\nabla \phi|}\right), \phi(x, y, 0) = \phi_0(x, y)$$

where $div(\cdot)$ is the divergence operator.

1) Minimising the Mumford-Shah Functional in Image: Chan and Vese [5] provide an alternative approach to the edge based stopping criterion. The authors suggest the stopping term based on Mumford-Shah segmentation techniques [19]. The motivation behind using this alternative stopping term is that in many cases, the edges in an image are not very well defined. Either it is ambiguous to position the edges across the gradient due to smoothly varying intensities [5] or it is difficult to select prominent edges due to presence of noise (as in the case of acoustic images). The method of Chan and Vese [5] is minimization of an energy based segmentation. Assuming that the image I is composed of two regions of piecewise constant intensities of distinct values I^i and I^o, and that the object of interest is represented by I^i, we define the curve C to be its boundary. Using the Heaviside function H, and the Dirac-Delta function δ_0,

$$H(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0 \end{cases}, \delta_0(z) = \frac{d}{dz} H(z)$$

the energy functional is formulated as

$$\begin{aligned} E(c_1, c_2, C, t) =& \mu \int_\Omega \delta_0(\phi(x, y, t)) |\nabla \phi(x, y, t)| \, dx \, dy \\ &+ \nu \int_\Omega H(\phi(x, y, t)) \, dx \, dy \\ &+ \lambda_1 \int_\Omega |I(x, y) - c_1|^2 \, dx \, dy \\ &+ \lambda_2 \int_\Omega |I(x, y) - c_2|^2 \, dx \, dy \end{aligned} \quad (6)$$

where, $\mu \geq 0$, $\nu \geq 0$, $\lambda_1, \lambda_2 > 0$ are fixed parameters. c_1 and c_2 are average intensity values inside and outside C. The constants c_1 and c_2 can also be written in terms of I and ϕ

$$c_1 = \frac{\int_\Omega I(x, y) H(\phi(x, y, t)) \, dx \, dy}{\int_\Omega H(\phi(x, y, t)) \, dx \, dy}, \quad (7)$$

$$c_2 = \frac{\int_\Omega I(x, y)(1 - H(\phi(x, y, t))) \, dx \, dy}{\int_\Omega (1 - H(\phi(x, y, t))) \, dx \, dy} \quad (8)$$

 ISBN: 978-1-61208-178-6

The variational level set approach gives the following Euler-Lagrange equation [5]

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi)\left[\mu\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|} - \nu - \lambda_1(I-c_1)^2 + \lambda_2(I-c_2)^2\right] \quad (9)$$

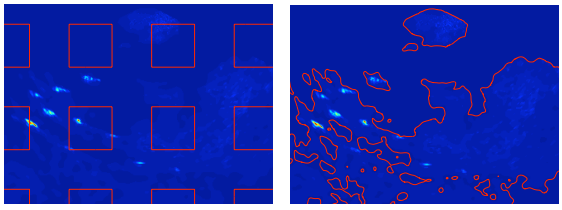
with the initial condition, $\phi(x,y,0) = \phi_0(x,y)$ and

$$\delta_\epsilon(z) = \frac{\partial}{\partial z}H_\epsilon(z) = \pi^{-1}\epsilon^{-1}\left(1+\frac{z^2}{\epsilon^2}\right)^{-1} \quad (10)$$

where, the regularised one-dimensional Heaviside function is given by:

$$H_\epsilon(z) = \frac{1}{2}\left(1+\frac{2}{\pi}\tan^{-1}\left(\frac{z}{\epsilon}\right)\right).$$

The acoustic images considered by Lianantonakis and Petillot [16] are of the seabed. Such images show strong textural variations of the bottom surface of the sea. In this paper, we restrict ourselves to acoustic images of freely swimming fishes. While such images are also corrupted by speckle noise, they do not show specific textural patterns. Figure 4(a) shows part of such an image where the fish cross sections are discriminated by very high intensities compared to the background. The presence of reflectance from air bubbles mixing into water, also contribute to the noise. While working with level sets, a standard procedure is to keep ϕ to a signed distance function [21]. A direct application of the level set equation given by equation (9), with $\phi(x,y,0) = 0$ initialized to set of squares regularly distributed over the image, shows that the evolution of the level set eventually stops at the wrong place (see Figure 4(b)). We used specific multi-beam acoustic noise removal techniques.



(a) Initialization contour. (b) Result at convergence.

Fig. 4. Application of the level set equation (9).

2) Noise suppression model: Considering the image I to be time varying, the basic idea behind noise suppression is to solve the following equation as an update step to the level set equation resolution in a single pass:

$$\frac{\partial I(x,y,t)}{\partial t} = k\cdot\max(0,\hat{c}-I(x,y,t)) \quad (11)$$

where k is a constant and $\hat{c}$ is a scalar parameter that is computed as an optimal threshold at any time step t based on $\phi(x,y,t)$.

The computation of $\hat{c}$ is based on the bounded subset I^i given by

$$I^i(x,y,t) = I(x,y,t)\cdot H_\epsilon(\phi(x,y,t)).$$

The values given by the set I^i are used to compute the weighted median [29] as shown in algorithm 1 which is used as $\hat{c}$ at that particular time step t.

Input: $I(x,y,t)$, $H_\epsilon(x,y,t)$
Output: $\hat{c}$
$V = \{v_i : v_i = I(x,y,t),\ x\in[1,l],\ y\in[1,m],\ i\in[1,n],\ n = l\cdot m\}$
$W = \{w_i : w_i = H_\epsilon(x,y,t),\ x\in[1,l],\ y\in[1,m],\ i\in[1,n],\ n = l\cdot m\}$
Sort V in ascending order
$W \leftarrow W\backslash\{w_z\}\ \forall w_z = 0$
$V \leftarrow V\backslash\{v_z\}\ ,\{v_z : v_z \in V,\ \forall z \text{ where } w_z = 0\}$
$S \leftarrow \sum_{k=1}^{n} w_k,\ w_k \in W$
Find index i such that $\sum_{k=1}^{i} w_k \le \frac{S}{2}, w_k \in W$
Find index j such that $\sum_{k=j}^{n} w_k \le \frac{S}{2}, w_k \in W$
Median $M = \{v_i, v_j\}$
$\hat{c} \leftarrow \min(v_i, v_j)$

Algorithm 1: Computation of weighted median

We now show that the estimate of $\hat{c}$ based on the weighted median is a good approximation for the grey-level threshold that separates the noise from the signal, and is robust in a way that the evolution of the level set converges with increasing t.

$H_\epsilon(z)$ attains values close to zero for regions outside C and values close to one inside C. In fact, $\lim_{z\to\infty} H_\epsilon(z) = 1.0$ and $\lim_{z\to-\infty} H_\epsilon(z) = 0.0$. At the start of level set evolution, I^i covers most of Ω and therefore, $H_\epsilon(z)$ attains values close to one for most of the intensity values. This results in computation of $\hat{c}$ which is equivalent to an unweighted median for values in I^i. A median is the central point which minimizes the average of absolute deviations. Therefore, a median better represents the noise level when the data contains high intensity values that are fewer in number, and a majority of intensity values that correspond to the noise. As a result, the initial iterations of the solution suppress the intensity values that are less than the median to a constant level (the median itself). One should expect the median value to increase as the level set contracts, but since we use a regularized Heaviside function as weight for the intensity values, the weighted median converges to zero since most of I contains intensity values of zero with near-zero weight.

Other variations of estimation of $\hat{c}$ are certainly possible, but we find that a weighted median based approach results in effective noise removal with very small information loss. For instance, a value of $\hat{c}$ taken to be c_1, the mean intensity inside C, does a similar suppression but with a high signal loss compared to the former.

3) CUDA Implementation for 3D Reconstruction: Equation (9) can be solved by discretization and linearization in

 ISBN: 978-1-61208-178-6

ϕ [5]. Discretization of equation (11) in I gives

$$\frac{I_{n+1}(x,y) - I_n(x,y)}{\Delta t} = k \cdot \max\left(0, \hat{c} - I_n(x,y)\right) = \begin{cases} 0, \text{if } I_n(x,y) \geq \hat{c} \\ k \cdot (\hat{c} - I_n(x,y)), \text{otherwise} \end{cases} \quad (12)$$

with $k = \frac{1}{\Delta t}$, and $t_{n+1} = t_n + \Delta t$. The above time discretization yields the following

$$I_{n+1}(x,y) = \begin{cases} 0, & \text{if } I_n(x,y) \geq \hat{c} \\ \hat{c} - I_n(x,y), & \text{if } I_n(x,y) < \hat{c} \end{cases} \quad (13)$$

Acoustic images captured by echo-sounders are generally taken as planar image scans by moving the echo-sounder in one direction, thereby sweeping a volume. Let us denote individual images as $I(x, y, \tau)$ for images taken after every $\delta\tau$ time interval. A volume is constructed by stacking these individual images in sequence and applying geometric correction for distance $\delta\tau(v)$ between individual slices, where v is the instantaneous speed of the instrument (the current data was captured with constant unidirectional instrument velocity). It must also be noted that the individual acoustic images are obtained from a set of acoustic intensity signals along beams by a polar transformation. The level set equations for curve evolution in $\mathbb{R}^2$ extend uniformly to surface evolution in $\mathbb{R}^3$. The second differential equation also holds true for noise suppression in a volume. Therefore, it is possible to reconstruct 3D moving fishes with the level set evolution of these equations combined.

Processing a huge dataset demands that a minimum of memory is consumed. We propose to keep two volumes in the host memory, one for the intensity values (I) and the other for the signed distance function (the implicit function, ϕ). The CPU manages the memory scheduling by dividing the volumes into small subvolumes that can be processed on the GPU. We keep two small 3D textures of size $128 \times 128 \times 128$, I_{GPU} and ϕ_{GPU}. A complete level set update is divided into a set of subvolume updates. Each subvolume in the two volumes is fetched to the GPU via 3D textures (read only, but with good cache coherence). Results of computations are written to CUDA memory and then transferred back to the CPU volumes. A simplified diagram of this is shown in Figure 5.

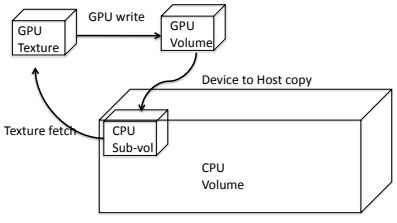


Fig. 5. Parallelization using the GPU.

CUDA exposes a set of very fast 16KB shared memory available to every multi-processor in a GPU. However, a 16 KB memory chunk is shared only between a thread block, and thus to make use of it the application must load different data for different blocks. Furthermore, the 16 KB limit poses a restriction on the amount of data that can be loaded at any point of time. Here, we use 3D textures for reading the data. Since we do not want to write back to the same texture (before a single step of filtering is complete), using the read-only 3D textures available to CUDA is a natural choice. 3D texturing has hardware support for 3D cache which accelerates any texture reads in succession. To load a 3D data (a small subset of the volume) from the global memory into the shared memory could be a little tricky and might not result in the same performance as provided by the specialized hardware for 3D texture cache. In our application, data writes are made to the global memory.

Signed distance transform is a global operation and cannot be implemented in a straightforward manner. We compute a local approximation of the Euclidean distance transform using the Chamfer distance. A narrow band distance transform is computed layer by layer using, what we call a d-pass algorithm. Every pass of the method adds a layer of distance values on the existing distance transform. The distance values are local distance increments computed in a $3 \times 3 \times 3$ neighborhood. Therefore, every single pass needs only local information to compute the distance values except at the border of the sub-volume. We therefore support every sub-volume with a one voxel cover from other adjoining sub-volumes, thereby reducing the computational domain to a volume of size $126 \times 126 \times 126$. The CPU scheduler takes care of the voxel cover. At the beginning, the interface (zero level) is initialized to a used specified bounding cuboid or a super-ellipsoid.

Computing average intensities (c_1 and c_2) is an operation that cannot be easily computed in a parallel fashion, and a reduction like method is required for the same. We employ a slightly different scheme to compute averages by using three accumulator sub-volumes on the GPU. These accumulators are essentially 3D sub-volumes of the same dimensions as of the textures. Every voxel in the accumulators accumulates (adds up), the values for H, $I \cdot H$, and $I \cdot (1 - H)$ for all the sub-volumes in the CPU volume(s). We then sum up the small sub-volume on the CPU to get the final sum and compute c_1 and c_2 values from it. Using a mixed mode CPU-GPU computation not only reduces the complexity of an inherently non-parallel operation, but also performs better by moving less expansive parts of the computation to the CPU.

Computing median on the GPU is not very straightforward since it is an order statistic and requires that the data be sorted. Therefore the computation of weighted median is very different than the one for average intensity value. Since sorting values of order of millions in every iteration of the solver is not a computationally good solution, we resort to the alternative definition of the median. A median is a value that divides the data-set into two sets of equal cardinalities. This definition is generalized for a weighted median. Therefore, for a data-set V with weights W associated with each value in the set, the

 ISBN: 978-1-61208-178-6

median value V_k is the value for which the following holds:

$$\sum_{i=0}^{k} W_i = \sum_{i=k+1}^{n} W_i$$

This equation can only be solved iteratively, starting with a guess index value k_0. In our CUDA implementation, we start with V_{k_0} to be the mean value c_1 and iteratively reach the weighted median. In every iteration, the increment $\triangle i$ for the index k_0 is computed as:

$$\triangle i = \begin{cases} \dfrac{\sum_{i=0}^{k} W_i - \sum_{i=k+1}^{n} W_i}{\sum_{i=0}^{k} W_i}, \text{ if } \sum_{i=0}^{k} W_i > \sum_{i=k+1}^{n} W_i \\ \dfrac{\sum_{i=k+1}^{n} W_i - \sum_{i=0}^{k} W_i}{\sum_{i=k+1}^{n} W_i}, \text{ if } \sum_{i=k+1}^{n} W_i > \sum_{i=0}^{k} W_i \end{cases}$$

The increment $\triangle i$ can be adaptively controlled to give results as precise as desired.

4) Solver update: A PDE update in the level set method comprises of computing the curvature energy and the external energy. In order to compute the curvature term (involving double derivatives) for a voxel in a sub-volume by centered differencing, we need information from a $5 \times 5 \times 5$ neighborhood with the current voxel at its center. Therefore, the sub-volume size needs a cover of two voxels on all sides, thus reducing the computational domain further down to $124 \times 124 \times 124$. The memory schedular performs additional computations to effectively cover the whole volume with the new setup. Once the energy terms are computed, the PDE solver kernel updates ϕ_{GPU} and uses c_1 to update I_{GPU}. These sub-volumes are then updated to the CPU main volume. It is often convenient to perform anisotropic diffusion on the input image so that the evolution of the level curve is smooth and ϕ is well behaved. Finally, the zero level surface is extracted from the evolved ϕ using the Marching Cubes method.

III. Results

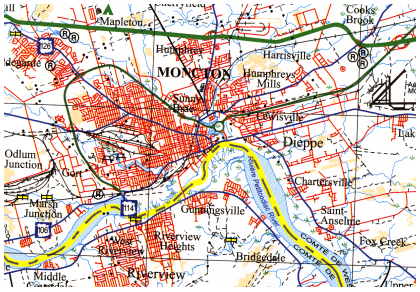
We observe that using the GPU for the parallelization of the processing of the imagery induces an overall speed-up that ranges between 10 and 20 times over CPU algorithms.

A. High signal to noise ratio imagery

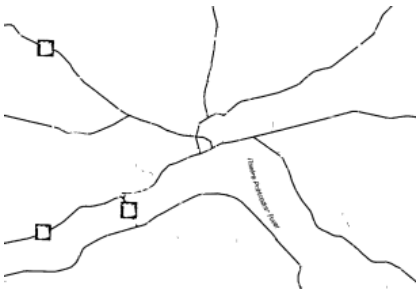
In this section, we show results of our new combined GPU-based methodology on high signal to noise ratio imagery (scanned maps and satellite imagery).

The extraction of a road network from a scanned map is shown in Figure 6.

In the following example, the coastline is extracted as the boundary of the selected object. The accuracy of the coastline rendition depends on the spatial resolution of the imagery. The beach of Seychelles shown in Figure 7(a) is mainly sandy and



(a) Scanned map.



(b) Dark blue class in the segmented image (corresponding to dark blue segments from the scanned map.

(c) Medial axis of the polygons represented in (c).

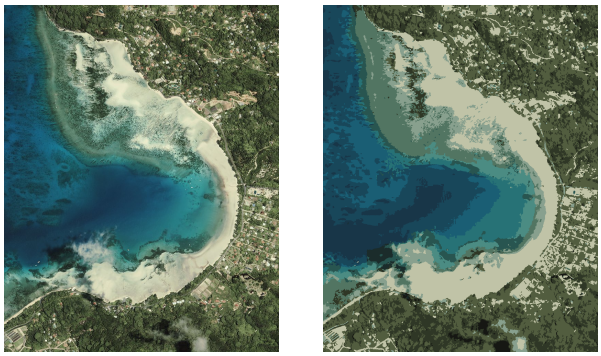
Fig. 6. City Map of Moncton, New Brunswick, Canada.

shows a wide variation in the ocean color. The color variation is primarily due to the depth of water. Figure 7(b) shows the result of the segmentation using mean shift and Voronoi skeletonization on Figure 7(a). Regions in Figure 7(b) that form the ocean are combined to extract the ocean boundary (see Figure 7(c)). The extended coastline representing the boundary of the sand beach is shown in Figure 7(d). The extended coastline shows the presence of a number of small polygons, since the roads connecting the beach have the same color value in the image, and are included in the selection.

B. Low signal to noise ratio imagery

We present experimental results on 2D multi-beam echo sounder (acoustic) images to show that the suppression scheme works well on such images. Figure 8 shows evolution of the level set. The parameters for this evolution were chosen to be: $\mu = 0.0005$, $\nu = 0$, $\lambda_1 = \lambda_2 = 1$, and $\epsilon = 2.5$. It can be seen that the original image suffers from speckle noise and that the final zero level contour approximates the fish boundaries very

 ISBN: 978-1-61208-178-6

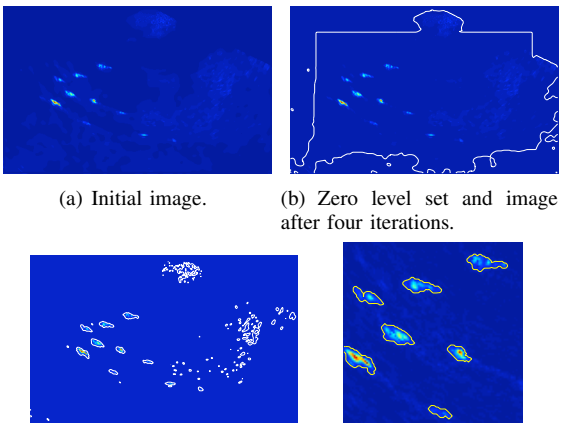


(a) Satellite image of Seychelles. (b) Over-segmented image.

(c) Extracted ocean boundary. (d) Extracted beach boundary.

Fig. 7. Feature polygon extraction from the satellite image of Seychelles.

well.

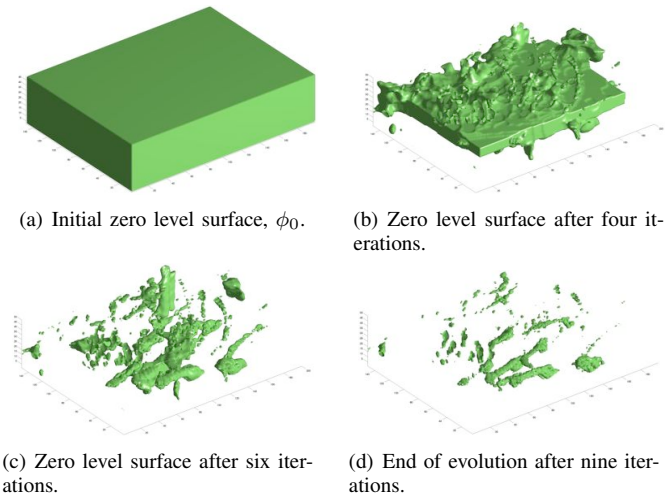


(a) Initial image. (b) Zero level set and image after four iterations.

(c) End of evolution after 16 iterations. (d) Final contour shown on part of the original image.

Fig. 8. Level set evolution on sample image, $\epsilon = 2.5$.

We next show results of application of the level set equation and the noise suppression scheme on a small 3D multi-beam echo sounder (acoustic) volume of size $150 \times 100 \times 50$. Fish intensities can be identified in dark green against a noisy background. The level set equation was initialized with the zero level set of ϕ_0 as the bounding box of the volume. The level set is then allowed to evolve with parameters,



(a) Initial zero level surface, ϕ_0. (b) Zero level surface after four iterations.

(c) Zero level surface after six iterations. (d) End of evolution after nine iterations.

Fig. 9. Level set evolution on sample volume, $\epsilon = 1.0$.

$\mu = 0.0005$, $\nu = 0$, $\lambda_1 = \lambda_2 = 1$, and $\epsilon = 1.0$. Figure 9 shows the evolution at different time steps and the final level surface.

We test the CUDA solver on a larger volume of size $686 \times 1234 \times 100$. This volume uses about 470 MB of CPU memory along with the same amount of memory consumed by the signed distance field. We test our implementation with the mobile GPU, GeForce 8600M GT (NVDIA CUDA compute capability of 1.1) with 256 MB of memory on a Mac OS X notebook with 2 GB of host memory. The total number of iterations required until convergence were 29, with a compute time of about 52 seconds per iteration (32 seconds without median computation). The signed distance field was reconstructed in a narrow band of width 20 voxels in every iteration.

In order to compare the 2D and 3D reconstructions, we show an overlay of 2D curves over the extracted 3D surface. This is shown in Figure 10. The results agree very well when the 2D image contains high intensity objects. The acoustic images were taken by scanning fishes in an aquarium and the images corresponding to the bottom of the aquarium (time slices with higher depth, 30 to 50 in Figure 10) contain almost no fishes. Therefore, these images contain very little useful information. The 2D level set evolution fails to detect fishes in these images. Therefore, the 3D results should be trusted since the 2D reconstruction does not consider information present in other image planes. We would like to comment that a ground truth segmentation is not practically possible for open sea. Evaluation of the extracted fish trails/schools by domain experts is under process because of marine surveys.

IV. CONCLUSIONS

This research work succeeds in achieving its primary goals by designing an effective GPU-based highly parallel methodology for automated vectorization of imagery. Based on the methodology, an interactive software application has been developed. It incorporates object extraction from input images

 ISBN: 978-1-61208-178-6

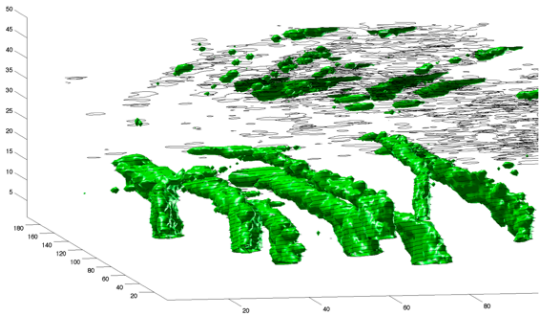


Fig. 10. Comparison of zero level 2D curves with the zero level 3D surface.

using color image segmentation or level sets evolution. The application allows either automated extraction of all features or manual selection of multiple objects for semi-automated extraction of targeted feature classes. It is worth noticing here that the extracted feature might not be a complete map object in the scanned image. This is due to labels and other features drawn over the feature of interest on a paper map.

Applicability of the Voronoi-based methodology to satellite imagery has been shown by extracting natural features like coastlines. Experiments done with satellite imagery show acceptable results too. Coastline delineation, snow cover mapping, cloud detection, and dense forest mapping are a few areas where satisfactory results can be obtained. Applicability of the level set evolution methodology has been shown on underwater acoustic images. Current work focuses on level set evolution based Digital Terrain Model (DTM) generation from Light Detection And Ranging (LIDAR) data sets. Any imagery can be processed using the above methodology: not only raster images, but also 2D and 3B beam-formed datasets like Computer Tomography or multi-beam echo sounder data.

REFERENCES

[1] F. Anton, D. Mioc, and A. Fournier, "2D Image Reconstruction using Natural Neighbour Interpolation," *The Visual Computer*, vol. 17, no. 3, pp. 134–146, 2001.

[2] S. Bagli and P. Soille, "Morphological automatic extraction of coastline from pan-European Landsat TM images," in *Proceedings of the Fifth International Symposium on GIS and Computer Cartography for Coastal Zone Management*, vol. 3, Genova, 2003, pp. 58–59.

[3] G. Bo, S. Delleplane, and R. D. Laurentiis, "Coastline extraction in remotely sensed images by means of texture features analysis," in *Geoscience and Remote Sensing Symposium, IGARSS '01*, vol. 3, Sydney, NSW, Australia, 2001, pp. 1493–1495.

[4] T. Chan and L. Vese, "A level set algorithm for minimizing the Mumford-Shah functional in image processing," *IEEE/Computer Society Proceedings of the 1st IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pp. 161–168, 2001.

[5] ——, "Active Contours Without Edges," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 10, no. 2, 2001.

[6] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[7] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: color image segmentation," in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. Washington, DC, USA: IEEE Computer Society, 1997, pp. 750–755.

[8] ——, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[9] K. Di, J. Wang, R. Ma, and R. Li, "Automatic Shoreline Extraction from High-Resolution Ikonos Satellite Imagery," in *Proceeding of ASPRS 2003 Annual Conference*, vol. 3, Anchorage, Alaska, 2003.

[10] K. R. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Zoology*, vol. 18, no. 3, pp. 259–278, 1969.

[11] C. M. Gold, "Crust and anti-crust: A one-step boundary and skeleton extraction algorithm," in *Symposium on Computational Geometry*. New York, NY, USA: ACM Press, 1999, pp. 189–196.

[12] C. M. Gold and J. Snoeyink, "A one-step crust and skeleton extraction algorithm," *Algorithmica*, vol. 30, no. 2, pp. 144–163, Jun 2001.

[13] C. M. Gold and D. Thibault, "Map generalization by skeleton retraction," in *Proceedings of the 20th International Cartographic Conference (ICC)*, Beijing, China, August 2001, pp. 2072–2081.

[14] N. K. Govindaraju, B. Lloyd, W. Wang, M. Lin, and D. Manocha, "Fast computation of database operations using graphics processors," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*. New York, NY, USA: ACM, 2005, p. 206.

[15] L. Guibas and J. Stolfi, "Primitives for the manipulation of general subdivisions and the computation of Voronoi Diagrams," *ACM Transactions on Graphics*, vol. 4, no. 2, pp. 74–123, 1985.

[16] M. Lianantonakis and Y. Petillot, "Sidescan sonar segmentation using active contours and level set methods," *Oceans 2005-Europe*, vol. 1, 2005.

[17] H. Liu and K. C. Jezek, "A Complete High-Resolution Coastline of Antarctica Extracted from Orthorectified Radarsat SAR Imagery," *Photogrammetric Engineering and Remote Sensing*, vol. 70, no. 5, pp. 605–616, 2004.

[18] R. Malladi and J. Sethian, "Image Processing Via Level Set Curvature Flow," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 92, no. 15, pp. 7046–7050, 1995.

[19] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. 42, no. 5, pp. 577–685, 1989.

[20] R. L. Ogniewicz, "Skeleton-space: A multiscale shape description combining region and boundary information," in *Proceedings of Computer Vision and Pattern Recognition, 1994*, 1994, pp. 746–751.

[21] S. Osher and R. Fedkiw, *Level sets and dynamic implicit surfaces*. Springer New York, 2003.

[22] S. Osher and J. Sethian, "Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.

[23] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[24] J. Sethian, "Theory, algorithms, and applications of level set methods for propagating interfaces," *Acta Numerica 1996*, pp. 309–395, 1996.

[25] O. Sharma and F. Anton, "CUDA based Level Set Method for 3D Reconstruction of Fishes from Large Acoustic Data," in *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision ; 17*, 2009.

[26] O. Sharma, D. Mioc, and F. Anton, "Feature extraction and simplification from colour images based on colour image segmentation and skeletonization using the quad-edge data structure," in *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG 2007*. University of West Bohemia, Plzen, Czech Republic, 2007, pp. 225–232.

[27] O. Sharma, D. Mioc, and A. Habib, "Road extraction from satellite imagery using fractals and morphological image processing," in *Proceedings of the 13th International Conference on Geoinformatics*, Toronto, Canada, 2005.

[28] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. PWS publishing, 1999.

[29] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: a tutorial," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on]*, vol. 43, no. 3, pp. 157–192, 1996.

 ISBN: 978-1-61208-178-6