Feature Extraction and Simplification from colour images based on Colour Image Segmentation and Skeletonization using the Quad-Edge data structure

Ojaswa Sharma
Department of Geodesy and Geomatics Engineering,
University of New Brunswick,
Fredericton, N.-B., E3B 5A3,
Canada
ojaswa.sharma@unb.ca

Darka Mioc
Department of Geodesy and Geomatics Engineering,
University of New Brunswick,
Fredericton, N.-B., E3B 5A3,
Canada
dmioc@unb.ca

François Anton
Department of Informatics and Mathematical Modelling,
Technical University of Denmark,
Building 321,
2800 Kgs. Lyngby, Denmark
fa@imm.dtu.dk

Abstract

Region features in colour images are of interest in applications such as mapping, climatology, change detection, medicine, etc. This research work is an attempt to automate the process of extracting feature boundaries from colour images. This process is an attempt to eventually replace manual digitization process by computer assisted boundary detection and conversion to a vector layer in a spatial database. In colour images, various features can be distinguished based on their colour. The features thus extracted as object border can be stored as vector maps in a spatial database after labelling and editing. Here, we present a complete methodology of the boundary extraction and skeletonization process from colour imagery using a colour image segmentation algorithm, a crust extraction algorithm and our new skeleton extraction algorithm. We also present a prototype application for completely automated or semi-automated processing of (satellite) imagery and scanned maps with an application to coastline extraction. Other applications include extraction of fields, clear cuts, clouds, as well as heating or pollution monitoring and dense forest mapping among others.

Keywords: colour image segmentation, polygon feature extraction, satellite imagery

1 INTRODUCTION

A lot of work on feature 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 [15]. A morphological segmentation based automated approach for coastline extraction has been suggested in [2]. Di et al. use the image segmentation algorithm by [7] to segment an image and detect the shoreline [9].

1.1 Image Segmentation

The segmentation method adopted here is the one provided by [6] which is based on feature space analysis.

Feature space analysis is used extensively in image understanding tasks. Comaniciu and Meer [6] 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 the case of an image. As an example, if we consider a colour 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 colour vector corresponding to a pixel from the image can be represented as a point in the feature space.

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

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

The *mean shift* vector at a location $\overrightarrow{x}$ is defined as:

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

In his work, Cheng shows that the mean shift vector, the vector of difference between the local mean and the center of the window $K(\overrightarrow{x})$, is proportional to the gradient of the probability density at $\overrightarrow{x}$ [5]. Thus mean shift is the steepest ascent with a varying step size that is the magnitude of the gradient. The fundamental use

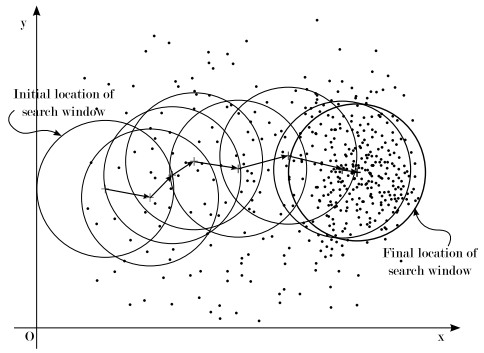


Figure 1: Mode seeking using mean shift algorithm.

of mean shift is in seeking the modes (i.e. the most frequent value occurring in a sampling of a random variable) that give regions of high density in any data. 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 [7]. The authors also prove that the mean shift vector converges to zero and eventually reaches the basin of attraction of that mode. This is shown graphically in Figure 1 where the initial position of the kernel window is chosen at a location far from the mode of the data and the window is moved by a magnitude equal to that of the mean shift vector at the current window location in each step. It can be seen that the mean shift vectors gradually converge as the window moves near the maximum density region.

In their research work, Comaniciu and Meer state a simple, adaptive steepest ascent mode seeking algorithm [6].

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 colour image segmentation.

The method described in [6] 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 colour, 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. For the implementation of the segmentation algorithm, the authors provide three segmentation resolution classes:

1. **Undersegmentation** refers to the lowest resolution with a minimum number of colours and only dominant regions of the image. The three parameters for this class are:
   $(0.4\sigma, 400, 10)$.
2. **Oversegmentation** refers to intermediate resolution and represents objects with some level of detail. The three parameters for this class are: $(0.3\sigma, 100, 10)$.
3. **Quantization** refers to the highest resolution and produces image with all the important colours with no object connectivity requirement. The three parameters for this class are: $(0.2\sigma, 50, 0)$.

Figure 2 shows the results of this segmentation algorithm on a natural image. Note the variation in number of colours for each segmentation type.

Later, Comaniciu and Meer provide an improvement [7] 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 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. In this research, since we are dealing with scanned maps, the simple mean shift based segmentation algorithm provides satisfactory results.

1.2 Boundary Extraction

The work presented in [1] leads to the extraction of object boundary from a set of sufficiently well sampled data points. The vertices of the Voronoi diagram approximate the medial axis of a set of sample points from a smooth curve. Vertices of the Voronoi diagram of the sample points were inserted into the original set of sample points and a new Delaunay triangulation was computed [1]. The circumcircles of this new triangulation approximate empty circles between the original boundary of the object and its skeleton. Thus, any Delaunay

(a) Original image with 108440 colours

(b) Undersegmented image with 8 colours

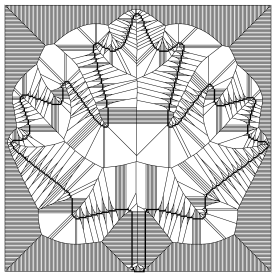
(c) Oversegmented image with 34 colours

(d) Quantized image with 49 colours

Figure 2: Colour image segmentation by [6]

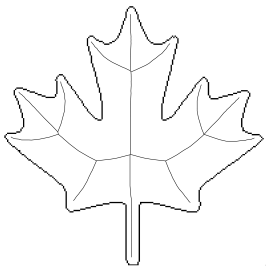
edge connecting a pair of the original sample points in the new triangulation is a part of the border [1].

Further research by [10] leads to a One-step border (crust) extraction algorithm. In a Delaunay triangulation, each Delaunay edge is adjacent to two triangles and the circumcenters of these triangles are the Voronoi vertices. A Voronoi edge connecting these two circumcenters is the dual edge to the Delaunay edge considered here. According to [10], a Delaunay edge is a part of the border if it has a circle that does not contain any Voronoi vertex. It is sufficient to test only the vertices of the dual Voronoi edge. The test is the standard *InCircle* test. Considering two



(a) The Voronoi diagram

(b) Anti-crust



(c) Skeleton

Figure 3: Skeleton as seen as the anti-crust.

triangles (p,q,r) and (r,q,s) sharing an edge (q,r) in a Delaunay triangulation and letting v be a vector orthogonal to edge $(r-q)$ in clockwise order, the test is: $(\overrightarrow{qs}\cdot\overrightarrow{rs})(\overrightarrow{qp}\cdot\overrightarrow{rp})\geq-(\overrightarrow{rs}\cdot v)(\overrightarrow{qp}\cdot v)$. This test will be true for an edge in the border set. Furthermore, those Delaunay edges that are not part of the border set have their dual Voronoi edges as being part of the skeleton.

1.3 Skeleton Extraction

Popular methods of skeleton extraction are thinning using mathematical morphology [13, chap. 9] and skeletonization using distance transform [4]. This research is concerned with skeleton extraction using the Voronoi diagram [1, 10, 16, 18].

The work presented in [1] shows that the "crust" or the boundary of a polygon can be extracted from an unstructured set of points provided the data points are well sampled. Gold and Snoeyink [11] further simplify their method and show that the boundary can be extracted in a single step (see section 1.2). Gold [10] discusses about "anti-crust" in context of skeleton extraction citing a brief introduction of this term in [1]. The idea behind getting skeleton is that a Voronoi edge is a part of the skeleton if its corresponding dual Delaunay edge is not a part of the border set (crust) and it lies completely within the selected object. Thus, selecting the Voronoi edges lying inside the selected object that are dual of the non-crust Delaunay edges should give us the skeleton (see Figure 3). The Voronoi edges thus selected form a tree structure called the "anti-crust" [10] that extend towards the boundary but do not cross it.

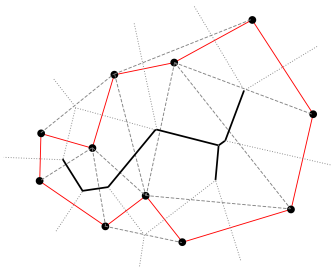


Figure 4: Anti-crust from the crust.

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 [10], it is easy to identify the Voronoi edges belonging to the anti-crust. In Figure 4, 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. 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. $e.Rot().Org \in I$
3. $e.Rot().Dst \in I$

Where $e.Rot().Org$ is the origin coordinate of edge $e.Rot()$, $e.Rot().Dst$ is the destination coordinate of 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 coordinates 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.

1.4 Skeleton Pruning

The "hairs" around the skeleton result from 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 [10]. A skeleton retraction scheme suggested by [12] gets rid of the hairs and also results in smoothing of the boundary of the object. Ogniewicz [17] presents an elaborate skeleton pruning scheme based on various residual functions. Thus a hierarchic skeleton is created which is good for multiscale representation. The problem of identifying skeleton edges now reduces to reasonably prune the anti-crust.

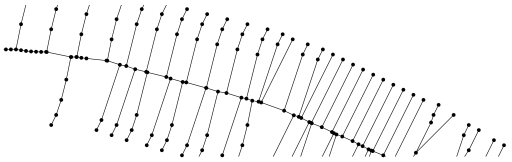


Figure 5: Hair around the skeleton composed of multiple edges.

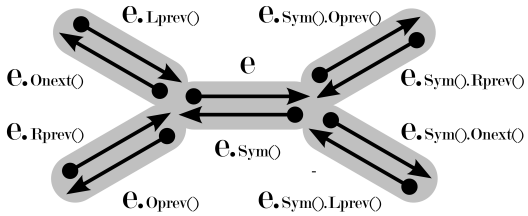


Figure 6: Accessing neighboring edges in a quad-edge.

Gold and Thibault [12] 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 [12] 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 5). Therefore, single retraction is not be sufficient to provide an acceptable skeleton.

2 OUR SKELETON APPROACHES

This is our main contribution together with the integration of this skeleton approach with existing techniques described above. We address the problem mentioned in Section 1.4 by pruning the leaf edges instead of retracting the leaf nodes. Leaf edge pruning produces satisfactory results and requires only two or three levels of pruning. Before pruning the leaf edges, these must be identified in the anti-crust using the operations provided by the quad-edge data structure (see Figure 6).

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:

$$e.Oprev() \notin T \text{ And } e.Onext() \notin T$$

Or

$$e.Sym().Oprev() \notin T \text{ And } e.Sym().Onext() \notin T.$$

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.

Finally, we have found a labelling scheme for the vertices of the border set that allows us to get the skeleton

without hairs and gaps from the inner Delaunay triangulation. We identify the vertices of the border set that are incident to two Delaunay edges of the border set. We call them branching vertices. We start to give numerical labels to the vertices of the border set starting at a branching vertex, and we increment the label each time we encounter a new branching vertex. On the example of Figure 4, there are three branching vertices. We get three labels 1, 2 and 3, and three branches for the skeleton. We keep only Voronoi edges between vertices having different labels.

3 AUTOMATED APPROACH TO SKELETONIZATION

The general approach adopted here is:

1. Segment a colour image into prominent objects.
2. Ask the user if all the objects are to be processed independently (automatic process) or an individual object (semi-automatic process) selected.
3. Collect sample points for each object to be processed.
4. Extract the skeletons using Delaunay/Voronoi diagram based algorithm.

Once objects are defined as homogeneous regions by the segmenter, the next step is to either select them all or some of them. To achieve this, the user is allowed to select a region on the image. If an object is composed of more than one regions then multiple object selection can be made and regions combined to form a single object. A wrongly selected region can be removed from the selection. The user input is processed and the selected region is highlighted and selected for next processing.

Once we have an object or all the objects chosen from an image, the next step is to sample its boundary in order to generate points used to construct the Delaunay triangulation. In order to automatically generate these sample points, edge pixels that are returned by a morphological edge detector are used. Using edge pixels also helps in generating a dense sampling which is required to give a better approximation of the skeleton [1]. Morphological edge detection on the binary image containing the selected object is performed and the edge pixels are then sequentially inserted into the Delaunay triangulation. The triangulation is updated after every insertion (using the incremental algorithm).

The Delaunay triangulation of the sample points is computed using the incremental algorithm given by [14] which is stored in the quad-edge data structure. This is followed by computation of the Voronoi vertices for all faces of the triangulation. The boundary of the object is extracted using the criterion given by [10]. The edges in the Delaunay triangulation are analyzed and flagged as being part of the boundary.

4 RESULTS WITH SATELLITE IMAGERY

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.

Since the system is designed to make use of homogeneity in colors of objects, natural objects are better suited for our analysis. A few cases of coastline extraction have been considered here. Figure 7 shows the complex coastline of Guinea Bissau. Segmentation results in four colours that define the water body out of which three define the coastline. Multiple selection enables combining these three regions together to form the complete coastline as shown in Figure 7(d).

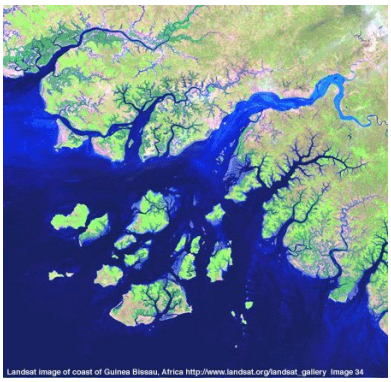
Satellite image of Lake Jempang, East Kalimantan, Indonesia (20 Jul 2002) obtained on-line from Centre for Remote Imaging, Sensing and Processing (http://www.crisp.nus.edu.sg/monthly_scenes /y2002/Jul02_i.html) is shown in Figure 8(a). The extracted lake boundary is shown in Figure 8(d).

The map in Figure 9(a) is obtained processing the photograph from the proposal for Decision Support for Flood Event Prediction and Monitoring (FEPM) project set forth by the Emergency Measures Organization, New Brunswick, Canada. It shows flood risk mapping from the 1973 flood in the Saint John river in Fredericton, New Brunswick and the risk of flooding in the area in a time span of 20 years (extracted in Figure 9(c) and (d)), 100 years (extracted in Figure 9(e) and (f)) and actual flood extent occurred in the year 1973 (extracted in Figure 9(g) and (h)). The flood plains are extracted as polygon vector boundaries from crust of the selected object.

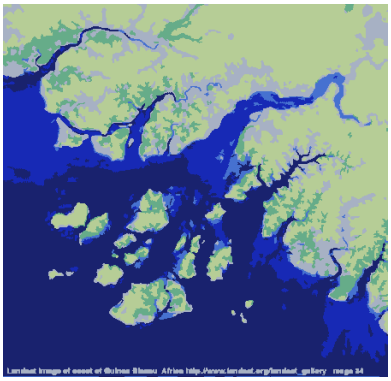
Figure 10(a) shows small portion of a map of the city of Moncton, New Brunswick, Canada (Map Number - 21I/02) scanned from the *New Brunswick Atlas* (second edition). The map is scanned at 300 DPI (dots per inch) resolution and 24-bit colour depth. The map mainly consists of three types of roads: national highway (green), highway (blue) and streets (red). Crisp road edges with well defined road patches can be seen in the segmented image (Figure 10(b)) that are extracted as skeletons (see Figures 10(d) and (f)). The skeletons show the presence of minor clutter due to misclassification during the segmentation process. Such a problem can be easily rectified in a GIS environment while manually editing the data after import.

5 TIME COMPLEXITY

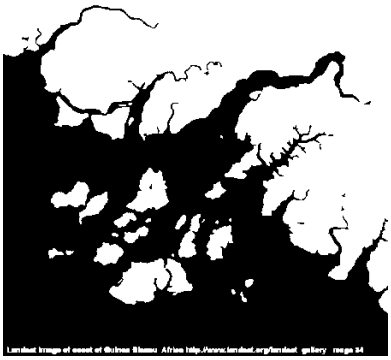
It is important to analyse the complete procedure for time complexity. Since the overall complexity will depend on the intermediate steps, these are first analysed individually. Comaniciu [8, p. 21] shows that the complexity of the probabilistic mean shift type algorithm



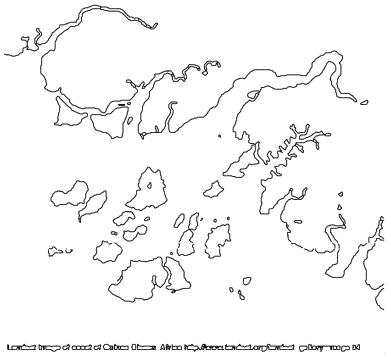
(a) Satellite image of Guinea Bissau



(b) Over-segmented image



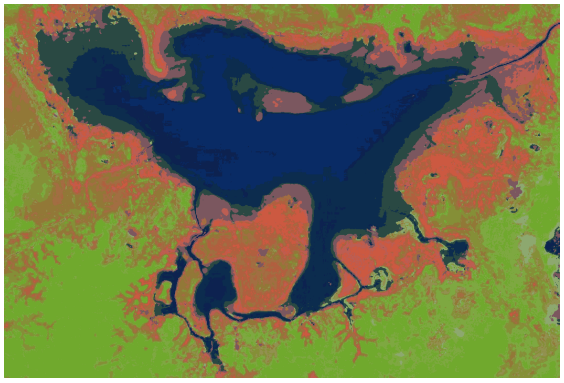
(c) North Atlantic ocean



(d) Extracted coastline

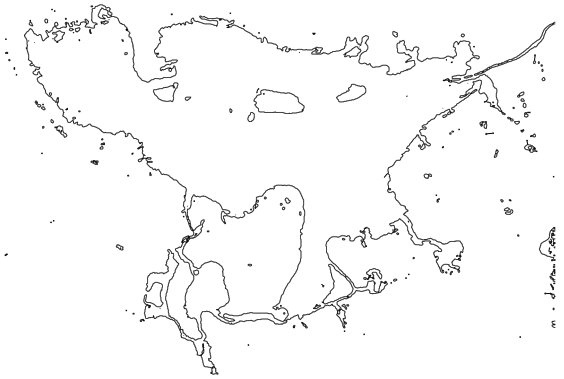
Figure 7: Feature boundary extraction from the satellite image of Guinea Bissau.

(a) Lake Jempang



(b) Segmented image

(c) Selected lake



(d) Lake boundary

Figure 8: SPOT satellite image of Lake Jempang, East Kalimantan, Indonesia (20 Jul 2002).

that is employed in the segmentation algorithm [6] is $O(mn)$, with $m \ll n$ where n is the number of pixels in the input image (or the number of feature vectors in the feature space) and m is the number of vectors in initial feature palette or clusters. Comaniciu [8, p. 29] claims that the segmentation algorithm is linear with the number of pixels in the image.

Object selection is implemented as a simple search of pixel colour values compared with the user selected points on the image. Therefore, it is $O(n)$ with the number of pixels n in the image.

Binary edge detection is implemented as subtraction of two binary images. One of the binary images is always the original image while the other is either a dilated or an eroded image of the original. For an image I of size $x \times y$ and a structural element of size $k \times l$, computation of either erosion or dilation requires $x \times y$ iterations, each requiring $(k \times l) + 1$ comparisons. Therefore, the time complexity is linear with number of pixels of the input binary image and linear with the area of the structural element. Overall time complexity can be said to be $O(mn)$ where n is number of pixels in the image and m is number of pixels in the structuring element. Further, in the structural element, only locations having a value of 1 are considered in the computations. In our case, the application uses a 3×3 structural element with only five values being 1's in the mask (i.e., $m = 5$). Therefore, we can safely say that edge detection is $O(n)$ with the number of pixels n in the image.

6 CONCLUSIONS

This research work succeeds in achieving its primary goals by designing an effective methodology for automated vectorization of features in colour images. The methodology enables extraction of boundaries of an object in a single step.

Based on the methodology, an interactive software application has been developed. It incorporates object extraction from input images using colour image segmentation. Segmentation based on clustering using a mean shift algorithm in feature space has been adopted here. The mean shift algorithm is a popular and robust method of clustering and provides good results in segmentation. The application allows selection of multiple objects for extraction of the boundary.

The applicability of the methodology to colour images has been shown by extracting natural or artificial features from (satellite) images. Applicability of the developed methodology can be easily extended to natural colour satellite imagery to extract homogeneous features. Coastline delineation, snow cover mapping, cloud detection, and dense forest mapping are a few areas where satisfactory results can be obtained.

REFERENCES

[1] N. Amenta, M. Bern, and D. Eppstein. The crust and the β-skeleton: Combinatorial curve reconstruction. *Graphical models and image processing: GMIP*, 60(2):125–135, 1998.

[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*, volume 3, pages 58–59, Genova, 2003.

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

[4] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27(3):321–345, 1984.

[5] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[6] 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)*, pages 750–755, Washington, DC, USA, 1997. IEEE Computer Society.

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

[8] Dorin I. Comaniciu. *Non-Parametric Robust Methods for Computer Vision*. PhD thesis, Rutgers, The State University of New Jersey, 2000.

[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*, volume 3, Anchorage, Alaska, 2003.

[10] Christopher M. Gold. Crust and anti-crust: A one-step boundary and skeleton extraction algorithm. In *Symposium on Computational Geometry*, pages 189–196, New York, NY, USA, 1999. ACM Press.

[11] Christopher M. Gold and Jack Snoeyink. A one-step crust and skeleton extraction algorithm. *Algorithmica*, 30(2):144–163, Jun 2001.

[12] Christopher M. Gold and David Thibault. Map generalization by skeleton retraction. In *Proceedings of the 20th International Cartographic Conference (ICC)*, pages 2072–2081, Beijing, China, August 2001.

[13] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Procesisng*. Prentice Hall, 2 edition, 2002.

[14] P.J. Green and R. Sibson. Computing dirichlet tessellations in the plane. *The Computer Journal*, 21(2):168–173, 1977.

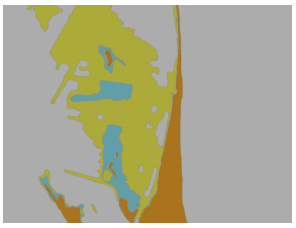
[15] H. Liu and K. C. Jezek. A complete high-resolution coastline of antarctica extracted from orthorectified radarsat sar imagery. *Photogrammetric Engineering and Remote Sensing*, 70(5):605–616, 2004.

[16] R. Ogniewicz. Automatic medial axis pruning by mapping characteristics of boundaries evolving under the euclidean geometric heat flow onto Voronoi skeletons. Technical Report 95-4, Harvard Robotics Laboratory, 1995.

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

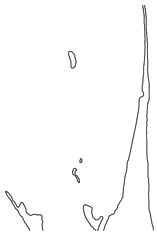
[18] R. L. Ogniewicz and O. Kübler. Hierarchic Voronoi skeletons. *Pattern Recognition*, 28(3):343–359, 1995.

(a) Flood extents in the city of Fredericton



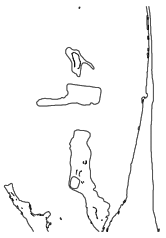
(b) Over segmented image 11 colours

(c) 20 years flood zone



(d) extracted boundary

(e) 100 years flood zone

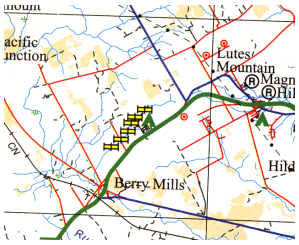


(f) extracted boundary

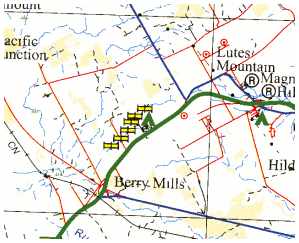
(g) 1973 flooded area

(h) extracted boundary

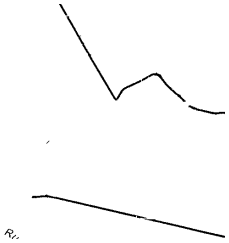
Figure 9: A portion of photographed flood risk map showing flood extents.



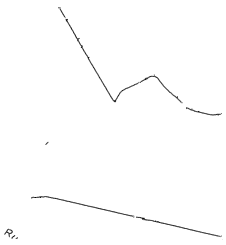
(a) Map with 62800 colours



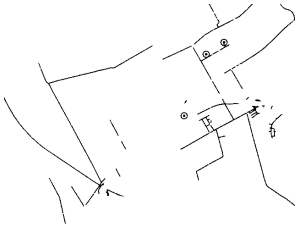
(b) Segmented image with 10 colours



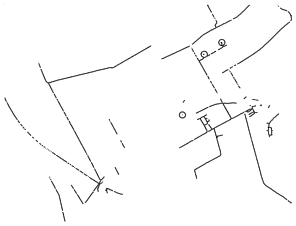
(c) Highway



(d) Skeleton of highway



(e) Streets



(f) Skeleton of streets

Figure 10: Extraction of various roads from a map.