Street Centreline Generation with an Approximated Area Voronoi Diagram

Steven A. Roberts[1], G. Brent Hall[2], and Barry Boots[1]

[1] Department of Geography and Env. Studies, Wilfrid Laurier University
sroberts@wlu.ca, bboots@wlu.ca

[2] Faculty of Environmental Studies, University of Waterloo
gbhall@fes.uwaterloo.ca

Abstract

This paper presents a methodology to generate a topologically correct street centreline from city block boundaries using ArcGIS software. The approach utilises Sugihara's (1992) point approximation algorithm as a starting point to create an area Voronoi diagram which forms the basis of the centreline. A recursive method is introduced to schematize the geometry of the Voronoi medial axis. The approach is applied to data from the City of Rosario, Argentina. The paper concludes with suggestions for further enhancements to the approach that have, among other things, the potential to automate attribution to network segments using adjacent polygon attributes.

1 Introduction

The use of spatial information technologies in an expanding range of application areas has meant that issues of data availability, quality and usefulness are now more prominent than in the past. Possibly the largest growth areas in the use of these technologies are in the private sector, spurred by the rapid emergence of mobile location-based service mapping (commonly referred to as LBS, or m-commerce), the continued growth of e-commerce applications, and the re-emergence of traditional operations research applications for routing, scheduling, fleet management, inventory tracking and delivery and other transport system-based logistics. These application areas have significantly increased the need for available, high quality (in terms of completeness, currency and accuracy) spatial data.

Central to the data needs of many of the new generation of Geographic Information Systems (GIS) applications is a connected, up-to-date, and topologically correct street network. Without a proper transportation network most, if not all, transportation logistics models and m- and e-commerce applications are rendered practically useless. However, with a transport network layer in place, multiple attributes, such as left and right and to and from address strings, carriageway capacities, condition and operational information as well as point feature location attributes (business, warehouse, pick up and delivery points etc.) are able to be used for m- and e-commerce. Hence, it is of considerable importance, especially in large metropolitan areas, to create and maintain these aspects of a GIS database.

A substantial literature has emerged in the past half decade from image processing that seeks to refine methods for automated and semi-automated street network extraction from aerial photography and satellite remote sensing imagery, cf. Couloiger and Ranchin (2000), Fiset and Cavayas (1997) and Cao and Qin (1998). This literature has applied a variety of modelling approaches and algorithms to the problem, with the general objective of updating road network mapping relative to the rapid pace of urban development. Surprisingly, however, there is very little published research on the issue of automated or semi-automated single line street network (slsn) creation and maintenance within the vector data domain. One possible explanation for this is that street centrelines can be manually digitized from the city block casings that automatically define the street polygon(s). However, this approach is very labour intensive and prone to operator error. While the presence of street polygons assists with human map visualisation and way-finding through placement of street names on maps, it is of no practical use for transportation logistics and other applications that require a fully connected, attributed single line network with directional topology.

This paper presents an innovative approach to street network generation from block casings using Sugihara's (Okabe et al., 2000, p. 287) point approximation algorithm as a starting point to create an area Voronoi diagram that forms the basis of the road centreline. A recursive method is introduced to schematize the geometry of the Voronoi medial axis, primarily to enforce "T-shaped" local configurations at T-intersections. The method is based on the removal of subsets of the set of centroids from the first generation Voronoi cells and using this modified set of centroids as the second generation generator set. The method is implemented in ArcGIS software using a straightforward macro program and some embedded Perl code to quicken processing time Eykamp (1999). The latter code replaces the macro language's sluggish implementation of looping via cursors and improves processing time in this case by a little over 30%. Results are compared against against other research Alnoor and Martinez (1996), Radke and Flodmark (1999) and commercial solutions such as ArcGIS's built-in centerline routine (ESRI, 2000, p. 9). The area Voronoi method is shown to be robust, with significantly more useful results produced compared with ArcGIS's built-in algorithm. With additional processing of the area Voronoi results, address strings could be extracted from city block faces to allow the almost immediate deployment of new generation GIS-based transportation logistics, m- and e-commerce applications.

The paper is structured as follows. In the next section the basis for the area Voronoi approach is described. The study area of the City of Rosario, Argentina is then briefly described. Next the methodology used is specified and the algorithm provided. The paper concludes with a short discussion of the area Voronoi results and suggestions for future enhancements.

2 The Area Voronoi Diagram and related structures

Consider a set of areas $A = \{A_1, \ldots, A_n\}$, $2 \le n < \infty$, in $\mathbb{R}^2$. Assume that A_i is a connected closed set and that the areas do not overlap each other. A_i need not be convex and may have holes in which another area may exist. Define the distance from a point p to an area A_i as the shortest Euclidean distance from p to A_i, i.e.,

$$d_s(p, A_i) = \min_{x_i} \{\|\mathbf{x} - \mathbf{x}_i\| \mid \mathbf{x}_i \in A_i\}, \tag{1}$$

where $\mathbf{x}$ and $\mathbf{x}_i$ are the location vectors of p and $p_i \in A_i$, respectively. Using this distance, we define a set by

$$V(A_i) = \{p \mid d_s(p, A_i) \le d_s(p, A_j)\ \ , j \ne i,\ i, j \in I_n\}. \tag{2}$$

We call this set the area Voronoi region associated with A_i , and the set of area Voronoi regions, $\mathcal{V}(A) = \{V(A_1), \ldots, V(A_n)\}$, the area Voronoi diagram generated by the generator set A. In many applications A is confined to a bounded region S in $\mathbb{R}^2$. Thus, we also consider the bounded area Voronoi diagram given

$$\mathcal{V}(A)_{\cap S} = \{V(A_1) \cap S, \ldots, V(A_n) \cap S\}. \tag{3}$$

As Okabe et al. (2000, p.187) demonstrate, $\mathcal{V}(A)$ can be constructed from its corresponding Voronoi line diagram $\mathcal{V}(L(A))$, where $L(A) = \{\partial A_1, \ldots, \partial A_n\}$ are the boundaries of A. Alternatively, $\mathcal{V}(A)$ can be approximated by replacing A_i with a finite number of points that approximate A_i, constructing the ordinary Voronoi diagram of the points, and removing any superfluous Voronoi edges and Voronoi vertices, (Okabe et al., 2000, p.287). We refer to this construction as the approximated area Voronoi diagram.

One structure which is related to the area Voronoi diagram in $\mathbb{R}^2$ is the medial axis which can be defined for a connected geometric figure C. The medial axis $M(C)$ is defined as the locus of all points that have at least two closest points on the boundary of C. Alternatively, $M(C)$ can be defined as the locus of the centres of all interior maximal disks of C, where a maximal disk is a disk which is not contained by any other disks that are included in C. More formally, the medial axis of C can be defined as

$$M(C) = \left\{ \mathbf{x} \mid \|\mathbf{x}-\mathbf{x}_i\| = \|\mathbf{x}-\mathbf{x}_j\| = \min_{y} \{\|\mathbf{x}-\mathbf{y}\| \mid \mathbf{y} \in \partial C\} \right\}, \text{ where, } \quad (4)$$

$$\mathbf{x}_i \neq \mathbf{x}_j,\ \mathbf{x}_i, \mathbf{x}_j \in \partial C,\ \mathbf{x} \in C.$$

$M(C)$ can be approximated in a fashion similar to the the area Voronoi diagram approximation by replacing ∂C with a finite number of points, constructing the Voronoi diagram of the points, and removing the superfluous edges. Ogniewicz (1993), Ogniewicz and Ilg (1992) and Ogniewicz and Kübler (1995) refer to this structure as the (discrete) Voronoi medial axis ([D]VMA). In turn, they create a (discrete) Voronoi skeleton (VSK) by attaching a measure of prominence and stability to each component of the VMA.

Area Voronoi diagrams and medial axes have been used for a variety of purposes, see (Okabe et al., 2000, pp. 188-189 and p. 185), respectively, for a review). Here we report on only those applications that have some relationship with the current one. These applications fall into two classes, document analysis and robot path planning.

Documents can be thought of as comprising of a mix of components such as text, tables, and graphics. Both Burge and Monagan (1995a), (1995b), (1995c) and Kise et al. (1998) use the approximated area Voronoi diagram of such components. The former to define neighbour relations that can be used to combine image elements into semantically meaningful objects and the latter to obtain boundaries which can be used for segmenting page images. Ilg (1990a), (1990b) and Ogniewicz and Ilg (1992) use the VSK in a procedure aimed at understanding information contained in line drawings. In particular, in the case of a street network, the VMA is used to define the virtual middle axes (spines) of the characteristic elongated enclosed regions (cheers) between two solid parallel lines used to represent the edges of surrounding block casings.

Robot path planning (also known as collision-free path planning) involves finding paths along which a robot can move without collision in a region containing a set of obstacles $A = (A_1, \ldots, A_n)$. The space in which the robot can move without collision is called the free (configuration) space. For a disk-shaped robot, radius d, in $\mathbb{R}^2$, the free space is equivalent to the complement of the area enclosed by parallel lines at distance d from A. The minimum distance between the obstacles is called the bottleneck width. This determines the critical radius (maximum value of d) for the robot. In we construct $\mathcal{V}(A)$, the edges included in the free space constitute collision-free paths. Further, the bottleneck width can be determined since the centre of the critical disk is found in the vertices of $\mathcal{V}(A)$. Examples for a disk-shaped robot in $\mathbb{R}^2$ are found in Ó'Dúnlaing and Yap (1985), Schwartz and Yap (1986), Meng (1987), Krozel and Andrisani (1990), Rao et al. (1991).

Related research includes creating polygon boundaries with Voronoi edges generated from labelled points digitally sampled from the interior of known polygon boundaries, Gold et al. (1996). Further, related developments on approximations to the shape of points include the generation of crusts (subsets of edges of the Delaunay triangulation - the dual graph of the Voronoi diagram) and skeletons from unlabelled points, Amenta et al. (1998), Gold (2000), Gold and Snoeyink (2001).

The VMA methodology suffers from the introduction of certain geometric artefacts when used in the context of automated slsn generation. These artefacts include, T-intersection and X-intersection displacements and line end noise, Flanagan et al. (1994). The approach described in this paper, directly addresses the first and last of these problems and is extensible to handle X-intersection displacements. Note that the VMA generation is substantially equivalent to area Voronoi diagram generation for slsn creation using the algorithm presented later in this paper.

In 1996 Alnoor and Martinez presented an approach to slsn creation from street blocks very similar to the present approach. In fact, the approach described in this paper was originally developed around the same time as Alnoor and Martinez in the context of creating an exhaustive polygon tiling from disjoint land-use planning policy polygons. However, the current approach provides more theoretical rigor and robustness than Alnoor and Martinez's case-based solution and also addresses the issue of intersection geometry with the addition of a Voronoi-based recursive methodology. Our implementation required only 69 lines of Arc Macro Language (AML) and embedded Perl code for the basic topologically correct slsn to be produced, with the addition of approximately 250 additional lines of code for the recursive intersection geometry adjustments.

More recently, Radke (1999) described a solution to automate slsn creation from street blocks, this time based on a Delaunay triangulation (the Voronoi diagram's dual). However, this method also resorts to a case-based approach to solving intersections and appears to be unsuccessful at automating the resolution of the "T" intersections, known as the T-displacement problem, Flanagan et al. (1994), noted above.

The method described below reasonably addresses the outstanding issues of automated slsn creation from polygonal street block casings. The next section introduces the street block data set used for testing the implementation of this methodology.

3 Study Area

The City of Rosario is located on the Rio Paraña in Santa Fe Province, north eastern Argentina. The urban fabric of this city comprises an excellent context in which to test the robustness of an approach to single line street network generation from city block casings. It contains highly regular, equal width, rectilinear streets (with right angled intersections), as well as curvilinear streets with irregularly spaced intersections. The street polygon and intersection widths range from a maximum of approximately 300 metres to a minimum of approximately 8 metres, reflecting magnitudes similar to those encountered in large metropolitan areas world-wide.

Both polygon and arc-node topology were built for the polygonal city block casings prior to the analyses discussed in the following section. The data set consists of 8,001 arcs (including 60,864 arc segments) and 7,954 nodes that comprise 7,937 street block polygons. Each city block is coded with a unique identification number.

4 Method

In this section we describe the processing steps in the area Voronoi-based centreline algorithm including T-intersection adjustment. As noted above, the algorithm follows Sugihara's approach and is summarized in pseudo-code following the discussion. A subset of the Rosario data as shown in Figure 1 is used for illustrative purposes in the sequel.

Formally, we consider the city structure to be a set of regions $A = (A_1, \ldots, A_n)$. The only other restriction on this data set is that the polygons be non-overlapping. They may be convex or non-convex. Non-convex polygons that represent blocks (or parcels) with cul-de-sacs must be split into convex partitions $A^{'}$ in order to capture their internal street segments. This step is included in the algorithm described in the sequel but is not yet implemented. However, convex partitioning is a standard computational geometry procedure, see O'Rourke (1998), so its implementation should be straightforward. Further, since some routing applications might wish to avoid dead-ends (unless these were source or destination locations) the lack of these segments might be a useful option to retain in a centreline creation algorithm. Finally, in this implementation using ArcGIS software we take advantage of stored polygon topology. Implementations in other software environments might require additional processing steps.

After the optional convex partitioning step, the vertices that define the polygon boundaries are densified to a given tolerance - $var1$ below. We used the values 5 and 10 metres on our test data set. This set of vertices $\mathcal{P}$ become the seeds for point Voronoi cell generation. Since these cells are eventually utilized to create the centreline, the level of densification determines how well the algorithm tracks curvilinear boundary features in the source polygons.

The next step is the creation of the Voronoi diagram $\mathcal{V}$. Here we use the ArcGIS THIESSEN command. In this function we can assign identification numbers to the Voronoi cells. We assign the identification numbers of the vertices' parent polygon boundary, but we have also tested assigning the parent polygon identification number as this is useful for other applications and modifications of the technique described here. Figure 2 shows the Voronoi tessellation and its seed points for a subset of the city road configuration, created from a densified set of vertices from the input city block boundaries. We now delete the edges of the cells in the Voronoi diagram $\mathcal{V}$ that share common identification numbers to create $\mathcal{V}'$. In our implementation we use the ArcGIS DISSOLVE command for this purpose.

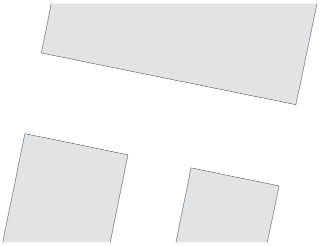

Fig. 1. A subset of input Rosario city block boundaries

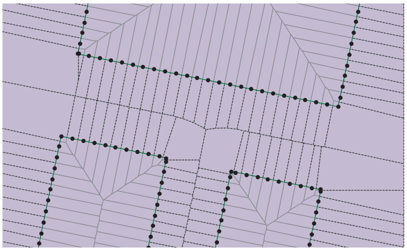

Fig. 2. Voronoi cells of a subset of input Rosario city block boundaries

The next several steps of the first part of the method are used to eliminate extraneous edges in $\mathcal{V}'$ to leave us with an approximated street centreline. We begin by creating the morphological closing $\mathcal{C}$ of A, Serra (1982), using $var2$. Here we implement the closing, following its definition, first creating the dilation of A, $Dil(A)$, by creating a positive polygon buffer of distance defined by $var2$ around the input polygon set A. Second we create the erosion of this set $Ero(Dil(A))$ by means of a negative polygon buffer of distance again defined by $var2$. We now use this structure $\mathcal{C}$ to eliminate the arcs of $\mathcal{V}'$ that fall outside the area covered by the input polygon set. Next, we delete edges in $\mathcal{V}'$ that intersect with original set A. This removes the Voronoi edges created inside the input polygons and also edges that fall within the tolerance set by $var3$. What remains of the Voronoi edges comprises our first pass street centreline.

The last step in this stage is a final clean-up for which we use a built-in ArcGIS line generalization function (that implements the Douglas-Pueker method) to simplify the edges in $\mathcal{V}'$ based on the tolerance set in $var4$ (see Figure 3). Figure 4 shows the final output centreline as the dark grey line segments for a larger subset of Rosario.

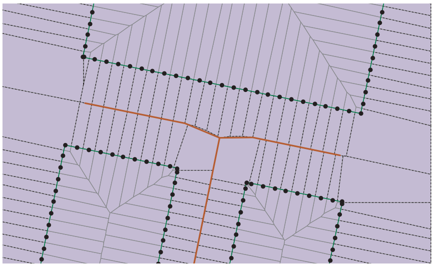

Fig. 3. SLSN from a subset of input Rosario city block boundaries

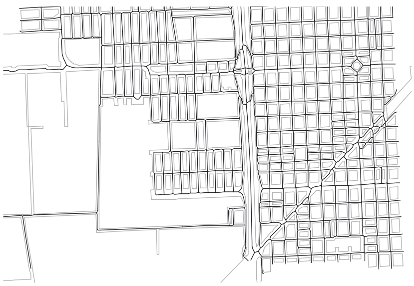

Fig. 4. Sample output centreline

Note that since partitioning the input polygons into convex polygons was not implemented in the results presented here, cul-de-sac or dead-end street segments are not generated. Our starting polygons were city blocks. Had we started with cadastral parcels as the input polygon set then most cul-de-sac street segments would be generated as most individual parcels are generally convex with respect to the street casement and it is the unique-ids of the input polygon boundaries that are used to combine (dissolve in GIS parlance) the individual Voronoi cells into the final polygon tiling. The edge set thus created is the street centreline. In a few places in the Rosario test data the street blocks were split into convex pieces and the correct cul-de-sac street segments were created.

Whilst the street centreline created is topologically correct and in fact is the true centreline of input polygon boundaries, it contains visible artefacts such as T-displacements, Flanagan et al. (1994). Thus, the next step involves schematizing the output network to introduce right angles at T-junctions to address the T-intersection displacement problem. X-intersection displacements are less of an issue but can also be addressed using an approach similar to that described below.

For the purpose of schematizing T-junctions we use the fact that a degenerate state in a Voronoi diagram of four co-circular generator points arranged in a rectangle creates a Voronoi vertex of degree four with right angles between the Voronoi edges (O'Rourke, 1998, pp. 159–160). To implement this idea we recursively generate a Voronoi tessellation with some generating points deleted or newly created. More details of this extension are provided next and Figures 5, 6 and 7 illustrate the idea.

First, clip $\mathcal{V}$ with A creating $\mathcal{V}''$ (see Figure 5). Next, create centroids of $\mathcal{V}''$ polygons. Then find polygons of $\mathcal{V}''$ with 2 sets of neighbouring polygons each with different source polygon, A'_i, call these polygons $\mathcal{TC}_i$. Call the list of polygons neighbouring $\mathcal{TC}_i$, $\mathcal{TCN}_{i,j}$ $(i \neq j)$. For each $\mathcal{TC}_i$ and for neighbouring sets $\mathcal{TCN}_{i,j}$ where $\|\mathcal{TCN}_{i,j}\| > 2$ find the minimum distance centroid from $\mathcal{TC}_i$'s centroid to the centroids of $\mathcal{TCN}_{i,j}$ or the spatial mean, $\overline{\mathbf{x}}_{\mathbf{s}}$, of the centroids of $\mathcal{TCN}_{i,j}$. Create a set $\mathcal{P}'$ consisting of all the centroids of $\mathcal{V}''$ except those centroids in the set $\{\mathcal{TCN}_{i,j}\}$ that were not the minimums found in the step above. Create the slsn as above but using $\mathcal{P}'$ in place of $\mathcal{P}$. Figure 6 shows the second generation Voronoi diagram. In this figure, in the left arm of the T-junction an existing centroid was found to be the minimum distance centroid from the set $\{\mathcal{TCN}_{i,j}, \overline{\mathbf{x}}_{\mathbf{s}}\}$ to $\mathcal{TC}_i$. In the right arm the spatial mean, $\overline{\mathbf{x}}_{\mathbf{s}}$, of $\mathcal{TCN}_{i',j'}$ was found to be the minimum distance from $\mathcal{TC}_{i'}$. Figure 7 shows the final schematized slsn result overlaid as a thick grey line on the second generation Voronoi cells. The entire algorithm is summarized below.

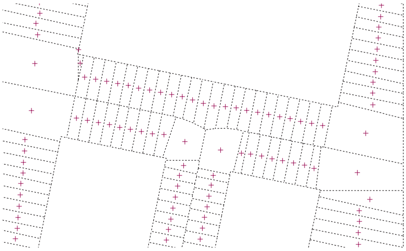

Fig. 5. Polygon data $\mathcal{V}''$ with added centroids

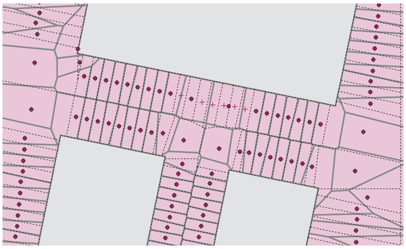

Fig. 6. Schematized T-junction Voronoi diagram

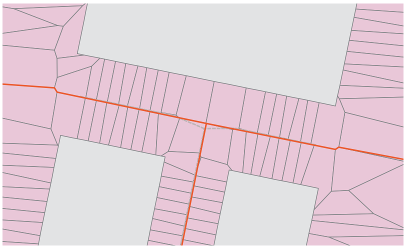

Fig. 7. Schematized SLSN from a subset of input Rosario city block boundaries

5 Centreline Algorithm

Input: Set of polygons, $A = \{A_1, \ldots, A_n\}$ $2 \leq n < \infty$, in $\mathbb{R}^2$ (see Figure 1). Assume that A_i is a connected closed set and that the areas do not overlap each other (A_i need not be convex).

Output: Approximation of centreline for set A

Variables: $var1$: vertex densification parameter (default value, 5m)
$var2$: morphological closing parameter (default value, 40m)
$var3$: selection distance tolerance (default value, 1.5m)
$var4$: line generalization parameter (default value, 1m)

Procedure:

Step 1. [Optional] Search set A for non-convex polygons. If polygon i is non-convex, partition it into j convex regions. Call this modified set of $n+m$ convex polygons (not all necessarily disjoint) A'.
Step 2. (1) *Densify* the vertices of each polygon in A' by $var1$, creating a set of points $\mathcal{P}$ (each point retains the id. number of the arc it came from).
Step 3. (2) Create a point Voronoi diagram $\mathcal{V}$ using $\mathcal{P}$ as the set of generator points (see Figure 2).
Step 4. (3) Delete edges in $\mathcal{V}$ based on shared id. numbers creating $\mathcal{V}'$.
Step 5. Create the morphological closing $\mathcal{C}$ of A using $var2$.
Step 6. *Clip* $\mathcal{V}'$ with $\mathcal{C}$.
Step 7. Delete edges in $\mathcal{V}'$ that intersect with original set A using $var3$.

Step 8. *Generalize* edges in $\mathcal{V}'$ based on $var4$.
Step 9. (4) Output $\mathcal{V}'$ (see Figure 3).

The next steps are optional and address the T-displacement problem.

Step 10. *Clip* $\mathcal{V}$ with A creating $\mathcal{V}''$ (see Figure 5).
Step 11. Create centroids of $\mathcal{V}''$ polygons.
Step 12. Find polygons of $\mathcal{V}''$ with 2 sets of neighbouring polygons each with different source polygon, A_i', call these polygons $\mathcal{TC}_i$. Call the list of polygons neighbouring $\mathcal{TC}_i$, $\mathcal{TCN}_{i,j}$ $(i \neq j)$.
Step 13. For each $\mathcal{TC}_i$ and for neighbouring sets $\mathcal{TCN}_{i,j}$ where $\|\mathcal{TCN}_{i,j}\| > 2$ find the minimum distance centroid from $\mathcal{TC}_i$'s centroid to the centroids of $\mathcal{TCN}_{i,j}$ or the spatial mean of the centroids of $\mathcal{TCN}_{i,j}$.
Step 14. Create a set $\mathcal{P}'$ consisting of all the centroids of $\mathcal{V}''$ except those centroids in the set $\{\mathcal{TCN}_{i,j}\}$ that were not the minimums found in Step 13 above.
Step 15. Repeat Steps 3–9 using $\mathcal{P}'$ in place of $\mathcal{P}$ (see Figures 6 and 7).

Notes:
i) Numbers in round brackets after step numbers refer to step numbers in Algorithm 4.13 of Okabe et al. (2000).
ii) Arc/Info GIS functions are set in italics to distinguish them from the standard computational geometry terminology.

6 Results and Conclusions

The slsn generation algorithm presented in this paper outperforms the ArcGIS CENTERLINE command in generating an accurate and topologically correct slsn, since it generates the Voronoi medial axis geometry for all street casement intersections. In fact, the ArcGIS CENTERLINE procedure failed to produce a reasonable street centreline for Rosario under any form of parameter setting. Further, the area Voronoi approach directly addresses in a novel way, based on automatically exploiting intrinsic properties of the generator set of a Voronoi diagram, the issue of schematizing the geometric representation of the slsn to provide a closer approximation to the appearance of classic manually created slsn's. This is an improvement both conceptually and practically to existing approaches by avoiding case-by-case solutions to the various intersection types that necessarily concentrate on creating new intersection geometry or modifying existing intersection geometry. Further, the method, when applied to the entire street fabric of Rosario, proved to be robust with the exception of cul-de-sacs, which can be solved using variations on the general area Voronoi approach as described earlier in the paper. It is worth noting that this approach is extensible in the sense that it can be easily applied both to very complex and large city structures as well as small and straightforward ones.

The methodology is also easily extended to induce further attributes to the slsn segments in addition to the street block identification numbers currently propagated to the generated Voronoi cells. This raises the possibility of automating attributed slsn generation to include block face-specific land-use, population or other planning and demographic information for transportation simulation and modelling analysis. In fact, if the slsn was generated from parcel polygons it should be possible to automate segmentation and attribution of the slsn for individual parcel frontages.

References

Alnoor, L. and Martinez, R. (1996). Automated derivation of high accuracy road centerlines using Thiessen polygons technique. In *ESRI User Conference Proceedings.*

Amenta, N., Bern, M., and Eppstein, D. (1998). The crust and the β-skeleton: Combinatorial curve reconstruction. *Graphical Models and Image Processing*, 60/2(2):125–135.

Burge, M. and Monagan, G. (1995a). *Extracting words and multi-part symbols in graphics rich documents*, volume 97 of *Lecture Notes in Computer Science*, pages 533–538. Springer-Verlag.

Burge, M. and Monagan, G. (1995b). Using the voronoi tessellation for grouping words and multi-part symbols in documents. In *Proceedings, Vision Geometry IV, SPIE International Symposium on Optics, Imaging and Instrumentation*, pages 116–124.

Burge, M. and Monagan, G. (1995c). Using the voronoi tessellation for grouping words and multi-part symbols in documents. Technical report, Institute of Systems Science, Johannes Kepler University.

Cao, W. and Qin, Q. (1998). A knowledge-based research for road extraction from digital satellite images. *Acta Scientiarium Naturalium Universitatis Pekinensis*, 34(2–3):254–263.

Couloiger, I. and Ranchin, T. (2000). Mapping of urban areas: a multiresolution modeling approach for semi-automatic extraction of streets. *Photogrammetric Engineering and Remote Sensing*, 66(7):867–874.

ESRI (2000). White paper on CENTERLINE command.

Eykamp, C. (1999). Supercharge AMLs with embedded Perl. *ArcUser*, pages 44–46.

Fiset, R. and Cavayas, F. (1997). Automatic comparison of a topographic map with remotely sensed images in a map updating perspective: the road network case. *International Journal of Remote Sensing*, 18(4):991–1006.

Flanagan, N., Jennings, C., and Flanagan, C. (1994). Automatic GIS data capture and conversion. In Worboys, M. F., editor, *Innovations in GIS I*, pages 25–38. Taylor & Francis.

Gold, C. (2000). Primal/dual relationships and applications. In *Proceedings of the 9th International Symposium on Spatial Data Handling*, pages 4a.15–4a.27.

Gold, C., Nantel, J., and Yang, W. (1996). Outside-in: an alternative approach to forest map digitizing. *International Journal of Geographical Information Systems*, 10(3):291–310.

Gold, C. and Snoeyink, J. (2001). A one-step crust and skeleton extraction algorithm. *Algorithmica*, 30:144–163.

Ilg, M. (1990a). Knowledge-based interpretation of road maps. In *Proceedings of the 4th International Symposium on Spatial Data Handling*, volume 1, pages 25–34.

Ilg, M. (1990b). Knowledge-based understanding of road maps and other line images. In *Proceedings, 10th International Conference on Pattern Recognition*, volume 1, pages 282–284. IEEE Computer Society Press.

K. Kise, A. S. and Iwata, M. (1998). Segmentation of page images using the area voronoi diagram. *Computer Vision and Image Understanding*, 70(3):370–382.

Krozel, J. and II, D. A. (1990). Navigation path planning for autonomous aircraft: Voronoi diagram approach. *Journal of Guidance, Control, and Dynamics*, 13(6):1152–1154.

Meng, A. C.-C. (1987). Flight path planning under uncertainty for robotic air vehicles. In *IEEE National Aerospace and Electronics Conference, Institute of Electrical and Electronics Engineers*, pages 359–366.

N. S. V. Rao, N. S. and Iyengar, S. (1991). A 'reaction' method for learned navigation in unknown terrains for a circular robot. *IEEE Transactions on Robotics and Automation*, 7(5):699–707.

Ó'Dúnlaing, C. and Yap, C. (1985). A 'reaction' method for planning the motion of a disk. *Journal of Algorithms*, 6:104–111.

Ogniewicz, R. (1993). *Discrete Voronoi Skeletons.* Konstanz: Hartung-Gorre Verlag.

Ogniewicz, R. and Ilg, M. (1992). *Voronoi skeletons: theory and applications.*

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

Okabe, A., Boots, B., Sugihara, K., and Chiu, S. (2000). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.* Wiley, second edition.

O'Rourke, J. (1998). *Computational Geometry in C.* Cambridge University Press, second edition.

Radke, J. and Flodmark, A. (1999). The use of spatial decompositions for constructing street centerlines. *Geographical Information Sciences*, 5(1):15–23.

Schwartz, J. and Yap, C. (1986). *Advances in Robotics.* Lawrence Erlbaum Associates.

Serra, J. (1982). *Image Analysis and Mathematical Morphology.* Academic Press Inc.