Segmentation of Urban Areas Using Road Networks[*]

Nicholas Jing Yuan
Microsoft Research Asia
nichy@microsoft.com

Yu Zheng
Microsoft Research Asia
yuzheng@microsoft.com

Xing Xie
Microsoft Research Asia
xing.xie@microsoft.com

ABSTRACT

Region-based analysis is fundamental and crucial in many geospatial-related applications and research themes, such as trajectory analysis, human mobility study and urban planning. In this paper, we report on an image-processing-based approach to segment urban areas into regions by road networks. Here, each segmented region is bounded by the high-level road segments, covering some neighborhoods and low-level streets. Typically, road segments are classified into different levels (e.g., highways and expressways are usually high-level roads), providing us with a more natural and semantic segmentation of urban spaces than the grid-based partition method. We show that through simple morphological operators, an urban road network can be efficiently segmented into regions. In addition, we present a case study in trajectory mining to demonstrate the usability of the proposed segmentation method.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: spatial databases and GIS, data mining.

General Terms

Algorithms

1. INTRODUCTION

In many geospatial-related applications, such as trip planning[18], urban planning/urban computing[6][25] and traffic analysis [24][15], an urban area is often segmented into sub-regions for in-depth analysis or complexity reduction. Intuitively, a digital map can be segmented into equal-sized grids, where each grid is a rectangle, as shown in Figure 2(a). While the grid-based approach offers simplicity in implementation, it has deficiencies as well: The segmented grids do not have semantic meanings, thus the results of the analysis based on these grids do not provide us with a natural understanding, with respect to the road network.

Typically, the road segments of a road network are categorizes into different levels according to their functions, e.g., Figure 1 presents the road network of Beijing with Level-0,1,2 road segments. Here, the Level-0 and Level-1 road segments (red-colored) are freeways and city expressways in Beijing, and Level-2 road segments (blue-colored) are urban arterial roads. These main roads naturally segment the urban area into sub-regions with varying sizes and shapes.

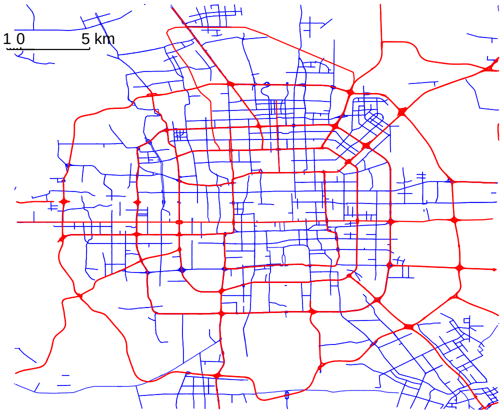


Figure 1: Level 0,1,2-road segments in the urban area of Beijing

Intrinsically, this segmentation is more natural than grid-based segmentation due to the following reasons: First, people live in these roads-segmented regions and POIs (points of interests) fall in these regions instead of the main roads. Second, the road-segmented regions as the origin and destination of a trip are the root cause of human mobility. In short, people travel among the road-segmented regions.

In a Geographical Information System (GIS), there are two major models to represent spatial data: *vector*-based model and *raster*-based model. Vector-based model uses geometric primitives such as points, lines and polygons to represent spatial objects referenced by Cartesian coordinates, while raster-based model quantizes an area into small discrete grid-cells (cuboids for the 3D spatial objects) indexing all the spatial objects. For example, the vector model of Beijing road network (shown in Figure 1) stores a *road segment* as a polyline (polygone for a circuit road), where a polyline is consist of a sequence of shape points, represented by coordinates. A road segment r is associated with a direction symbol ($r.dir$) and two terminal points ($r.s$, $r.e$). If $r.dir$=one-way, r can only be traveled from $r.s$ to $r.e$, otherwise, people can start from both terminal points, i.e., $r.s \rightarrow r.e$ or $r.e \rightarrow r.s$. Instead, the raster model rasterizes the road network into grid-cells, where each road segment is represented by a sequence of cell-IDs. Figure 3(a) and Figure 3(a) reveal a portion of the Beijing road network stored with vector model and raster model(726×726 grid-cells) respectively.

Both of the two models have advantages and disadvantages depending on the specific applications. For instance, vector-based method is more powerful for precisely finding shortest-paths, whereas it requires intensive computation when performing topological analysis, such as the problem of map simplification [10], which

[*] This technical report details the process of map segmentation utilized in our previous papers[15],[25] and [23]

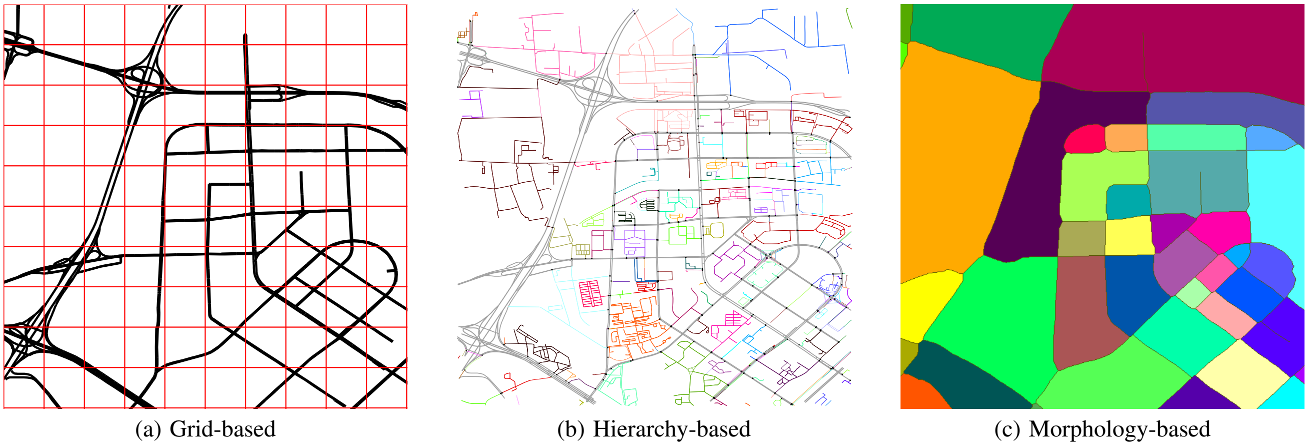

(a) Grid-based (b) Hierarchy-based (c) Morphology-based

Figure 2: Different map segmentation methods

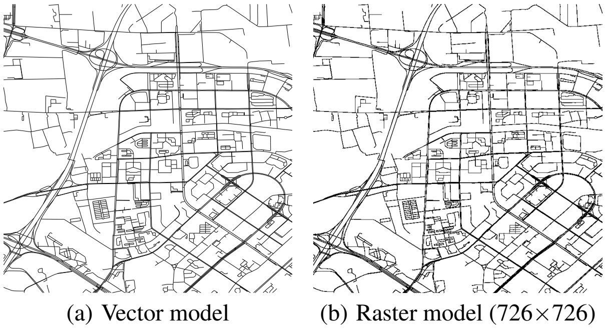

(a) Vector model (b) Raster model (726×726)

Figure 3: A portion of Beijing road network printed from vector and raster models

is proved to be NP-complete [9]. On the other hand, raster-based model is more computational efficient and succinct for territorial analysis, but the accuracy is limited by the number of cells used for discretizing the road networks.

In our method, we choose the raster-based model to represent the road network and utilize morphological image processing techniques to address the task of map segmentation. Specifically, a raster-based map is regarded as a binary image (e.g., 0 stands for road segments and 1 stands for blank space). In order to remove the unnecessary details, such as the lanes of a road and the overpasses, we first perform a *dilation* operation to thicken the roads. As a result, we can fill the small holes and smooth out the unnecessary details. Second, we obtain the skeleton of the road networks by performing a *thinning* operation. This operation recovers the size of a region which was reduced by the dilation operation, while keeping the connectivity between regions. The last step is to perform a connected component labeling (CCL) that finds individual regions by clustering "1"-labeled grids. We detail the above operations in Section 3 and provide a case study in Section 4.

2. RELATED WORK

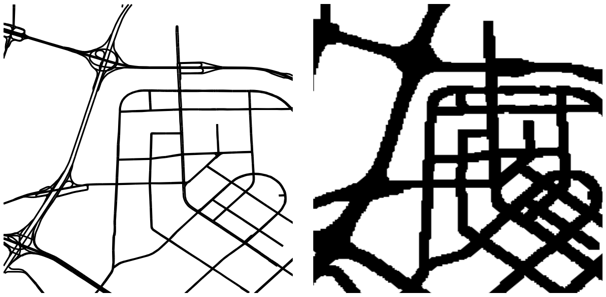
Grid-based map segmentation is extensively used in geospatial-related analysis. Krumm and Horvitz [13] predict the drivers' destination by mapping the historical trips into grid-cells and learning the destination probabilities with respect to each cell. Powell et al. [18] propose an approach to suggest the most profit grid for taxi drivers by constructing a *spatio-temporal profitability map* with a grid-based segmentation, where the probabilities are calculated by the historical data. Compared with grid-based segmentation, our solution considering the high-level roads as the boundary are more natural for studying the human mobility on a map, as described in Section 1.

Gonzalez et al. [11] propose a novel road network partition approach based on the road hierarchy. Specifically, the road networks are first divided into areas by high level roads, then the partition process is recursively performed for each area. The partition process is implemented by finding the strongly connected components after the removal of the intersection nodes connected to high level roads as well as the terminals of high level road segments themselves. Figure 2(b) presents the results of this approach for a portion of Beijing road network. However, this method does not work in our scenario since 1) our desired region bounded by high level road segments may contain several strongly connected components, and 2) we aim to segment the whole area instead of just the road nodes into regions, i.e., we need a mapping from any locations represented by latitudes and longitudes (within the bounding box of the road network) to the region IDs.

Morphology operators are widely used in Geographical information systems as well as image processing. Saradjian and Amini [20] employs mathematical morphology for map simplification from remote sensing images by extracting skeletons from the image and convert the structure into vectors. Similar works are presented in [16],[17] and [8]. Different from the above methods which are based on remote sensing images, we aim to segment the urban area represented by vector-based model into regions, instead of simplifying the map or extracting structures from the map.

3. MAP SEGMENTATION

We display the vector-based road network in a plane by performing map projection [22], which transforms the surface of a shpere (i.e., the Earth) into a 2D plane (We use Mercator projection in our approach [1, 4]). Then we convert the vector-based road network into the raster model by gridding the projected map [5]. Intuitively, each pixel of the projected map image can be regarded as a grid-cell of the raster map. Consequently, the road network is converted to a binary image, e.g., 1 stands for the road segments (termed as foreground) and 0 stands for the blank areas (termed as background). This section introduces an mathematical morphological approach



(a) Before dilation (b) After dilation

Figure 4: Dilation operator

for segmenting the binary-image-based road network into regions. In general, a morphological operator calculates the output image given the input binary image and a *structure element*, whose size and shape are pre-defined.

3.1 Dilation

Dilation is one of the basic morphological operators. Let A be a binary image and B be the structure element, the dilation of A by B is defined as:

$$A \oplus B = \bigcup_{b \in B} A_b, \tag{1}$$

where $A_b = \{a + b | a \in A\}$, i.e., the translation of A by the vector b. The dilation operator is commutative, which can also be defined by:

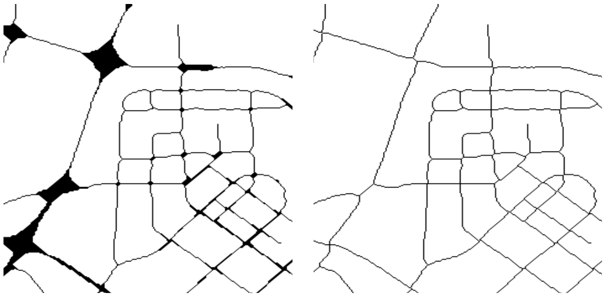
$$A \oplus B = B \oplus A = \bigcup_{a \in A} B_a. \tag{2}$$

Actually, for any p in A, after the dilation, $p = 1$ iff the intersection between A and B, centred at p, is not empty.

The purpose of the dilation operation is to remove the unnecessary details for map segmentation, avoiding the small connected areas induced by these unnecessary details such as bridges and lanes. Figure 4(a) plots a portion of road network before the dilation operator. As is shown, the small holes between the lanes and viaducts are filled, where we use a 3×3 matrix with all values one as the structure element B.

3.2 Thinning

As a consequence of the previous dilation operator, the road segments are turgidly thickened. In this step, we aim to extract the skeleton of the road segments while keeping the topology structure (such as the Euler number) of the original binary image. Specifically, the *thinning operator* is performed to remove certain foreground pixels from the input binary image. For a given pixel in the input image, whether it should be removed depends on its neighbouring pixels. Lam et al. [14] present an in-depth survey of existing thinning algorithms, most of which are implemented in a recursive manner. According to the way an algorithm check the pixels, the thinning algorithms can be categorized into parallel thinning algorithms and sequential thinning algorithms. The sequential algorithms check each pixel according to a fixed order in each iteration, and the deletion of pixel p in the nth iteration is not only related to the results in the $n - 1$th iteration, but also relate to the pixels examined before p in the nth iteration; In contrast, for the parallel algorithms, the deletion of p only depends on the results in the $n - 1$th iteration, therefore, can be performed independently in a parallel manner.



(a) After 7 iterations (b) Until convergence

Figure 5: Thinning operator

Here, we employ the subfields-based parallel thinning algorithm proposed in [12]. This method first divide the binary image space into two disjointed subfields in a checkerboard pattern. Each iteration is consist of two sub-iterations in these two subfields:

- In the first sub-iteration, check every pixel p in the first subfield, delete p iff Condition 1, 2 and 3 are all satisfied.
- In the second sub-iteration, check every pixel p in the second subfield, delte p iff Condition 1, 2 and 4 are all satisfied.

CONDITION 1. $X_H(p) = 1$, *where*

$$X_H(p) = \sum_{i=1}^{4} b_i$$

$$b_i = \begin{cases} 1 & \textit{if } x_{2i-1} = 0 \textit{ and } (x_{2i} = 1 \textit{ or } x_{2i+1} = 1) \\ 0 & \textit{otherwise} \end{cases}$$

CONDITION 2. $2 \leq \min\{n_1(p), n_2(p)\} \leq 3$, *where*

$$n_1(p) = \sum_{k=1}^{4} x_{2k-1} \vee x_{2k}$$

$$n_2(p) = \sum_{k=1}^{4} x_{2k} \vee x_{2k+1}$$

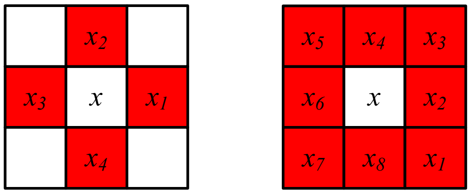
CONDITION 3. $(x_2 \vee x_3 \vee \bar{x}_8) \wedge x_1 = 0$

CONDITION 4. $(x_6 \vee x_7 \vee \bar{x}_4) \wedge x_5 = 0$

Figure 5(a) and Figure 5(b) are the results after 7 iterations and until convergence (no pixel are deleted any more) respectively.

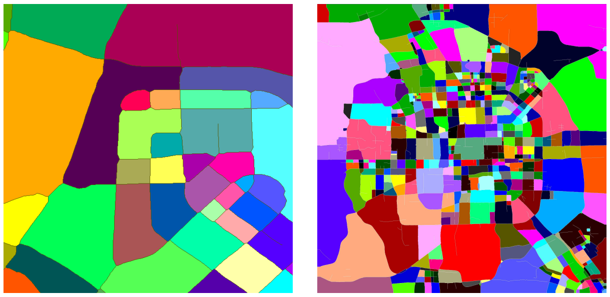
3.3 Connected Component Labeling

The *connected component labeling* operation finds the connected 0 pixels (the blank area) in the binary image, after the thinning operation. For a given pixel x, the neighbouring 4 pixels shown in Figure 6(a) are called the 4-neighbours of x. Similarly, the 8 neighbouring pixels shown in Figure 6(b) are called the 8-neighbours of x. We call the sequence $y_1, y_2, \ldots, y_n$ an 8-path (4-path), if $\forall i = 1, 2, \ldots, n-1$, y_{i+1} is an 8-neighbour (4-neighbour) of y_i. We say a region Q in a binary image is 8-connected (4-connected) iff all the pixels in Q have the same value and for any two pixels in Q, there exist an 8-path (4-path) connecting the two pixels. Note that in the previous thinning operation, connectivity paradox may be induced if we keep the same type of connectivity (4-connected or 8-connected) for both the foreground and the background[19].



(a) 4-neighbours (b) 8-neighbours

Figure 6: The 4-neighbours and 8-neighbours of x



(a) Local result (b) Global result

Figure 7: Segmented regions after connected component labeling

Since it is desired for the road segments (foreground) to have unit width, typically, we preserve the 8-connectivity of the foreground (i.e., the 8-connectivity does not change before and after the thinning process for the road segments) and the 4-connectivity of the background [14].

There exist many algorithms for connected component labeling. Here, we employ the classical two-pass algorithm [21], the main steps are as follows:

- Set the initial label l to 1 (note the label here has nothing to do with the 0-1 value in the binary image). In the first pass, scan each pixel from left to right and from top to bottom. If a pixel x is a backgroud pixel (with value 0),
  - Retrieve all the 4-neighbours of x, label x with the smallest label among its 4-neighbours; if there are no labeled 4-neighbours of x, label x with the new label $(l \leftarrow l + 1)$.
  - Store the equivalence of x's label and its 4-neighbours' labels.
- Perform the second pass: scan from left to right and from top to bottom, for each pixel x, label x with the smallest label among all the labels in the equivalent class of x's label.

Applying this algorithm to the binary image 5(b), we obtain the segmented regions as shown in Figure 7(a). Figure 7(b) presents the result for the whole road network of Beijing.

4. A CASE STUDY

After the connected component labeling, we obtain the segmented regions as well as their labels (termed as region IDs) for the whole map. Here, all the road segments form a region labeled as 0. Therefore, given a GPS point represented by latitude and longitude,

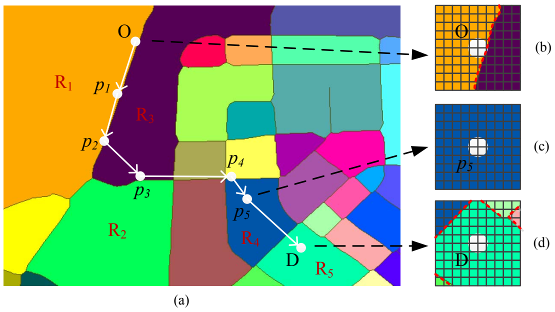


Figure 8: Trajectory mapping

we can determine which region it belongs to (by examining the region ID of this GPS point's occupied pixel). However, in many applications related to trajectories [23][15], instead of mapping isolated points, we need to transform a whole trajectory into a region ID sequence. For example, $T : S \to p_1 \to p_2 \cdots \to p_5 \to D$ in Figure 8 is a trajectory originated from region R_1 to region R_5. Note that some points may located along the roads (such as p_1) while some points actually enter the non-road regions (such as p_5). In order to obtain a meaningful mapping from the raw trajectory to the region IDs, for each point x of the trajectory, we apply the following rules:

- If x is the start/destination point, we first retrieve its k nearest neighbouring pixels $N(x)$. Then we assign x with the most non-zero (not on road segments) frequent label (region ID) in $N(x)$, e.g., the origin O is assigned in region R_1 and the destination D is labeled in region R_5, as shown in Figure 8 (b) and 8 (d) respectively.
- If x is not the start/destination point, if all its k nearest neighbouring pixels $N(x)$ have an identical label (region ID), we assign x with this label, as depicted in Figure 8 (c); otherwise, we label x with 0 (along the road).

As a result, T is mapped to the region ID sequence $R_1 \to 0 \to 0 \to 0 \to 0 \to R_4 \to R_5$.

Based on the above approach, we map a large number of GPS trajectories of taxicabs during 3 month [23] to the segmented regions, and then extract the Origin-Destination (OD) pairs. Figure 9 presents the top-200 frequent OD pairs for the time window 8am–10am on an average of weekdays, where a darker color indicates a higher frequency. Figure 10 further shows the number of transitions

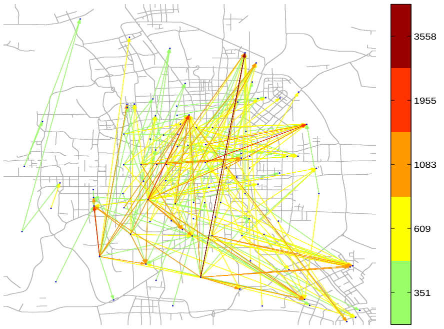


Figure 9: Top-200 frequent OD pairs on weekdays during 8am to 10am

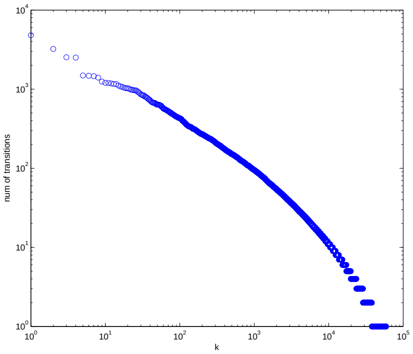


Figure 10: Number of transitions w.r.t. k

(OD pairs) changing over k (log-log plot), which suggests a typical Zipf distribution.

5. CONCLUSION

In this technical report, we present a morphological map segmentation method based on high level road segments of urban road networks. Compared with traditional segmentation approaches, such as the grid-based method and the hierarchy-based method, our method produces a more meaningful and semantic segmentation, where the high-level road segments are utilized as natural boundaries for the segmented regions.

APPENDIX

A. PSEUDO-CODE

```
Input: Vector-based road network consisting of high-level road
       segments $G$
Output: Matrix $M$;
/* the numbers in the matrix indicate the
   identities of segmented regions                */
Convert $G$ into a binary image $B$;
/* construct a raster-based model from a
   vector-based model                             */
$B \leftarrow$ Dilation($B$);
/* Perform dialation                              */
$B \leftarrow$ Thinning($B$);
/* Perform thinning until the convergence         */
$M \leftarrow$ ConnectedComponentLabeling($B$);
/* Consider 4-connectivity for the regions and
   8-connectivity for the road segments           */
return $M$
```

B. USEFUL LINKS

For the input format of the vector-based road network and the vector-to-raster conversion, please refer to [7]; For the morphological operators, such as dilation and thinning, refer to [3]; For connected component labeling, refer to [2].

References

[1] http://msdn.microsoft.com/en-us/library/aa940991.aspx.

[2] http://www.mathworks.cn/help/toolbox/images/ref/bwlabel.html.

[3] http://www.mathworks.cn/help/toolbox/images/ref/bwmorph.html.

[4] http://msdn.microsoft.com/en-us/library/bb429590.aspx.

[5] http://www.mathworks.cn/help/toolbox/map/f7-10852.html.

[6] http://research.microsoft.com/en-us/projects/urbancomputing/.

[7] http://www.mathworks.cn/help/toolbox/comm/ref/vec2mat.html.

[8] M. ANSOULT, P. SOILLE, and J. LOODTS. Mathematical morphology- a tool for automated gis data acquisition from scanned thematic maps. *Photogrammetric engineering and remote sensing*, 56 (9):1263–1271, 1990.

[9] R. Estkowski. No steiner point subdivision simplification is np-complete. In *Proc. 10th Canadian Conf. Computational Geometry*. Citeseer, 1998.

[10] R. Estkowski and J. Mitchell. Simplifying a polygonal subdivision while keeping it simple. In *Proceedings of the seventeenth annual symposium on Computational geometry*, pages 40–49. ACM, 2001.

[11] H. Gonzalez, J. Han, X. Li, M. Myslinska, and J. P. Sondag. Adaptive fastest path computation on a road network: a traffic mining approach. In *Proceedings of the 33rd international conference on Very large data bases*, VLDB '07, pages 794–805, 2007. ISBN 978-1-59593-649-3.

[12] Z. Guo and R. Hall. Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, 32(3):359–373, 1989.

[13] J. Krumm and E. Horvitz. Predestination: Where do you want to go today? *Computer*, 40(4):105–107, 2007.

[14] L. Lam, S. Lee, and C. Suen. Thinning methodologies-a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence*, 14(9):869–885, 1992.

[15] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing. Discovering spatio-temporal causal interactions in traffic data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1010–1018. ACM, 2011.

[16] E. López-Ornelas. High resolution images: segmenting, extracting information and gis integration. *World Academy of Science, Engineering and Technology*, 54:172–177, 2009.

[17] C. Martel, G. Flouzat, A. Souriau, and F. Safa. A morphological method of geometric analysis of images: Application to the gravity anomalies in the indian ocean. *Journal of Geophysical Research*, 94 (B2):1715–1726, 1989.

[18] J. Powell, Y. Huang, F. Bastani, and M. Ji. Towards reducing taxicab cruising time using spatio-temporal profitability maps. In *Proceedings of the 12th International Symposium on Advances in Spatial and Temporal Databases*, SSTD '11, 2011.

[19] A. Rosenfeld and J. Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM (JACM)*, 13(4):471–494, 1966.

[20] M. Saradjian and J. Amini. Image map simplification using mathematical morphology. *International Archives of Photogrammetry and Remote Sensing*, 33:36–43, 2000.

[21] L. Shapiro and G. Stockman. *Computer Vision. 2001*. Prentice Hall, 2001.

[22] J. Snyder. *Map projections–A working manual*. Number 1395. USGPO, 1987.

[23] J. Yuan, Y. Zheng, and X. Xie. Discovering regions of different functions in a city using human mobility and pois. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

[24] Y. Zheng and X. Zhou. *Computing with spatial trajectories*. Springer-Verlag New York Inc, 2011.

[25] Y. Zheng, Y. Liu, J. Yuan, and X. Xie. Urban computing with taxicabs. In *Proceedings of the 13th International Conference on Ubiquitous Computing Ubicomp 2011*.