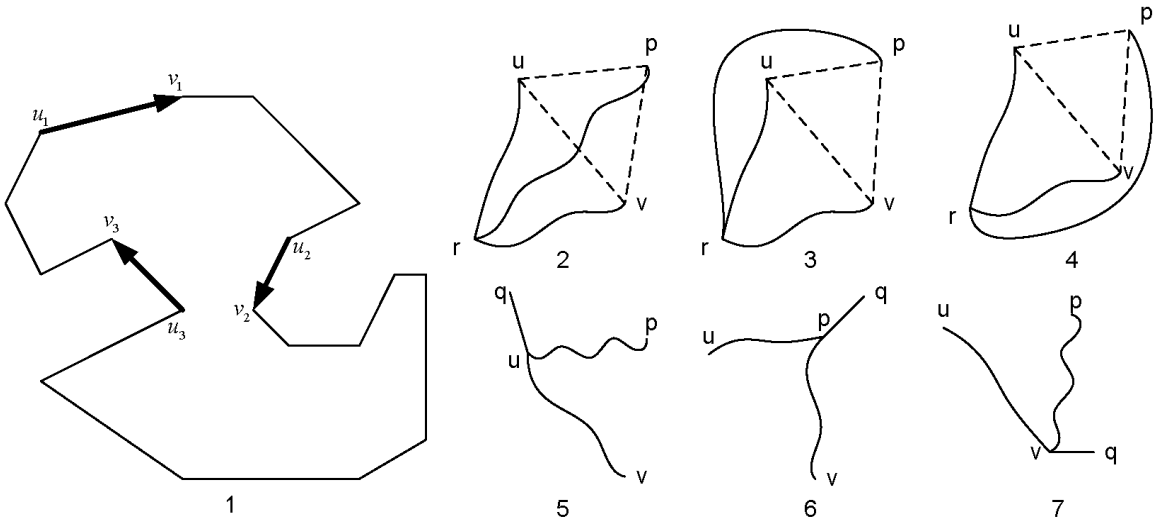
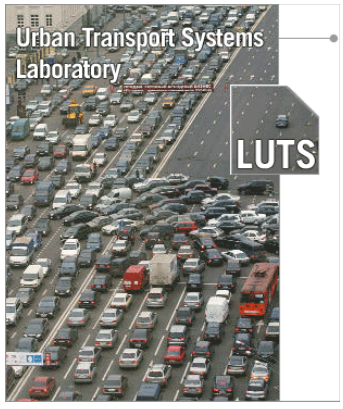


Exploring Spatial Characteristics of Urban Transportation Networks

Yuxuan Ji

Nikolas Geroliminis

STRC 2011 **May 2011**



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

STRC 2011

Exploring Spatial Characteristics of Urban Transportation Networks

Yuxuan Ji
Urban Transport Systems Laboratory
Ecole Polytechnique Fédérale de Lausanne (EPFL)
GC C2 399, Station 18, 1015 Lausanne, Switzerland
phone: +41-21-69-35397
fax: +41-21-69-35060
yuxuan.ji@epfl.ch

Nikolas Geroliminis
Urban Transport Systems Laboratory
Ecole Polytechnique Fédérale de Lausanne (EPFL)
GC C2 389, Station 18, 1015 Lausanne, Switzerland
phone: +41-21-69-32481
fax: +41-21-69-35060
nikolas.geroliminis@epfl.ch

May 2011

Abstract

It has been shown recently that a Macroscopic Fundamental Diagram (MFD) exists in urban transportation networks under certain conditions. MFD builds a relationship invariant of demands between the output and accumulation of the network. It is further utilized to efficiently facilitate the design and implementation of control strategies to optimize the performance of the networks. However, MFD is not universally expected. Previous research demonstrates the existence of MFDs in homogeneous networks with similar link densities. More recent work focuses on the partitioning of a heterogeneous transportation network based on different congestion levels. A desired partitioning produces homogeneous regions with similar link densities to guarantee a well-defined MFD and spatially compact shapes to ease the implementation of control measurements (Ji and Geroliminis, 2011). Based on recently proposed partitioning mechanism, this paper further explores the spatial characteristics of sub-networks (sub-regions or clusters) in urban transportation networks. In this paper, a metric is defined to evaluate the spatial compactness of each cluster in the network. More specifically, a boundary angle is defined for each boundary node in the sub-network to measure the smoothness of boundary and an area of incompact region along the boundary is estimated to evaluate the relative smoothness of the boundary compared to the whole region. In order to obtain the metric, a fast graph traversal algorithm is proposed, which can produce a clockwise (or counter-clockwise) sequence for the spatially coordinated boundary nodes along any network. Firstly, each sub-region of a transportation network is built as an undirected graph with each edge as a link and vertex as its endpoint. A spanning tree is then built from this undirected graph. Secondly, a linked list data structure is designed with the neighbours linked in an order based on their spatial clockwise (or counter-clockwise) sequence around each vertex. Additional leaves as representatives of the boundary nodes that are not leaves are properly added to the tree. Finally, a preorder traversal algorithm is designed based on the spanning tree to produce the desired sequence of boundary nodes along each sub-network. The algorithm takes $O(n)$ and the effectiveness is proved and

validated. By applying the boundary smoothness metric to our previous clustering results, we show that the spatial compactness is appropriately guaranteed for each region and the future control policies can therefore be easily implemented based on the partitioning and MFDs. We believe that the proposed algorithms and ideas will have broad applications in the fields of network and graph theory.

Keywords

macroscopic fundamental diagram, network partitioning, graph theory, spatial characteristics

1 Introduction

Analysis of traffic flow theory and modeling of vehicular congestion has mainly relied on fundamental laws, inspired from physics using analogies with fluid mechanics, many particles systems and the like. One main difference of physical systems and vehicular traffic is that humans make choices in terms of routes, destinations and driving behavior, which creates additional complexity to the system. While most of the traffic science theories make a clear distinction between free-flow and congested traffic states, empirical analysis of spatio-temporal congestion patterns has revealed additional complexity of traffic states and non-steady state conditions (see for example Muñoz and Daganzo (2003); Helbing *et al.* (2009)). Thus, the known fundamental diagram (initially observed for a stretch of highway and provide a steady-state relationship between speed, density and flow) is not sufficient to describe the additional complexity of traffic systems and it also contains significant experimental errors in the congested regime (see for example Kerner and Rehborn (1996) for a highway stretch or Geroliminis and Daganzo (2008) for a city street).

Nevertheless, it was recently observed from empirical data in Downtown Yokohama Geroliminis and Daganzo (2008) that by aggregating the highly scattered plots of flow vs. density from individual loop detectors, the scatter almost disappeared and a well-defined Macroscopic fundamental Diagram exists between space-mean flow and density.

The idea of an MFD with an optimum accumulation belongs to Godfrey (1969) but the verification of its existence with dynamic features is recent Geroliminis and Daganzo (2007, 2008). These papers showed, using a micro-simulation and a field experiment in downtown Yokohama, (i) that urban neighborhoods approximately exhibit a "Macroscopic Fundamental Diagram"(MFD) relating the number of vehicles to space-mean speed (or flow), (ii) there is a robust linear relation between the neighborhood's average flow and its total outflow (rate vehicles reach their destinations) and (iii) the MFD is a property of the network infrastructure and control and not of the demand, i.e. space-mean flow is maximum for the same value of vehicle density independently of time-dependent origin-destination tables. References Daganzo (2007); Geroliminis and Daganzo (2007) introduced simple control strategies to improve mobility in homogeneous city centers building on the concept of an MFD. The main logic of the strategies is that they try to decrease the inflow in regions with points in the decreased part of an MFD.

Despite these recent findings for the existence of MFDs with low scatter, these curves should not be a universal recipe. In particular, networks with an uneven and inconsistent distribution of congestion may exhibit traffic states that are well below the upper bound of an MFD and much too scattered to line along an MFD. By analyzing real data from a medium-size French city Buisson and Ladier (2009) showed that heterogeneity has a strong impact on the shape/s-

catter of an MFD. Recent findings from empirical and simulated data Geroliminis and Sun (2010); Mazloumian *et al.* (2010) have identified the spatial distribution of vehicle density in the network as one of the key components that affect the scatter of an MFD and its shape. They observed well defined relations between flow and density when link density variance is constant. In other words, the average network flow is consistently higher when link density variance is low, for the same network density.

These findings are of great importance because the concept of an MFD can be applied for heterogeneously loaded cities with multiple centers of congestion, if these cities can be partitioned in a small number of homogeneous clusters. The goal of the partitioning is to partition a network into regions with small variances of link densities and smooth and compact spatial shapes. This condition is also needed when simple perimeter control strategies are applied and each cluster is considered as a reservoir. If a cluster contains subregions with significantly different levels of congestion and weird shapes, the control strategies will be inefficient. The work presented in this paper further explores the spatial characteristics for heterogeneous transportation networks.

Based on recently proposed partitioning mechanism, we explore the spatial characteristics of sub-networks (sub-regions or clusters) in urban transportation networks. We design a metric to evaluate the spatial compactness of each cluster in the network. In the metric, a boundary angle is defined for each boundary node in the sub-network to measure the smoothness of boundary and an area of incompact region along the boundary is estimated to evaluate the relative smoothness of the boundary compared to the whole region. In order to obtain the metric, we propose a graph traversal algorithm, which can produce a sequence for the spatially coordinated boundary nodes in a clockwise order along any network. By connecting the nodes based on this sequence we can obtain a closed boundary that circles the whole network. The algorithm takes $O(n)$ and the effectiveness is also proved. We further validate the algorithm and the metric by applying them to the previous clustering results, and show that the spatial compactness is appropriately guaranteed for each sub-network. Therefore, the future control policies can be easily implemented based on the partitioning and MFDs.

The remainder of this paper is organized as follows. Section 1 designs the algorithm of boundary construction and Section 3 proves its effectiveness. Section 4 designs the shape metrics based the partitioning criteria defined in previous research work. The algorithms and metrics are implemented and validated by simulation in Section 5. Secontion 6 concludes our work.

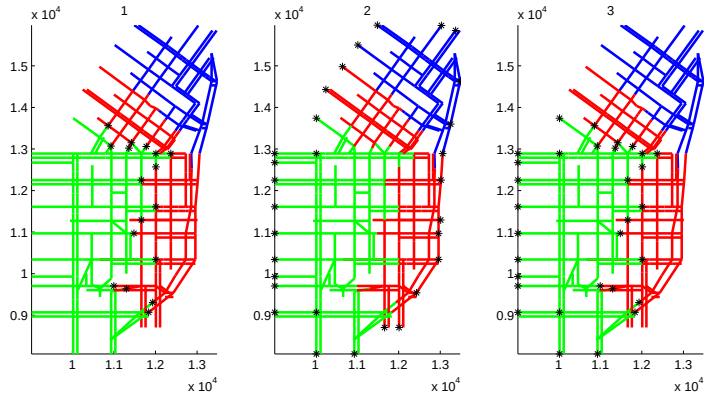


Figure 1: Boundary node detection.

2 Boundary Construction

In order to analyze the shape for a given network, we need first to construct the spatial boundary of the network. A transportation network consists of links and their endpoints. It can be equivalently considered as a graph G with each edge e as a link and node v as an endpoint. Therefore, to draw the spatial boundary for a given network, we need to first identify all the nodes on the boundary and second build a correct sequence for these boundary nodes. In a network clustered into different regions or sub-networks, there are two categories of boundary nodes for these sub-networks. In the first category, a boundary node connects two edges which belong to different clusters and so is easy to be identified. We mark this category as B_1. For example, the boundary nodes in the first category for the green cluster are shown as the black nodes in Figure 1.1. In the second category, a boundary node is on the boundary of the whole network. This category is marked as B_2. Identifying the second category of boundary nodes is also straightforward. For each node v in the graph, if there exists a quadrant (there are four quadrants based on v) which does not contain any other node or edge, we say v is on the boundary of the whole network. We also mark this quadrant as an empty quadrant of node v. For example, the boundary nodes for the San Francisco transportation network are shown in Figure 1.2. Hence the boundary nodes for the green cluster can be easily obtained as shown in Figure 1.3.

With the boundary nodes bing identified, the main task remains building a proper sequence of the boundary nodes for a given cluster (or sub-network). A proper boundary node sequence means that the boundary line based on this sequence can circle all the links and endpoints inside. Figure 2.1 shows part of a sub-network with red links and boundary nodes A, B, C, D, etc. Figure 2.2 shows a improper node sequence '$\cdots ADBC \cdots$' since it cuts some link out of the boundary. Figure 2.3 and Figure 2.4 show two proper boundary lines with node sequence '$\cdots ABCD \cdots$' and '$\cdots ACBD \cdots$', so the solution is not unique.

Therefore, we propose a fast graph traversal algorithm based on spanning tree to generate the

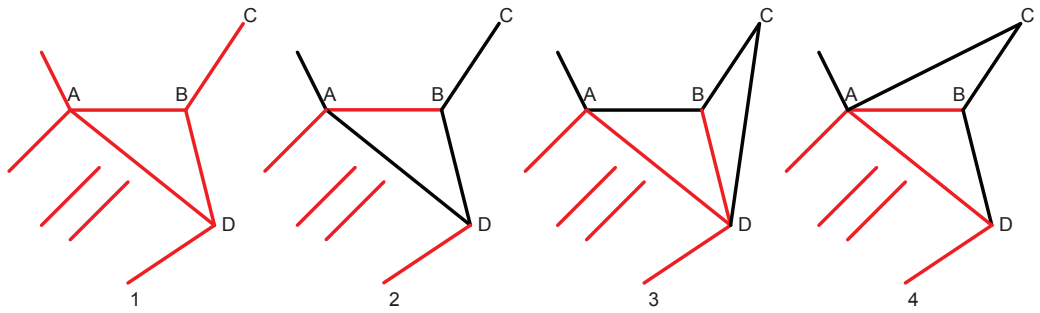


Figure 2: Examples for boundary node sequence.

proper boundary sequence for any connected plane graph built from a transportation network. We first describe the algorithm in detail and then prove its effectiveness. The algorithm is composed of three steps.

In the first step, we build the sub-network as an undirected graph with each edge e as a link and vertex v as the endpoint of a link, and generate a spanning tree from this undirected graph. Note that the generated spanning tree is not unique. The computational complexity of this step is $O(n)$ with n as the number of vertices in the graph.

In the second step, we build a special linked list data structure for the tree. Specifically for each v, all the neighbors u of v are linked in an order based on their spatially clockwise sequence around v. One approach is to sort all the degrees of the angles, each of which is formed by rotating clockwise around v from the vector $\overrightarrow{vl}$ to vector $\overrightarrow{vu}$ with l of coordinates $l_x = -\infty$ and $l_y = v_y$. Note that the linked list of v can start with any of its neighbor as long as the order is clockwise. Next, for each of the identified boundary nodes that are not leaf nodes in the tree of the sub-network, we add a representative node for it into the tree. Suppose v is a boundary node in the sub-network but not a leaf in the spanning tree. We add a leaf u on the node of v and specify the coordinates of u so that vector $\overrightarrow{vu}$ points towards the outside of the sub-network. This can be done in the following way. If v belongs to the first category of the boundary nodes, we can assign u with the coordinates of p where edge $\overline{vp}$ belongs to a neighboring cluster and insert u into the linked list (note that u should be placed in the correct position based on its angle with v calculated in the same way as above). If v belongs to the second category, we can simply assign u an arbitrary pair of coordinates that lie in the empty quadrant of v. The complexity for building a linked list is $O(|n| + 2|e|)$ and $O(|u|log|u|)$ for sorting where $|u|_{\max} \ll |n|$. This step is formalized in Algorithm 1.

In the last step, we preorder traverse the tree. We can start from any node in the tree. Suppose the first visited node is v. Then we visit the first child (or neighbor) of v, say u, in its linked list (note that we can also start with any other child of v). Then we search v in the linked list of u and the node right after v, say p, will be the next one to be visited, if p has not been visited yet. Note that the linked list is visited as a cyclic list, which means that if v is the last node in

Algorithm 1 ADD-ADDITIONAL-LEAVES(T, B)

Input: A tree T of sorted (ascending) linked list structure, set of boundary nodes B.
Output: A new tree T.

{Add a new leaf as a representative for each of the boundary nodes that are not leaves}

for every v in T **do**
  if v is not a leaf **and** $v \in B$ **then**
    generate a new leaf node u
    add v to the linked list of u
    specify a reference l with $l_x = -\infty, l_y = v_y$
    if $v \in B_1$ **then**
      find an edge $\overline{vp}$ in the neighbor cluster
      $u_x \leftarrow p_x, u_y \leftarrow p_y$
      $u_a \leftarrow$ clockwise angle from $\overrightarrow{vl}$ to $\overrightarrow{vu}$
    else
      $u_a \leftarrow$ clockwise angle from $\overrightarrow{vl}$ to the empty quadrant of v
    end if
    insert u into the sorted linked list of v based on u_a
  end if
end for
return T

the linked list of u, then we locate the first node in the list as the next node. And all the other nodes are visited in the same way as p. In short, the preorder traversal algorithm is the same as the common procedure except that when we first visit u from a call of the father node v, the first node to be visited is determined by the position of v in the linked list of u, instead of starting from the first one in the list. This traversal algorithm runs recursively and will produce a proper sequence for the boundary nodes based on the order of the visited leaves. We mark the set of leaves as L. If a leaf is a representative, then the original node will replace this leaf in the final boundary node sequence. We mark the set of the nodes as representatives by R. The complexity of this step is also $O(n)$, so the whole algorithm only takes $O(|n| + 2|e|)$. This traversal algorithm is formalized in Algorithm 2 and the final boundary node sequence is produced by Algorithm 3.

3 Effectiveness of the Algorithm

In this section we analyze the effectiveness of the proposed algorithm.

Proposition (Boundary Construction) *Given a plane and connected spatial graph G and the set of its boundary nodes B, the proposed algorithm will produce a cyclic sequence S for all the nodes in B. By connecting each pair of neighbor nodes in S, a closed boundary can be obtained which circles inside all the edges and vertices in G.*

Algorithm 2 TRAVERSE-TREE(f, v, T)

Input: A tree T of sorted (ascending) linked list structure, father node f, current node v.
Output: Leaf sequence S.

{Traverse the tree and produce a leaf sequence. $visited = FALSE$ is global.
In the first entry of the recursion, $f = 0$ and v can be any node in the tree.}

$visited[v] \leftarrow TRUE$
if $v \in L$ **then**
  add v to the end of S
end if
if $f = 0$ **then**
  for every u in the linked list of v **do**
    TRAVERSE-TREE(v, u, T)
  end for
else
  find the next node p after f in the cyclic sorted linked list of v
  while $visited[p] = FALSE$ **do**
    TRAVERSE-TREE(v, p, T)
    find the next node q after p in the cyclic sorted linked list of v
    $p \leftarrow q$
  end while
end if

Algorithm 3 PRODUCE-BOUNDARY-SEQUENCE(G, B)

Input: Original graph G built from the sub-network and the set of boundary nodes B.
Output: Boundary node sequence F.

create a spanning tree T with a cyclic sorted (ascending) linked list from G
$T \leftarrow$ ADD-ADDITIONAL-LEAVES(T, B)
$S \leftarrow$ TRAVERSE-TREE($0, 1, T$)
for every $v \in S$ from the first to the end **do**
  if $v \in R$ **then**
    $u \leftarrow$ original node v represents
  else
    $u \leftarrow v$
  end if
  if $u \in B$ **then**
    place u at the end of the sequence F
  end if
end for
return F

Here we prove the proposition. The proof is based on the existence of a desired sequence.

Proof *For a connected plane graph G, a proper cyclic sequence for boundary nodes is denoted as S_{prop}, along which the boundary nodes will form a clockwise circle around the graph without leaving out any edge or vertex outside. This implies that for any pair of neighbor nodes $\overrightarrow{uv}$ in S_{prop} (i.e., with v right after u), the area on the immediate right side of vector $\overrightarrow{uv}$ resides in*

the graph G, while that on the left side of the vector is out of G. Figure 3.1 draws a graph with an arbitrary shape and illustrates the above implication by several pairs of neighboring boundary nodes in different positions, as shown by $\overrightarrow{u_1v_1}$, $\overrightarrow{u_2v_2}$, $\overrightarrow{u_3v_3}$, etc. Assume that the proposed algorithm produces an improper sequence S_{improp} for boundary nodes. This means that there exists some pair of neighboring boundary nodes $\overrightarrow{uv}$ with their immediate left side area belonging to the graph region. Therefore, for a proper sequence S_{prop}, some boundary nodes should lie between u and v (e.g., '$\cdots u \cdots p \cdots v \cdots$') so that S_{prop} can include the missed region by S_{improp}, as shown in the dotted triangle ΔUPV in Figure 3.2. In this case, the generated spanning tree based on G can only be of the structure shown as the real curve lines in Figure 3.2 (the curve line implies there may be some other nodes on it, and r is the most closest common root of u, p and v). The tree is neither possible to be Figure 3.3 in which u would no longer be a boundary node, nor Figure 3.4 in which v would not be a boundary node.

Based on the structure in Figure 3.2, suppose that u is just visited and so the algorithm goes back to r. Then the next branch to be visited would be the middle one where p resides on, based on the clockwise linked list of r. After all the nodes on the middle branch are visited, the algorithm goes to the branch where v resides on. So based on the algorithm, p will definitely be visited before v, which conflicts with the initial assumption that the algorithm gives v right after u. In the generated spanning tree, r may coincide with any of u, p and v as shown in Figure 3.5, 3.6 and 3.7. However in any of the three cases, p will still be visited before v. For example, if the tree is produced as in Figure 3.7, then a leaf q will be added as a representative of v, in any direction opposite to the graph region. Still suppose the currently just visited node is u, then the algorithm goes back to v, and will next visit the branch of p, and finally visit the branch of q, based on the clockwise linked list of v. Note that although v is visited before p, the position of q which is the representative of v will be the final position for v in the boundary node sequence. The other two cases can be analyzed similarly. Therefore, based on the proof by contradiction, we conclude that the algorithm can not produce any improper boundary sequence. Hence the effectiveness of the algorithm is proved.

The proposed algorithm will be implemented and further validated by the simulation of a real urban transportation network.

4 Metric Development

In this section, we design a shape metric for a network (i.e., any of the clustered region or sub-network) to evaluate the smoothness of its boundary. The objective of this metric is twofold. Firstly, the non-smoothness or weird shape of a boundary along a region should be captured.

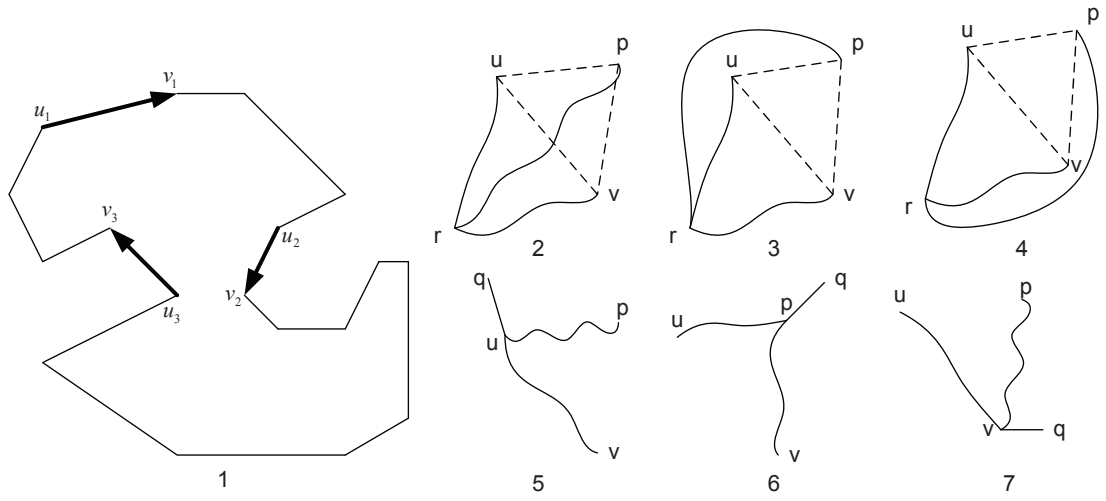


Figure 3: Proof of the algorithm.

Since the boundary is built by appropriately connecting a series of nodes (i.e., the end points of the links) along the region, we can achieve this goal by checking every boundary angle formed by a boundary node with its two neighbor nodes. If this angle is too small or sharp, it implies a bad shape or non-smoothness part of the boundary. For instance in Figure 4.1, if we define all the angles smaller than $\pi/3$ as bad angles, then $\angle AIH$ will be a bad angle and the area of ΔAIH is identified as a bad shape. If the threshold for bad angle is defined by $\pi/2$, then both $\angle AIH$ and $\angle DEF$ are considered too sharp and the two corresponding triangles ΔAIH and ΔDEF can be used to represent the bad area. Secondly, the smoothness of the boundary should be evaluated together with the whole region. This means that for two regions with the same weird boundary shapes but different network size, they should be measured with different degrees of smoothness. For instance, Figures 4.2 and 4.3 have the same bad angles and areas as shown above the dotted lines, if the threshold for bad angle is $\pi/2$. However, since the whole region inside the boundary in Figure 4.3 is much bigger than Figure 4.2, we say the boundary along Figure 4.3 is more smooth than the one of Figure 4.2. In order to achieve this goal, we compute the ratio of the bad areas over the whole region to obtain a relative value for smoothness of the boundary. Based on these two objectives, the metrics are designed in detail as follows.

Firstly we build a clockwise (or counter-clockwise) sequence for the boundary nodes along each region to draw the shape of the region. Secondly, we design a boundary angle measure for each boundary node in a certain region for the degree of smoothness around this node. Specifically, the boundary angle for node i is defined as:

$$BoundaryAngle(i) = \overrightarrow{i(i+1)} \overset{clockwise}{\rightarrow} \overrightarrow{i(i-1)}, \tag{1}$$

where $\overrightarrow{i(i+1)}$ denotes the vector from node i to node $i+1$ and so the boundary angle at node i

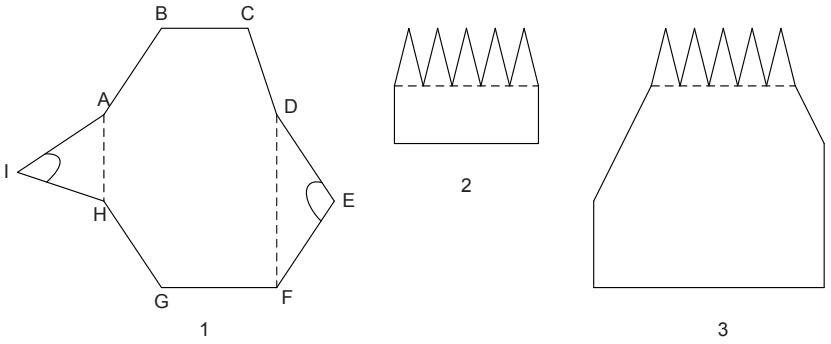


Figure 4: Boundary examples.

is measured by clockwise rotating vector $\overrightarrow{i(i+1)}$ to vector $\overrightarrow{i(i-1)}$ around node i. In addition, if two boundary angles α and β satisfy $\alpha + \beta = 2\pi$, we say they have the same degree of smoothness given that $||\overrightarrow{i(i+1)}|| \approx ||\overrightarrow{i(i-1)}||, \forall i \in S_B$ where S_B denotes the set of nodes on boundaries. Hence we can equivalently have all the boundary angles less than π and evaluate the smoothness of a boundary node by setting a threshold value for smoothness such as $\pi/2$. For instance, if a boundary angle $\alpha > \pi/2$, we say it is smooth. Otherwise we tag it as a non-smooth node. Finally, for each of the non-smooth boundary node i, we calculate the area of a triangle formed by nodes i, $i-1$ and $i+1$. Then the non-smoothness of the a region R can be roughly estimated with the dimensionless quantity:

$$NonSmoothness(R) = \sum\nolimits_i A(i)/A(R), \tag{2}$$

where $A(i)$ is the area of a non-smooth boundary node as before and $A(R)$ is the area of the whole region. Therefore, based on this definition, the smoothness of the boundary along a region is appropriately measured as a relative value, which implies that for a large region the existence of only a few non-smooth boundary nodes will not seriously affect the whole smoothness measure while for a small region it can be the opposite, as expected in the objectives. With the above metric we can now evaluate the spatial smoothness of all the regions in a partitioning.

5 Implementation and Simulation

In this section, the proposed algorithms and metrics are implemented and simulated in a real urban transportation network. The boundary construction algorithm is tested for different clusters in the network as shown in Figure 5. Figure 5.1-5.4 show the algorithm works well, while Figure 5.5 shows some part of the yellow region is not included in the boundary. This is because the vertices that are not circled by the boundary are not boundary nodes. It implies that there does not exist a closed boundary surrounding the whole network, given an incomplete set of boundary nodes. However, it is obvious to see the algorithm can still produce a proper

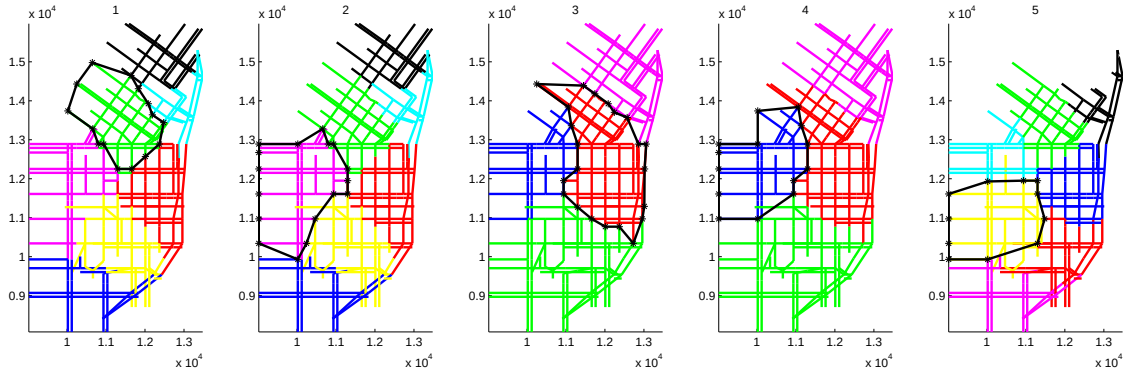


Figure 5: Boundary construction for different clusters.

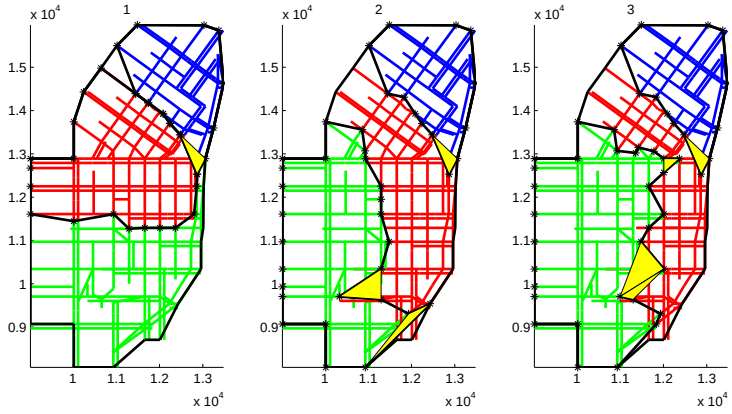


Figure 6: Spatial compactness measure of partitioning at $t = 70$

boundary for the network, by taking these missed boundary nodes as internal nodes for the network.

Next we analyze the spatial metric designed above to show the smoothness of some cluster boundaries in the network. We test this metric for three kinds of partitioning in our previous paper in which we mainly compared their variance metrics. The analysis on the variance metrics has showed significant improvement of the partitioning quality (i.e., more intra-cluster similarity and inter-cluster dissimilarity) from the initial segmentation to the final results based on the previously proposed partitioning mechanism. However our objectives of partitioning also include spatial requirement of compactness to facilitate future design of control strategies. In the first two steps of partitioning the spatial compactness is guaranteed by setting a distance threshold which strictly keeps the spatial connectivity and in the final step the adjustment is made based on the TV metric under some constraint of spatial compactness (e.g., the lower bound on the number of simultaneously adjusted links). Thus we now evaluate the spatial compactness metric and show the effectiveness of our final boundary adjustment algorithm which can further decrease the variance while at the same time maintain the spatial smoothness properly.

We evaluate the shape metric for three partitioning results given by initial Ncut in Figure 6.1,

Table 1: Spatial compactness evaluation at $t = 70$

Partitioning	Ncut	Merging	Bo. Adj.
Shape metric (Non-smoothness)	0.62%	2.73%	3.31%

after merging in Figure 6.2 and boundary adjustment in Figure 6.3 (all with 3 clusters). The yellow areas are the non-smooth regions in each partitioning detected by the spatial compactness metric. The spatial non-smoothness metric is shown in Table 1. Note that the smoothness along the external network boundary is not included. It is clear that the spatial compactness is properly maintained through the presented mechanism.

6 Conclusions and Future Work

Traffic congestion is increasing in urban cities. In this paper, in order to study the existence of MFD and traffic control from a macroscopic level, the spatial characteristics of the urban transportation networks are further explored based on our previously designed partitioning mechanism and the obtained results. Specifically, boundary construction algorithm is designed for a spatial network that can be built as a plane and connected graph. The algorithm takes $O(n)$ and its effectiveness is proved in detail. Furthermore, a shape metric for the network based on the criteria of the previous partitioning mechanism is designed. Finally the algorithms and the metrics are implemented and simulated in a real urban transportation network. The algorithm is further validated and the spatial metric applied to our previous clustering results show that the spatial compactness is appropriately guaranteed for each sub-network and the future control policies can therefore be easily implemented based on the partitioning. In the future work, we will continue to study the traffic propagation by exploring the spatial and temporal features of congestion, and design control strategies for the heterogeneous network with different levels of congestion.

References

Buisson, C. and C. Ladier (2009) Exploring the impact of homogeneity of traffic measurements on the existence of macroscopic fundamental diagrams, *Transportation Research Record*, **2124**, 127–136.

Daganzo, C. F. (2007) Urban gridlock: Macroscopic modeling and mitigation approaches, *Transportation Research part B*, **41** (1) 49–62.

Geroliminis, N. and C. F. Daganzo (2007) Macroscopic modeling of traffic in cities, paper

presented at *86th Annual Meeting of the Transportation Research Board*, Washington, DC. Paper No. 07-0413.

Geroliminis, N. and C. F. Daganzo (2008) Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings, *Transportation Research Part B-Methodological*, **42** (9) 759–770.

Geroliminis, N. and J. Sun (2010) Properties of a well-defined macroscopic fundamental diagram, paper presented at *Transportation Research Board 89th Annual Meeting*, Washington, DC. Paper No. 10-3521.

Godfrey, J. W. (1969) The mechanism of a road network, *Traffic Engineering and Control*, **11** (7) 323–327.

Helbing, D., M. Treiber, A. Kesting and M. Schönhof (2009) Theoretical vs. empirical classification and prediction of congested traffic states, *European Physical Journal B*, **69** (4) 583–598.

Kerner, B. S. and H. Rehborn (1996) Experimental properties of complexity in traffic flow, *Physical Review E*, **53** (5) 4275–R4278.

Mazloumian, A., N. Geroliminis and D. Helbing (2010) The spatial variability of vehicle densities as determinant of urban network capacity, *Philosophical Transactions of Royal Society A*. in press.

Muñoz, J. C. and C. F. Daganzo (2003) Structure of the transition zone behind freeway queues, *Transportation Science*, **37** (3) 312–329.