Route-Aware Edge Bundling for Visualizing Origin-Destination Trails in Urban Traffic

Wei Zeng¹, Qiaomu Shen², Yuzhe Jiang², Alex Telea³

1. Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
2. The Hong Kong University of Science and Technology
3. University of Groningen
Introduction
- OD Trails in Urban Traffic
- Prior Edge Bundling Methods
- Limitations of KDEEB

Route-Aware Edge Bundling
- Preprocessing:
  - map matching $\rightarrow$ hierarchical route structure construction $\rightarrow$ trail abstraction
- Bundling
  - optimal kernel size setting $\rightarrow$ density map generation
- Evaluation
  - Bundle termination
  - Bundle deviation

Conclusion and Future Work
Introduction
- OD Trails in Urban Traffic
- Prior Edge Bundling Methods
- Limitations of KDEEB

Route-Aware Edge Bundling
- Preprocessing:
  - map matching → hierarchical route structure construction → trail abstraction
- Bundling
  - optimal kernel size setting → density map generation
- Evaluation
  - Bundle termination
  - Bundle deviation

Conclusion and Future Work
Urban traffic data, e.g.,
- Taxi trips in New York, Beijing, Shenzhen
- Public transportation data in Singapore
- Electric scooter tracks in Stuttgart

Origin-destination (OD) is a fundamental concept in transportation, summarizing (people/vehicle/good) movements across geographic locations.

Properties of OD trails in urban traffic
- Locations
- Times
- Road network
- Multi-modes
OD Trail Visualization

- **Density Map**
  - Summarize trajectories and overview distribution.

- **Spatial Aggregation**
  - Partition underlying territory into appropriate areas.

- **Map Matching**
  - Align position records with road network data.

- **Direct depiction**
  - Directly plot trajectories into 2D/3D displays.

References:

- [Kwan, 2000]
- [Kapler and Wright, 2004]
- [Scheepens et al., 2011]
- [Wood et al., 2010]
- [Andrienko and Andrienko, 2011]
- [Krüger et al. 2018]
Prior Edge Bundling Methods

- **Geometry-based** methods: Use control mesh to specify how similar edges are routed.
  - Pros: Flexible to make control mesh
  - Cons: Constructing control mesh can be (very) slow

- **Force-based** methods: Model interaction between spatially close trails as a force field.
  - Pros: No need to make external control mesh
  - Cons: Slow – cannot handle a few thousands trails at interactive rates

- **Image-based** methods: Employ image-processing methods to accelerate the bundling process.
  - Pros: Feasible for GPU implementation – can process millions of trials at interactive rates.
  - Cons: No consideration of spatial constraints when applied to OD trails.
**Prior Edge Bundling Methods**

- **Constrained Bundling**: Specific constraints are considered.
  - Ambiguity
  - 3D curved surfaces
  - Directions
  - Obstacles avoidance
  - **Vector map**

*Vector map for Swiss commuter data [Thöny & Pajarola, 2015]*

*Map matching*  
*Vector map*
We chose KDEEB for the basis of our method:
  - Fast in speed, meanwhile simple enough to implement
  - Be able to incorporate specific constraints

KDEEB pipeline
  - Sampling
  - Gradient estimation
  - Advection
  - Smoothing

Iterate $n$ times until stable layout
  - Predefined 10 or 15 times
  - Automatically determined at runtime?
Limitations of KDEEB: What is a suitable $pr$?

- KDEEB: 5% of graph drawing size
  - $5\% \times \sqrt{1440^2 + 720^2} = 80.5$

Graphs with different values of $pr$: $P_r = 120$, $P_r = 80$, $P_r = 40$, $P_r = 20$. 
Limitations of KDEEB: Road neglect
Limitations of KDEEB

Map Matching

KDEEB ($pr = 60$)

KDEEB ($pr = 21$)

Artifacts
Introduction
- OD Trails in Urban Traffic
- Prior Edge Bundling Methods
- Limitations of KDEEB

Route-Aware Edge Bundling
- Preprocessing:
  - map matching → hierarchical route structure construction → trail abstraction
- Bundling
  - optimal kernel size setting → density map generation
- Evaluation
  - Bundle termination
  - Bundle deviation

Conclusion and Future Work
Route-Aware Edge Bundling

- RAEB pipeline: 1) Preprocessing, 2) Bundling, and 3) Evaluation

Diagram showing the process:
- Raw Road Network → Simplified Road Network → Hierarchical Route Structure → Preprocessing
- Raw Urban Traffic → OD Trails on Road Network → Abstract OD Trails → Bundling Evaluation
- Density Map → Gradient Map → Sampled Edges → Bundled Graph
- Frechet Distance → Sampled Edges → Smooth Bundles
- Previous Image → Current Image → Final Image

Input:
- Road network
- OD trails

Output:
- Level of details
Build a simplified hierarchical road and traffic network representation.

- Map matching: shortest path for OD only, ST-matching for GPS traces
- Hierarchical structure construction: route length, road hierarchy, flow magnitude
- Trail abstraction: route awareness ($p_{ra}$)
- KDEEB applied to the hierarchical structure.
  - Optimal kernel size setting
  - Density map generation

\[
\rho_{raeb}(x \in \mathbb{R}^2) = \sum_{y \in D} K \left( \frac{||x - y||}{p_r} \right) + \theta \sum_{r \in R_{aware}} \Theta(||x - r||).
\]

**Algorithm 1 KernelSizeSetting**

**Input:** Top \( N \) routes \( P = \{P_1, ..., P_N\} \)

**Output:** Initial kernel size \( p_r \)

1: for \( i = 1 \) to \( N \) do
2:     for \( j = i + 1 \) to \( N \) do
3:         \( d[i][j] = d[j][i] = \text{DiscreteFrechetDistance}(P_i, P_j) \)
4:     end for
5: end for
6: \( C = \text{DBSCAN}(P, \varepsilon, \text{minNum}) \);
7: \( C_{max} = \text{argmax}_{C_i \in C} |C_i| \);
8: \( d_{geo} = 0 \);
9: for each \( P_i \in C_{max} \) do
10:     for each \( P_j \in C_{max} \) \&\& \( i \neq j \) do
11:         \( d_{geo} = d_{geo} + d[i][j] \);
12:     end for
13: end for
14: \( p_r = d_{geo} / |C_{max}| / (|C_{max}| - 1) / 2 \);
15: return \( p_r \)
**Evaluation**

- **Termination**: Bundle stability ($\rho_s$) to determine when to stop iteration
- **Bundling**:
  
  \[
  NMI(X,Y) = \frac{2MI(X,Y)}{H(X) + H(Y)}
  \]
  
  \[
  MI(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left( \frac{p(x,y)}{p(x)p(y)} \right)
  \]
Contents

- Introduction
  - OD Trails in Urban Traffic
  - Prior Edge Bundling Methods
  - Limitations of KDEEB

- Route-Aware Edge Bundling
  - Preprocessing:
    - map matching → hierarchical route structure construction → trail abstraction
  - Bundling
    - optimal kernel size setting → density map generation
  - Evaluation
    - Bundle termination
    - Bundle deviation

- Conclusion and Future Work
Application 1: Synthetic Data
Application 1: Synthetic Data

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Edge samples</th>
<th>$p_n$</th>
<th>Time (sec.)</th>
<th>Deviation (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>KDEEB</td>
<td>RAEB</td>
</tr>
<tr>
<td>Synthetic</td>
<td>4.6M</td>
<td>13</td>
<td>40.3</td>
<td>50.7</td>
</tr>
<tr>
<td>New York</td>
<td>3.1M</td>
<td>11</td>
<td>34.3</td>
<td>42.9</td>
</tr>
<tr>
<td>Shenzhen</td>
<td>1.3M</td>
<td>8</td>
<td>13.8</td>
<td>22.8</td>
</tr>
</tbody>
</table>
Application 2: NYC Taxi

(d) KDEEB \( (p_r = 21) \)

(a) Map Matching
(b) KDEEB \( (p_r = 60) \)
(c) KDEEB \( (p_r = 40) \)

(e) RAEB \( (p_r = 21, p_{ra} = 0) \)
(f) RAEB \( (p_r = 21, p_{ra} = 1) \)
(g) RAEB \( (p_r = 21, p_{ra} = 3) \)
(h) RAEB \( (p_r = 21, p_{ra} = 5) \)
Application 2: NYC Taxi

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Edge samples</th>
<th>$p_n$</th>
<th>Time (sec.)</th>
<th>Deviation (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>KDEEB</td>
<td>RAEB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KDEEB</td>
</tr>
<tr>
<td>Synthetic</td>
<td>4.6M</td>
<td>13</td>
<td>40.3</td>
<td>50.7</td>
</tr>
<tr>
<td>New York</td>
<td>3.1M</td>
<td>11</td>
<td>34.3</td>
<td>42.9</td>
</tr>
<tr>
<td>Shenzhen</td>
<td>1.3M</td>
<td>8</td>
<td>13.8</td>
<td>22.8</td>
</tr>
</tbody>
</table>
Application 3: Shenzhen Taxi

(a) Map Matching
(b) KDEEB
(c) RAEB

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Edge samples</th>
<th>$p_n$</th>
<th>Time (sec.)</th>
<th>Deviation (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>KDEEB</td>
<td>RAEB</td>
</tr>
<tr>
<td>Synthetic</td>
<td>4.6M</td>
<td>13</td>
<td>40.3</td>
<td>50.7</td>
</tr>
<tr>
<td>New York</td>
<td>3.1M</td>
<td>11</td>
<td>34.3</td>
<td>42.9</td>
</tr>
<tr>
<td>Shenzhen</td>
<td>1.3M</td>
<td>8</td>
<td>13.8</td>
<td>22.8</td>
</tr>
</tbody>
</table>
Discussions

- RAEB constrains trails to a given road network
  - Route awareness ($p_{ra}$): controls how bundles follow roads at a user-selected hierarchy level.
  - Kernel size ($p_r$): determined by both the road network geometry and its resolution in image space.
  - Bundling stability ($p_s$): automatically stops bundling when this similarity exceeds a given threshold.

- RAEB outperforms KDEEB on both synthetic and real OD trails
  - Visually more realistic
  - Quantitively closer to ground-truth results
  - Comparable running time

- Limitations and future work
  - Visual hints on bundle deformation
  - Incorporate directional bundling techniques
  - Local and adaptive parameter settings: $p_{ra}$ and $p_r$
Thank You!

Dr. Zeng Wei
Associate Researcher
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
E-mail: wei.zeng@siat.ac.cn
Web: zeng-wei.com