Roadscape-based Route Recommender System using Coarse-to-fine Route Search

Koji Kawamata
Ryukoku University
Otsu, Shiga, Japan
t18m057@mail.ryukoku.ac.jp

Kenta Oku
Ryukoku University
Otsu, Shiga, Japan
okukenta@rins.ryukoku.ac.jp

ABSTRACT

We propose the Roadscape-based Route Recommender System (R3), which provides diversified roadscape-based routes. Given starting and destination points, R3 provides four types of roadscape-based routes: rural-, mountainous-, waterside-, and urban-prior routes. To reduce the computational cost, we propose a coarse-to-fine route search approach that consists of a roadscape-based clustering method, a roadscape cluster graph, a coarse-grained route search, and a fine-grained route search. We evaluated the performance of R3 using network data for a real road. The experimental results show that using coarse-grained route search can significantly reduce route search time.

CCS CONCEPTS

• **Information systems → Social recommendation**;

KEYWORDS

route recommender system, route search, roadscape

1 INTRODUCTION

Cars are driven not only for transportation but also for the pleasure of it. Some people want to drive along the seaside or on rural roads while enjoying their favorite landscape. We call such roadside landscapes "roadscapes." In such situations, it is not always the best solution to provide the shortest or the fastest route. An alternative solution is to provide routes with favored roadscapes even if they involve a detour.

Given starting and destination points, a route recommender system provides routes from the starting point to the destination point. The majority of traditional route recommender systems provide the shortest routes [3, 7], the fastest routes [4, 5, 9, 11, 12], or popular routes [1, 6, 8, 10]. As mentioned above, the shortest and the fastest routes do not always satisfy the user's demands. Systems that recommend popular routes provide routes many people are interested in. Wei et al. [8] extract popular routes by mining road links many people are interested in from their trajectories. Such route recommender systems consider the attractiveness of routes based on the wisdom of crowds, without considering the content features of routes.

In this paper, we focus on the roadscape as a route feature and propose the Roadscape-based Route Recommender System (R3), which provides diversified routes on the basis of roadscapes. Given starting and destination points, R3 provides four types of roadscape-based routes: rural-, mountainous-, waterside-, and urban-prior routes. For example, a user who likes waterside views can select waterside-prior routes from the four types of routes provided. To develop such a route recommender system, we have proposed a method for estimating roadscapes of given road links. In particular, we defined rural, mountainous, waterside, and urban elements as the roadscape elements, which are basic elements that compose a roadscape, through preliminary experiments. We defined a roadscape vector each of whose elements corresponds to a roadscape element and proposed a method for estimating such roadscape vectors for given road links. We presuppose that R3 is to be used on road network data with roadscape vectors.

Traditional route searching algorithms, such as the Dijkstra algorithm [2], are given the costs of road links and find a route that minimizes the sum of their costs. The simplest approach is to apply the traditional method and reduce the costs of the road links having the targeted roadscape elements. However, there is a high computational cost in applying such a method to a very large road network.

To reduce the computational cost, we propose a coarse-to-fine route search approach. We focus on the concept that similar roadscapes do not exist as fragments but in clusters. For example, there are some areas composed of similar roadscape elements, such as rural areas, mountainous areas, waterside areas, and urban areas. Based on this characteristic, we expect that we can reduce the computational cost by clustering similar roadscape areas in advance.

In this approach, we firstly extract areas—roadscape clusters—composed of similar roadscape elements by using a roadscape-based clustering method. Secondly, we create a roadscape cluster graph whose nodes correspond to the roadscape clusters and whose links correspond to the links between roadscape clusters. In the route searching process, given the roadscape cluster graph and starting and destination points, we roughly find four types of roadscape-based routes, which are the roadscape cluster sets passed through, one for each roadscape element; we call this the coarse-grained route search. Then, we find specific routes that connect the roadscape clusters in each type of route; we call this the fine-grained route search.

The contributions of this paper are as follows:

- We propose the Roadscape-based Route Recommender System (R3), which provides diversified roadscape-based routes, namely, rural-, mountainous-, waterside-, and urban-prior routes.
- To reduce the computational cost, we propose a coarse-to-fine route search approach that consists of a roadscape-based clustering method, a roadscape cluster graph, a coarse-grained route search, and a fine-grained route search.
- We evaluate the performance of R3 using network data for a real road. The results show that using coarse-grained route search can significantly reduce route search time.

2 PRELIMINARIES

Definition 1: Road network. A road network is a directed weighted graph $G = (V, E)$, where V is a set of road nodes and $E \subseteq V \times V$ is a set of road links. A road node $v_i \in V$ represents an intersection or an endpoint of a road. A road link $e_k = (v_i, v_j) \in E$ is a directed link from the starting node v_i to the ending node v_j. A road link e_k is assigned a cost w_k according to the length of the link.

Definition 2: Roadscape element. Roadscape elements are basic elements that compose a roadscape. We define four roadscape elements: rural, mountainous, waterside, and urban elements. These elements were selected by preliminary experimentation.[1]

Definition 3: Roadscape vector. A roadscape vector is defined as a four-dimensional probability vector each of whose elements corresponds to one of the respective roadscape elements. We define a roadscape vector of a road link e_i as $\boldsymbol{s}(e_i) = (s_i^r, s_i^m, s_i^w, s_i^u)$. Each element of the vector denotes the probability of how strongly e_i includes the corresponding roadscape element. Therefore, the sum of the values over all elements is 1.

Definition 4: Roadscape cluster. A roadscape cluster $C_j \in C$ is represented by a set of road links having similar roadscape vectors. A roadscape vector $\boldsymbol{s}(C_j)$ of roadscape cluster C_j is represented by the mean vector of the roadscape vectors of the road links included in cluster C_j. Therefore, we define $\boldsymbol{s}(C_j)$ as follows:

$$\boldsymbol{s}(C_j) = \frac{1}{|C_j|} \sum_{i \in C_j} \boldsymbol{s}(e_i). \tag{1}$$

Here, $|C_j|$ denotes the number of road links included in the roadscape cluster C_j.

Definition 5: Roadscape cluster graph. A roadscape cluster graph is a directed weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of roadscape clusters C_i and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of links between roadscape clusters. A link $l_k = (C_i, C_j) \in \mathcal{E}$ is a directed link from the starting node C_i to the ending node C_j. The road link l_k is assigned a cost vector $\boldsymbol{\omega_k} = (\omega_k^r, \omega_k^m, \omega_k^w, \omega_k^u)$ based on the roadscape vector $\boldsymbol{C_j}$ of ending roadscape cluster C_j. Each element of $\boldsymbol{\omega_k}$ denotes a cost for the corresponding roadscape; these are used for roadscape-based route searching. For example, ω_k^r is the cost referenced when searching for rural-prior routes.

Definition 6: Intra-cluster similarity of roadscape vector. The intra-cluster similarity is the mean similarity between all pairs of road links included in the cluster. We denote the intra-cluster similarity of roadscape cluster C_j as intra_sim(C_j). The value of intra_sim(C_j) is calculated as follows:

$$\text{intra\_sim}(C_j) = \frac{1}{n|C_j|} \sum_{i \in C_j} \sum_{k \in C_j} \cos(\boldsymbol{s}(e_i), \boldsymbol{s}(e_k)). \tag{2}$$

Here, e_i and e_k are road links included in cluster C_j, and n denotes the total number of links in the road network. The value of $\cos(\boldsymbol{s}(e_i), \boldsymbol{s}(e_k))$ is calculated as follows:

$$\cos(\boldsymbol{s}(e_i), \boldsymbol{s}(e_k)) = \frac{\boldsymbol{s}(e_i) \cdot \boldsymbol{s}(e_k)}{|\boldsymbol{s}(e_i)||\boldsymbol{s}(e_k)|}. \tag{3}$$

[1]The preliminary experimentation to select the roadscape elements was done via crowdsourcing. These four elements are specific to Japanese road network data. Details are outside the scope of this paper.

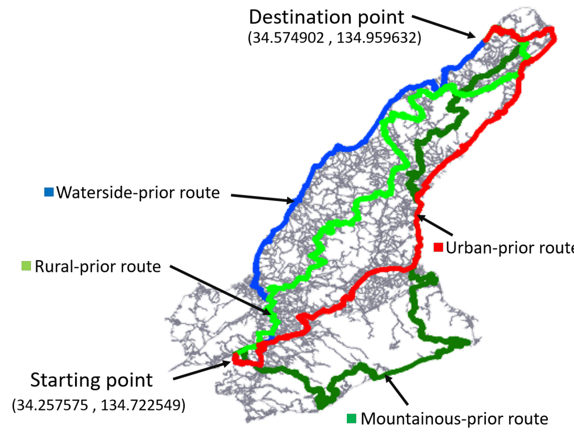

Figure 1: Recommended roadscape-based routes.

3 ROADSCAPE-BASED ROUTE RECOMMENDER SYSTEM

3.1 System Overview

Our proposed Roadscape-based Route Recommender System (R3) provides four types of roadscape-based routes: rural-, mountainous-, waterside-, and urban-prior routes. Figure 1 shows a result provided by R3. When a user inputs starting and destination points on the map, the four types of roadscape-based routes are provided in different colors.

It is assumed that R3 will be used with a road network with roadscape vectors. The steps of R3 are as follows:

(1) Generate roadscape cluster graph based on road network with roadscape vectors.
(2) Roughly find four types of roadscape-based routes in the roadscape cluster graph based on the starting and destination points that are input (coarse-grained route search).
(3) Find a detailed route that connects roadscape clusters in each type (fine-grained route search).
(4) Recommend four types of routes in different colors on the map.

Here, step (1) can be performed offline because this process does not depend on the inputs. In the next sections, we describe steps (1)–(3) in detail.

3.2 Generating Roadscape Cluster Graph

3.2.1 Roadscape-based Clustering. Given a road network, we form roadscape clusters based on proximities of pairs of road links and similarities between their roadscape vectors. Adjacent road links belong to the same cluster if their similarity is greater than or equal to a given threshold value. Figure 2 shows the result of applying roadscape-based clustering to the road network of Awaji

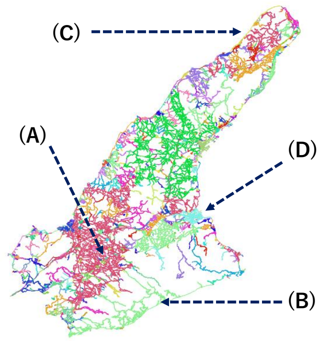

Figure 2: Result of applying roadscape-based clustering to the road network of Awaji Island, Japan. Each color corresponds to a given cluster.

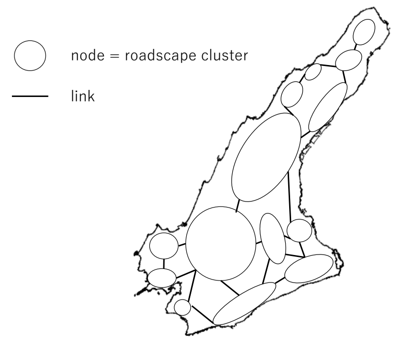

Figure 3: Example of a roadscape cluster graph created for Awaji Island's road network.

Island, Japan. Here, area A corresponds to a rural area, area B corresponds to a mountainous area, area C corresponds to a waterside area, and area D corresponds to an urban area.

Algorithm 1 shows the pseudocode for roadscape-based clustering. We explain the clustering process as performed by Algorithm 1 as follows:

Algorithm 1 Roadscape-based clustering.

Require: Target link e_i, Cluster ID k

function ROADSCAPECLUSTERING(e_i, k)
  Cluster ID of $e_i \Leftarrow k$
  linkList $\Leftarrow$ getLink(e_i): Get links adjacent to e_i.
  for each e_j in linkList
    if Cluster ID of $e_j = 0$ **then**
      if $\cos(s(e_i), s(e_j)) >= \alpha$ **then**
        roadscapeClustering(e_j, k)
      end if
    end if
  end for
  return 0
end function

We randomly select a road link from the road network. Let e_i be the target link, and let e_j be one of the links adjacent to e_i. Here, if two links are connected to a common node, the links are considered adjacent. Furthermore, let $\boldsymbol{s}(e_i)$ and $\boldsymbol{s}(e_j)$ be roadscape vectors of the respective links.

The roadscape-based clustering algorithm is called as `roadscapeClustering`(e_i, k). First, add k as the cluster ID of e_i. Second, get all links adjacent to e_i, and set them into `linkList`. For each link $e_j \in$ `linkList`, perform the following process. If a cluster ID has not been assigned to e_j, $\cos(\boldsymbol{s}(e_i), \boldsymbol{s}(e_j))$ (Equation (3)) is calculated. If $\cos(\boldsymbol{s}(e_i), \boldsymbol{s}(e_j))$ is greater than or equal to the threshold α, cluster ID k of e_i is added as the cluster ID of e_j. Furthermore, `roadscapeClustering`(e_j, k) is recursively called. The above process is repeated until the cluster ID has been added to all of the links in the road network.

We define the roadscape cluster obtained by the above process as $C_k \in C$, where k corresponds to the cluster ID. In addition, roadscape vector $\boldsymbol{s}(C_k)$ of cluster C_k is calculated by Equation (1).

3.2.2 Generating Roadscape Cluster Graph. After extracting the roadscape clusters, we create the adjacency matrix for all roadscape clusters. The adjacency matrix for the roadscape clusters is represented as the $|C| \times |C|$ matrix $\mathcal{A} = [a_{ij}]_{|C| \times |C|}$. If $a_{ij} = 1$, clusters C_i and C_j have at least one common node; otherwise, they do not have a common node.

We then create the roadscape cluster graph based on the adjacency matrix. Figure 3 gives an example of the roadscape cluster graph created for Awaji Island's road network. Here, a node in the roadscape cluster graph corresponds to a roadscape cluster, and a link corresponds to the adjacency relationship between clusters.

3.2.3 Assigning Costs to Roadscape Cluster Graph. In order to execute the coarse-grained route search described in the next section, we assign costs to the links of the roadscape cluster graph in advance. A link cost is calculated based on the roadscape vector of the roadscape cluster corresponding to the link's destination. If the targeted roadscape element of the next roadscape cluster destination is emphasized, let its link cost be lower; on the other hand, if it is not emphasized, let its link cost be higher. For example, for a case in which a rural element is targeted, if the rural element of the next roadscape cluster destination is emphasized, let its link cost be lower; otherwise, let its link cost be higher. By assigning costs in such a way, the route to the roadscape cluster where the rural element is emphasized is more likely to be chosen in the route search.

A cost vector $\boldsymbol{\omega}_k$ of link $l_k = (C_i, C_j)$ is calculated as follows:

$$\boldsymbol{\omega}_k = d_k(1 - \boldsymbol{s}(C_j)^2). \tag{4}$$

Here, d_k is the length of link l_k.

3.3 Coarse-grained Route Search

As the first search, we execute the coarse-grained route search method. This method roughly finds four types of roadscape-based routes in the roadscape cluster graph. The process is as follows:

(1) Given starting and destination points, get roadscape clusters and starting and destination clusters, which include the starting and destination points, respectively.

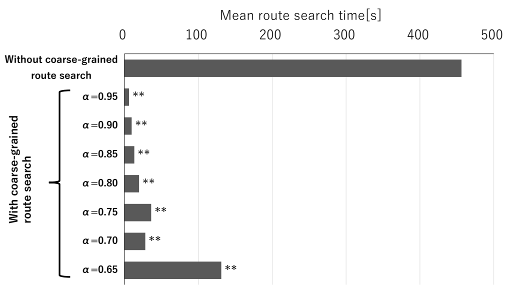

Figure 4: Comparison of route search times.

(2) For the targeted roadscape element, find a route that minimizes the sum of the link costs related to the targeted elements using Dijkstra's algorithm [2] on the roadscape cluster graph.
(3) Repeat step (2) for each roadscape element.

Thus, we obtain four types of coarse-grained routes as the roadscape cluster sets that are passed through for each roadscape element.

3.4 Fine-grained Route Search

As the second search, we execute the fine-grained route search method for each coarse-grained route. This method finds detailed routes that connect roadscape clusters. The process for each targeted element is as follows:

(1) Find common road nodes of each adjacent cluster in the roadscape cluster sets captured by the coarse-grained route search.
(2) Find the shortest route from the starting point to the first common road node that is adjacent to the next cluster.
(3) While there are common road nodes, find the shortest route from the common road node to the next common node.
(4) Find the shortest route from the last common node to the destination point.
(5) Generate a route that connects all the routes obtained.

Here, we again use Dijkstra's algorithm [2] to find the shortest routes. Finally, we obtain four types of fine-grained routes.

4 RESULTS

In this section, we evaluate the performance of the proposed R3 method using network data for a real road in Awaji Island, Japan. The road network data are derived from OpenStreetMap,[2] and they include 102,506 road nodes and 212,050 road links for the area of Awaji Island. For this area, roadscape vectors for all road links are available on the web.[3]

R3 introduces a coarse-grained route search as preprocessing to reduce the route search time instead of performing a route search on all road links. In this section, we compare the route search times using coarse-grained route search with those not using it.

First, we prepare the following five pairs of starting and destination points.

(a) $(34.257575, 134.722549) \rightarrow (34.574902, 134.959632)$
(b) $(34.317774, 134.676412) \rightarrow (34.348304, 134.896255)$
(c) $(34.499798, 134.938260) \rightarrow (34.293801, 134.788816)$
(d) $(34.545838, 134.923368) \rightarrow (34.440009, 134.912038)$
(e) $(34.208185, 134.814500) \rightarrow (34.430861, 134.830634)$

For each pair, we execute the route search algorithm that emphasizes each roadscape element and measure the route search time. We regard this execution as one trial. We execute this trial ten times for each pair and calculate the mean of the route search times across trials.

We implemented the route search algorithm using Java and managed the road network data using PostgreSQL 9.5. We conducted experiments on a computer equipped with an Intel Core i5-6200U CPU (2.8 GHz), 8 GB memory, 256 GB SSD, and Linux Mint 18.2.

Figure 4 shows the mean route search times for methods with and without coarse-grained route search. For the method with coarse-grained route search, the figure includes the route search time for each value of α. $**$ indicates that a significant difference ($p < 0.01$) could be confirmed when comparing with the method without coarse-grained route search by the paired t-test (one-sided test). We can see from Figure 4 that the route search time can be shortened by using coarse-grained route search. The figure also shows that the higher the value of α was, the shorter the route search time was. In particular, when $\alpha = 0.95$, the search time with coarse-grained route search was 6.24 s, whereas it was 456 s when coarse-grained route search was not used. Consequently, we can say that the use of coarse-grained route search can significantly reduce route search time.

5 CONCLUSIONS

In this paper, we have proposed a Roadscape-based Route Recommender System (R3) that provides diversified roadscape-based routes. Given starting and destination points, R3 provides four types of roadscape-based routes: rural-, mountainous-, waterside-, and urban-prior routes. To reduce computational costs, we proposed a coarse-to-fine route search approach that consists of a roadscape-based clustering method, a roadscape cluster graph, a coarse-grained route search, and a fine-grained route search.

We evaluated the performance of R3 using real road network data with roadscape vectors in the area of Awaji Island. The results show that using coarse-grained route search can significantly reduce route search time. In the future, we will conduct user tests to evaluate our system from the users' perspective.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers JP15K12151 and JP16HO593.

REFERENCES

[1] Zaiben Chen, Heng Tao Shen, and Xiaofang Zhou. 2011. Discovering Popular Routes from Trajectories. In *Proceedings of the 2011 IEEE 27th International Conference on Data*, Vol. 4. 900–911.
[2] Edsger Wybe Dijkstra. 1959. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* 1 (1959), 269–271.
[3] Andrew V Goldberg and Chris Harrelson. 2005. Computing the shortest path: A search meets graph theory. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. 156–165.

[2]https://www.openstreetmap.org/
[3]https://zenodo.org/record/1405255#.W4Yyb-j7T-g

[4] Hector Gonzalez, Jiawei Han, and Xiaolei Li. 2007. Adaptive fastest path computation on a road network: a traffic mining approach. In *Proceedings of the 33rd international conference on Very large data bases*. 794–805.

[5] Evangelos Kanoulas, Yang Du, Tian Xia, and Donghui Zhang. 2006. Finding fastest paths on a road network with speed patterns. In *Proceedings of the 22nd International Conference on Data Engineering*, Vol. 2006. 10.

[6] Wuman Luo, Haoyu Tan, Lei Chen, and Lionel M. Ni. 2013. Finding time period-based most frequent path in big trajectory data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 713–724.

[7] Michalis Potamias, Francesco Bonchi, Carlos Castillo, and Aristides Gionis. 2009. Fast shortest path distance estimation in large networks. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 867–876.

[8] Ling-yin Wei, Wen-chih Peng, Bo-chong Chen, and Ting-wei Lin. 2010. Eleventh International Conference on Mobile Data Management PATS : A Framework of Pattern-Aware Trajectory Search. In *Mobile Data Management*. 362–377.

[9] Ling-yin Wei, Wen-chih Peng, Chun-shuo Lin, and Chen-hen Jung. 2009. Exploring Spatio-Temporal Features. In *Advances in Spatial and Temporal Databases, Lecture Notes in Computer Science*. 399–404.

[10] Ling-Yin Wei, Yu Zheng, and Wen-Chih Peng. 2012. Constructing popular routes from uncertain trajectories. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 195–203.

[11] Jing Yuan, Yu Zheng, Chengyang Zhang, and Wenlei Xie. 2010. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 99–108.

[12] Jing Yuan, Yu Zheng, Liuhang Zhang, XIng Xie, and Guangzhong Sun. 2011. Where to find my next passenger. In *Proceedings of the 13th international conference on Ubiquitous computing*. 109–118.

