Lets Not Stare at Smartphones while Walking: Memorable Route Recommendation by Detecting Effective Landmarks

Shoko Wakamiya
NAIST
wakamiya@is.naist.jp

Hiroshi Kawasaki
Kagoshima University
kawasaki@ibe.kagoshima-u.ac.jp

Yukiko Kawai
Kyoto Sangyo University
kawai@cc.kyoto-su.ac.jp

Adam Jatowt
Kyoto University
adam@dl.kuis.kyoto-u.ac.jp

Eiji Aramaki
NAIST
aramaki@is.naist.jp

Toyokazu Akiyama
Kyoto Sangyo University
akiyama@cc.kyoto-su.ac.jp

ABSTRACT

Navigation in unfamiliar cities often requires frequent map checking, which is troublesome for wayfinders. We propose a novel approach for improving real-world navigation by generating short, memorable and intuitive routes. To do so we detect useful landmarks for effective route navigation. This is done by exploiting not only geographic data but also crowd footprints in Social Network Services (SNS) and Location Based Social Networks (LBSN). Specifically, we detect point, area, and line landmarks by using three indicators to measure landmark's utility: *visit popularity*, *direct visibility*, and *indirect visibility*. We then construct an effective route graph based on the extracted landmarks, which facilitates optimal path search. In the experiments, we show that landmark-based routes outperform the ones created by baseline from the perspectives of the lap time and the number of references necessary to check self-positions for adjusting route directions.

Author Keywords

Location-based social networks; route search; visibility

ACM Classification Keywords

H.2.8 Database Applications: Spatial databases and GIS; D.4.7 Organization and Design: Interactive systems

INTRODUCTION

Various route navigation systems have been developed so far. Majority of them provide effective routes in terms of distance or time required to reach destinations. However, the route instructions given by automatic systems and those by humans tend to be different: e.g., the former usually output instructions with street names used as reorientations, while the latter often use landmarks [18]. Due to this mismatch, when following route directions suggested by automatic systems users have to check the directions multiple times, often, by eyeballing maps on screens of their handheld devices. This not only makes the orienteering and way-finding difficult but it also leads to concerns of safety while walking or cycling [19]. Furthermore, the premise of conventional navigation systems such as a voice navigation is that they can identify users' positions based on GPS. Such systems cannot be used in case when GPS is not available or when wayfinders are not able to use any electronic devices during moving. Note that scenarios when users cannot use smart devices nor GPS happen quite commonly, e.g., when moving by a bicycle, traveling abroad. They may also occur during system failures due to power outage, earthquake, etc. *As a solution, a small number of navigation instructions is preferred, as being easy to be memorized and requiring low number of map references.*

Using landmarks for route navigation is quite effective [17, 3, 6]. Landmarks increase users' spatial awareness by informing about surroundings and by decreasing effort required for constructing mental representations of unfamiliar cities. *It is thus more natural, intuitive and efficient for users to memorize route directions in terms of landmarks.*

Different from traditional route suggestion methods, we propose in this paper an approach for recommending efficient, reliable and memorable routes. The objective is to automatically construct simple and intuitive navigation routes for input pairs of origin and destination locations. We assume that an effective route is composed of useful landmarks and we propose discovering them based on two aspects. The first is *landmark visibility* at each intersection, while the second is *social recognition* of landmarks. To estimate the level of landmark visibility, first, we construct a 3D representation of a terrain. We then measure the visibilities of all buildings at all intersections assuming the point of view of an average height person (about 60 inches or 153 cm). With the visibility measured in this way, we are able to select highly visible buildings to be considered as reliable landmarks based on the building's and surrounding terrain's shapes and arrangements.

On the other hand, to estimate the landmark's social recognition we propose to analyze microblogs and Location-Based Social Networks (LBSN) such as Twitter and Foursquare. In particular, we make the following assumptions:

1. *Places visited by many people are on average easier to be noticed than places visited by only few.*

UbiComp '16, September 12-16, 2016, Heidelberg, Germany.

ACM XXX-X-XXXX-XXXX-X/XX/XX...$15.00.
http://dx.doi.org/XX.XXXX/XXXXXXX.XXXXXXX

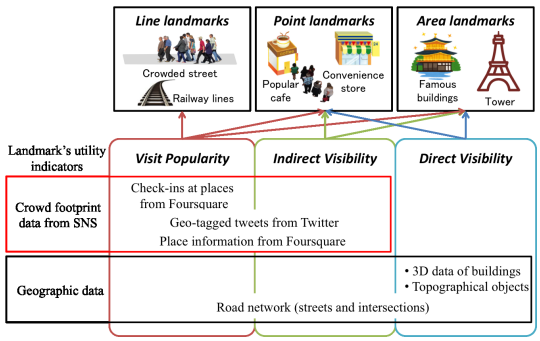


Figure 1: Overview of extracted landmarks.

2. *Popular landmarks are frequently mentioned by many people, including persons at distant locations.*

According to the first assumption, social recognition can be determined by analyzing location stamps of tweets and Foursquare checkins. The reason for proposing the second assumption is the higher probability of landmark recognition. For popular landmarks various signposts or signboards are often provided including navigational directions or related advertisements. Moreover, such locations tend to be known well by city inhabitants, so users may be able to receive additional directions in case they get lost. In consequence, even if such landmarks may not be directly seen due to terrain or other obstacles, users may still be able to locate them.

Previous research [3, 6] mainly focused on two types of landmarks: local and global landmarks. *Local landmarks* are places easily recognized by users who are in their vicinity (*e.g.*, gas station, post offices). On the other hand, *global landmarks* are taller structures that can be noticed from far away, e.g., high-rise buildings or towers. Note that due to their typically larger sizes, global landmarks only indicate approximate directions for navigating users.

However, the remaining problem is that it is difficult to use these two types of landmarks together due to their different characteristics such as visibility and position identification. Consider the following recommendation: "Walk toward *the red tower* and turn right at *the gas station*." Since there can be several possible routes leading to the tower (a global landmark), a navigating person cannot be always expected to reach the destination, i.e., the gas station (a local landmark). The problem results from the hidden assumption that the user is going to take the same route when following the global landmark as the one implied by a route giver. In reality, however, such an assumption cannot be assured. As a solution, we introduce a new category of landmarks, *line landmarks*. Their purpose is to smoothly "connect" local and global landmarks for efficient navigation. Using this new type of landmarks, one could instruct the user as follows: "Walk toward *the red tower* freely until you face *a tram-line*. Then go straight along *the tram-line* and turn right at *the gas station*." Note that since the tram-line (a line landmark) has continuous geometry, the navigating person has high chance to reach it, irrespective of the actual route taken (assuming, of course, his/her moving direction is more or less correct).

We propose then detecting three kinds of effective landmarks in this work as follows (see Fig. 1 for overview): **Point landmark** is similar to local landmarks discussed above. Although, it is characterized by rather poor visibility, its relatively high visit popularity within local areas should facilitate easy discovery (*e.g.*, post office, restaurant). Navigating users are expected to identify their locations accurately when received a route recommendation with point landmarks (*e.g.*, "Go straight and turn right at *the post office*.") **Line landmark** is a continuous line connecting multiple crowded intersections such as popular streets, rivers, and railways. In this work, for simplicity, we model high traffic streets as line landmarks. Note that since line landmarks are characterized by 1D ambiguity along a route, they were not commonly used as landmarks in prior research. **Area landmark** is similar to global landmarks being a tall and distinctive structure that can be recognized from far away (high direct visibility). In addition, we regard an area landmark to be gathering considerable crowd's attention even from users who are at different locations and who cannot directly see it (high indirect visibility). Area landmarks also enable users to freely choose their routes until the next decision point. By using landmarks and additional navigation points, we construct a simple route graph that facilitates an optimal route search in terms of the number of landmarks as well as distance. Given a start S and a destination D, we search for a user-friendly route from within the set of possible routes between the locations on the route graph.

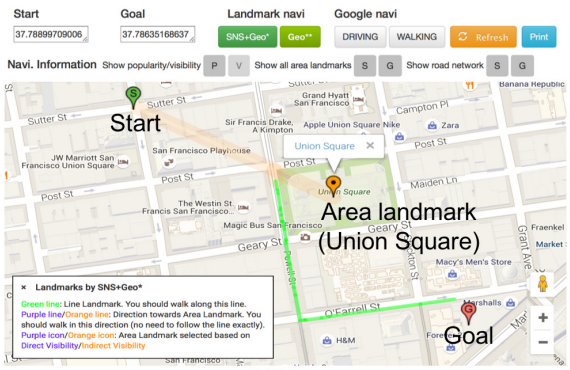
Formally, in a typical representation, R is the set of possible routes $R = \{r_1, r_2, ..., r_l\}$ between S and D such that $r_i = (S, p_1, p_2, ..., p_n, D)$ and p_j is the point for changing moving direction. Conventional route search systems try to find minimal routes r_i in terms of their length ($Dist(r_i)$) and/or the numbers of decision points ($Count(p_j)$).

On the other hand, in our approach, the set of possible routes between S and D is represented as $LR = \{lr_1, lr_2, ..., lr_m\}$ where $lr_i = (S, lp_1, lp_2, ..., lp_o, D)$ and lp_j is a navigation point. Same as in the conventional route recommendation systems, a navigation point can be a point for changing the moving direction (e.g., crossing at which a user needs to turn left or right). However, unlike in those systems, it can also be a landmark towards which or along which the user should walk. When lp_j is an area landmark, the path from lp_{j-1} towards lp_j until lp_{j+1} can be freely chosen by a user. This is because lp_j, being an area landmark, is only used for indicating the moving direction. Note that in the case of conventional routes the path from p_{j-1} to p_j is essentially a fixed route segment (bounded by two change points), which the user is required to follow, since otherwise she may get lost.

In the experiments conducted on two cities: San Francisco, USA and Kagoshima, Japan, we demonstrate that the landmark-based routes are more efficient than those given by Google Directions. We have also constructed an online system [14] to generate and test the proposed route recommendation method. Fig. 2 shows its interface.

The contributions we make are summarized as follows:

1. We introduce a novel type of navigation system that aims to output memorable and intuitive routes for users navigating in unfamiliar areas. To determine high-quality routes, we



(a) Map

The number of the used landmarks 3

#	Category	Name
1	Area-Indirect	Union Square
2	Line	POST ST
3	Line	MASON ST

(b) Textual instructions

(c) Images of area landmark

Figure 2: Interface of LandmarkNavi [14]. After selecting start, destination and the type of route graph, the system searches for a route and its directions.

Figure 3: Streets and intersections in SF.

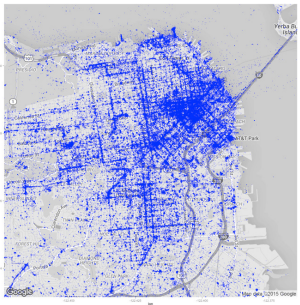


Figure 4: Geographic distribution of tweets in SF.

define three types of landmarks: point, area, and line, and estimate landmark's utility by analyzing crowd footprints data as well as 3D geographic data.

2. We evaluate the usefulness of routes with landmarks detected by the proposed method with real users in both the simulated real space and in the real space.
3. We demonstrate an online route recommendation system [14] for SF downtown, Kagoshima and Kyoto cities.

DATA MODEL

Road Network

In this work, we detect useful landmarks by exploiting not only geographic data but also crowd footprint data from LBSN. By "geographic data" we mean aggregate data about intersections, streets, buildings, and topological objects in a given area.

As a foundation, we utilize a road network that is an abstraction of a real road structure. A road network is represented as a graph, $G = (V, E)$, where V is the set of nodes that are intersections[1] and E is the set of edges, which are sequential components of streets S. Fig 3 shows as an example the road network in San Francisco downtown.

[1]Road ends are treated as intersections, too.

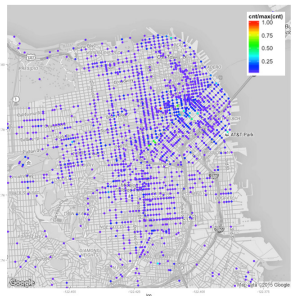


Figure 5: The ratio of tweets at each intersection in SF.

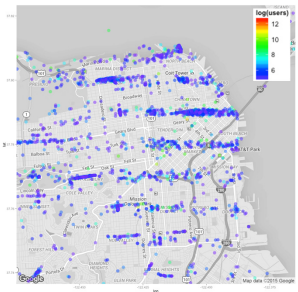


Figure 6: The ratio of check-ins at each place in SF.

Each intersection v is associated with the tuple: <vID, $coordinates$>. vID is the identification number for the intersection and $coordinates$ are its central coordinates. Each street $s \in S$ consists of several sequential segments: $< e_1, ..., e_n >$. Then, each segment e is represented by a tetrad of attributes: <eID, $street\ name$, vID_{from}, vID_{to}>. eID is the identification number for the segment of a street and the $street\ name$ is the name of street. vID_{from} and vID_{to} mean identification numbers for the from-node and the to-node of the segment.

Data Structure of Tweets

Each tweet t is represented as quintuple of basic attributes: <$tweetID$, $userID$, $content$, $timestamp$, $coordinates$>. The $tweetID$ and $userID$ are identification numbers for the tweet and the user, respectively. The $content$ is the text of the tweet, the $timestamp$ means the time when the tweet was written, and the $coordinates$ is GPS coordinates consisting of $latitude$ and $longitude$ of the tweet. We assign each tweet t to the geographically closest intersection v by computing the Euclidean distance between the coordinates of t and the coordinates of an intersection $v \in V$ in the road network. Thanks to this, the attributes of an intersection corresponding to each tweet <vID, $coordinates$> are added to the above-listed set of basic attributes of the tweet. Following an example of SF, Fig 4 shows the geographic distribution of tweets based on their original coordinates, while Fig. 5 represents the ratio of tweets corresponding to each intersection. The higher (lower) the ratio of the assigned tweets, the redder (more blue) the colours of the intersections are.

Constructing Place Database

In order to detect point and area landmarks in a given city, we need to first collect candidate places and to construct a place database (DB). As candidates we utilize places that have been added/edited by users of Foursquare (*e.g.*, restaurants, parks, or convenience stores.). Each place is represented as a sextuple: <pID, $name$, $coordinates$, $catID$, $checkins$, $users$>. pID is the identification number of the place, the $name$ is the place name, the $coordinates$ are GPS coordinates consisting of $latitude$ and $longitude$ based on the place address, and the $catID$ is the identification string of the primary category of the place. $checkins$ and the $users$ are attributes concerning crowd-based statics of the place. The former is the total number of check-ins at the place, while the latter is the total number of unique users who made check-ins at the place. Note that while the first 4 attributes of a place are static, the values of the $checkins$ and the $users$ are continuously updated.

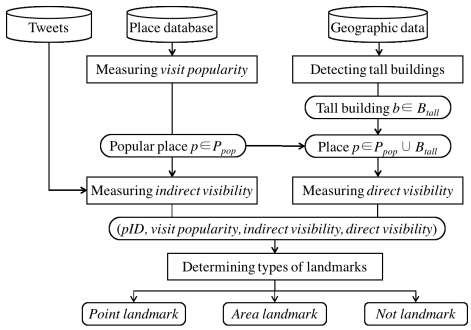


Figure 7: Chart flow for detecting landmark types (except the line landmarks) and for determining landmark's utility.

EXTRACTING USEFUL LANDMARKS

In this section we explain how to detect useful landmarks from the above-discussed candidate pool by exploiting geographic data and crowd footprint data from Twitter and Foursquare. Three indicators of each place are calculated for determining the candidate landmark's utility: *visit popularity*, *indirect visibility*, and *direct visibility*. Based on the values of these indicators, we will later distinguish between the point and area landmarks. Fig. 7 summarizes the way to detect the point and area landmarks. Finally, line landmarks will be detected by estimating *visit popularity* of each street.

Measuring Visit Popularity of Places

First, popularity of places is estimated by measuring *visit popularity* for each place. Fig. 6 shows the geographic distribution of places in SF city taken from the place DB. The colours ranging from blue to red are allocated based on the statistics of each place. We assume that the number of check-ins is correlated with the number of people within the place. Although there are several places characterized by high *visit popularity*, some of the check-ins are repeatedly made by the same users (*e.g.*, owners of these places). We use then the number of users, *users*, checked in each place as the *visit popularity* of the place. Places with high *visit popularity* are then selected to become candidates of point and area landmarks (see Fig. 7).

Measuring Indirect Visibility of Places

Here, we describe the way to compute *indirect visibility* for popular places detected by the method described above. Our approach relies on comparing the locations of users who mentioned places in their tweets with the locations of these places following the concept of *collective spatial attention* introduced in [1]. We can then find popular places that users tweet about from far away. The intuition here is that there should be many clues regarding such places that enable users to find effective ways to reach them, even, if the places may not be directly visible. For instance, a popular tourist spot 'Kinkaku-ji Temple (Golden pavilion)' in Kyoto, Japan cannot be seen from far away, yet, one can find various road signs (e.g., at intersections) that show directions to it, several buses bound for it and many related advertisements. Indeed, this temple is frequently mentioned in tweets originating from within other places (even distant ones) than its location.

To compute the indirect visibility, two characteristics of locations, *location mention* and *location difference*, are obtained using the attributes of tweets. The former is directly extracted from the place DB by matching the tweet content with a place name. If matched, the following tuple corresponding to a place is added to each tweet: <*place*, *location difference*> where *place* consists of a tuple of <*pID*, *coordinates*>. The value of *location difference* is a distance between the *coordinates* of *place* of a tweet and the *coordinates* of the *intersection* to which the tweet is assigned. When the average value of *location differences* for a given place is high, we regard the place as "indirectly visible" from distant intersections.

Algorithm 1: Measuring indirect and direct visibility. Note that the values of α and β are set based on the area size of the target city and the number of tweets at each intersection.

input : Popular places $P_{pop} \subset P$, Tweets T, Tall buildings $B_{tall} \subset B$
output : Indirect visibility and direct visibility of popular places

```
// Measuring Indirect Visibility
for place p in P_pop do
    // A set of tweets mentioning p
    T_p ← Match(p,T)
    if AvgDist(p,T_p) > α then
        for intersection v in V do
            // Declaring a set of indirect visibility of p
            IV_p ← ∅
            // Measuring p's indirect visibility at v, iv_p^v
            iv_p^v ← |T_p^v ⊂ T_p : ∃t ∈ T_p^v|
            if iv_p^v > β then
                // Storing place ID v.vID and indirect
                   visibility of p at v
                IV_p ← (v.vID, iv_p^v)
    // Measuring p's indirect visibility, in_vis_p
    in_vis_p ← Count(IV_p)
// Measuring Direct Visibility
for place p in P_pop ∪ B_tall do
    for intersection v in V do
        // Declaring a set of direct visibility of p
        DV_p ← ∅
        // Measuring p's direct visibility at v, dv_p^v
        dv_p^v ← CountPixels(p,v)
        if dv_p^v > δ then
            // Storing place ID v.vID and direct visibility
               of p at v
            DV_p ← (v.vID, dv_p^v)
    // Measuring p's direct visibility, d_vis_p
    d_vis_p ← Count(DV_p)
```

The first part of Algorithm 1 describes the detailed way to compute the indirect visibility, in_vis_p, for each popular place p ($\in P_{pop}$)[2]. The indirect visibility of places becomes higher if the places gather lots of attention from distant intersections.

Measuring Direct Visibility of Places

The third indicator (besides the above-discussed visit popularity and indirect popularity) for determining landmark's utility, *direct visibility*, is measured by analyzing 3D geographic data. In particular, high-rise buildings and towers seen from within larger areas can be detected by this measure. The latter part of Algorithm 1 describes the procedure for measuring direct visibility.

We use a map involving 3D shape information of all buildings in a city with 3D computer graphics (3DCG). The top n tallest

[2]Note that a place whose name indicates multiple locations in the place DB such as 'Starbucks' and '7-eleven' is ignored as it is usually difficult to disambiguate *location mention* without any additional context.

(a) Rendered image (b) Actual scene

Figure 8: Rendered image (a) of an actual scene (b) in SF.

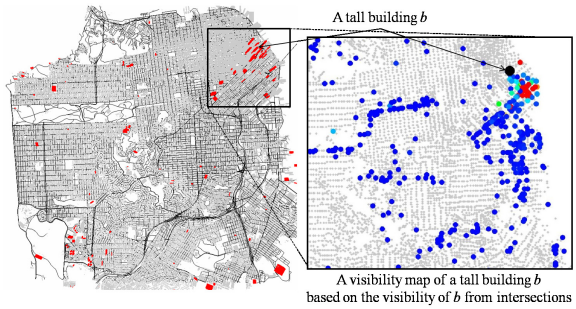


Figure 9: Tall buildings and a visibility map. Left: Tall buildings (red shapes) extracted based on their relative heights. Right: A visibility map of a building (denoted by a black point). Grey points show intersections and coloured points ranging between blue and red represent intersections from where the building can be seen.

buildings are selected from all the buildings in each block (m km^2 size) that is set by dividing a city.

The next task is to construct a visibility map of each building to detect its visible area. This is done by checking what can be seen from intersections. For this, we first render 360 degree hemisphere from each intersection using standard CG library and OpenGL[3]. Next, unique IDs and colours are assigned to each building to be used in the rendered 3D images. Unique colours are also assigned to topographical objects and to background. Then, a panoramic view of 360 degrees of the city from each intersection on the road network is rendered by applying 3DCG techniques. Fig. 8(a) is an example of the rendered image and Fig. 8(b) is its actual scene seen from the same intersection. Rendering buildings makes it possible to determine whether the buildings can be seen from intersections and thus to accurately approximate their visibility at the intersections. The determination is performed by computing the number of pixels of visible parts of the place (in the function $CountPixels(p,v)$ in Algorithm 1).

Fig. 9 shows the map indicating tall buildings in San Francisco and an example visibility map of one of such buildings. In this figure, intersections from where this building (represented by a black point) can be seen are shown by points with the colour ranging from blue to red. The colours become redder as the visibility of the building is higher. The grey points represent intersections from which the values of visibility of the building are less than 30 pixels[4].

[3]We render the image with 3000*1000 pixels with the cylindrical coordinates system, and thus, the ratio of 30 pixels varies depending on the azimuth angle. Since most buildings stand near the horizontal level or above, it is around 0.5-1.0 degrees field of view.

[4]The threshold was decided by examining the visibility of landmarks by preliminary human testing using Google Street View.

Table 1: Landmark types based on their indicators.

Pattern	Visit popularity	Indirect visibility	Direct visibility	Landmark Type
1	High	Low	Low	Point
2	High	Low	High	Area
3	High	High	Low	Area
4	High	High	High	Area
5	Low/Not measured	Not measured	High	Area
6	Low/Not measured	Not measured	Low	Not

Classifying Places into Point and Area LMs

We describe now how to assign a landmark type based on the values of the computed indicators. We focus first on area and point landmarks. The extraction of line landmarks will be explained in the next section.

Table 1 shows patterns of the values of the three indicators and the final classification of each place to the one of the following classes: *Point landmark*, *Area landmark*, and *Not landmark*. Places characterized only by high visit popularity are regarded as *Point landmark* (pattern 1). We determine *Area landmark* if the values of either indirect visibility or direct visibility of a place are high. Therefore, places corresponding to patterns 2, 3, 4, and 5 are classified into *Area landmark*. Especially, the places falling into the pattern 4 and those in the pattern 5 are extracted and characterized using only the crowd footprint data and geographic data, respectively. Note that our method calculates the indirect visibility of only places having high visit popularity. Places matching the pattern 6 are not popular and are difficult to be seen from further intersections even if they are tall buildings. Therefore, we do not regard them as useful landmarks (*Not landmark*). All the places classified as a point or area landmark are stored in a landmark database.

Extracting Line Landmarks

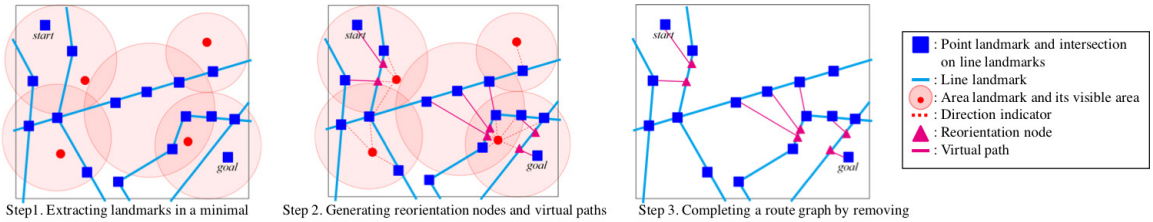
We next describe how to find line landmarks. Assuming that crowded streets are easier to be found than less crowded ones, we will regard high traffic streets as line landmarks. We thus detect the line landmarks by estimating *visit popularity* of streets based on the gathered tweets.

In our method to extract line landmark, we simply assume that most of the tweets originate from pedestrians and the number of tweets is correlated with the number of people within the place. Under these assumptions, we first extract the set of crowded intersections $V' \subset V$, which are determined based on the number of tweets assigned to these intersections. Next, we search for the set of sequential crowded intersections corresponding to segments of the same street V'_s ($\subset V'$). Each street $s \in S$ is then scored by the total number of tweets sent from its respective crowded intersections $v \in V'_s$. The value of the score of each street is then considered as *visit popularity* of the street. Finally, the set of streets $S_{pop} \subset S$ with high *visit popularity* are extracted as line landmarks and are stored in the landmark DB.

SEARCHING FOR ROUTES

Constructing Simple Route Graph

Since there can be many possible routes leading toward area landmarks, it is non-trivial to use point landmarks and area landmarks together. To solve this problem we generate a route graph that connects point landmarks and area landmarks using line landmarks. In addition, a *virtual path* is generated for aggregating the possible routes directing an area landmark.



Step1. Extracting landmarks in a minimal bounding rectangle

Step 2. Generating reorientation nodes and virtual paths for connecting each area landmark and point landmarks or intersections on line landmarks in its visible area

Step 3. Completing a route graph by removing area landmarks and direction indicators

Figure 10: Route graph construction using useful landmarks.

We discuss the entire process below (it is also summarized in Fig. 10). First, based on the extracted landmarks, we construct a route graph *RG* obtained by setting two points, a starting point *start* and a destination point *goal*. *RG* consists of *RV* and *RE*, which are respectively the set of nodes and the set of edges. Next, a minimal bounding rectangle is formed based on the two points, and landmarks inside the rectangle are extracted from the landmark DB as shown by Step 1 in Fig. 10. In the figure, small blue rectangles indicate point landmarks or intersections on line landmarks, which are used as nodes in *RV*. Cyan lines represent line landmarks, red points mean area landmarks and red circles represent *visible areas* of the area landmarks computed based on their indirect/direct visibility [5]. Note that the starting and destination points are assigned to the closest nodes for facilitating the route search.

Next, to connect each area landmark and the corresponding nodes in its visible area, additional edges called *direction indicators* (red dotted lines) are generated between them (see Step 2 of Fig. 10). Also additional nodes (magenta triangles) called *reorientation nodes* are set on the intersection of a line landmark and the direction indicators. A segment from the node to any reorientation node is extracted as a virtual path (magenta lines). Here, each virtual path is used only for indicating the directions towards a corresponding area landmark when moving from other nodes positioned within the landmark's visible area. In other words, virtual paths allow users to freely choose their ways toward area landmarks until the users reach the next landmarks, which are represented by the reorientation nodes in the route graph.

Consequently, *RV* (the set of nodes) consists of point landmarks, intersections on line landmarks, and the reorientation nodes. *RE* (edge set) contains line landmarks and virtual paths (see Step 3 of Fig. 10). Each edge $re \in RE$ has the assigned ID that is used for route search as follows: If the edge is a line landmark, it has the line landmark's ID. If the edge is a virtual path, it has the ID of the virtual path and the one of the corresponding area landmark. Although area landmarks are not included in the route graph because they make unnecessary routes such as round routes, they are indirectly indicated by the assigned *landmarkID*. Therefore, a route with area landmarks can be searched on the generated route graph.

Route Search

In order to search for a route, we can apply various route search algorithms such as Dijkstra algorithm and Genetic Algorithm (GA) [11]. We choose the latter one in this work[6]. A route $lr \in LR$ from a starting node rv_{start} to a destination node rv_{goal} is represented as a sequential list consisting of n nodes, $lr = \{rv_{start}, \ldots, rv_i, \ldots, rv_{goal}\}$ (as if a variable-length chromosome was utilized). Then, we define an evaluation function $Cost(lr)$ and determine the most appropriate route lr that can minimize $Cost(lr)$.

$$Cost(lr) = \sum_{i=1}^{n-1} Cont(re_{i-1,i}, re_{i,i+1}) + \lambda \sum_{i=0}^{n-1} Dist(rv_i, rv_{i+1})$$

Cont is a Boolean function for checking the continuity of landmarks used in continuous edges $re_{i-1,i}$ (between rv_{i-1} and rv_i) and $re_{i,i+1}$ (between rv_i and rv_{i+1}) in *RE*. It returns 1 if *landmarkID*s of these edges are different, or 0 if same. $Dist(rv_i, rv_{i+1})$ is a function to calculate the Euclidean distance between two nodes rv_i and rv_{i+1} of the edge $re_{i,i+1}$. This evaluation function makes it possible to shorten the length of a route to decrease the number of landmarks contained in the route and to increase the visible ratio of area landmarks from the route. In addition, we can easily adjust the balance whether to give priority to the length of route or to the number of landmarks by changing parameter λ. Since we generate a simple route graph when compared to an actual road network, optimal routes can be found in manageable time.

IMPLEMENTATION

User Interface

Fig. 2(a) shows the prototype system's interface that recommends memorable routes. First a user specifies the start and destination points on a map. Next, he or she selects a type of a route graph to be generated by pressing a button above the map. In the current implementation, two types of route graphs are available: (i) a route graph constructed with landmarks extracted by considering both indirect and direct visibility measured using both tweets and geographic data (SNS+Geo) and (ii) a route graph consisting of landmarks extracted based on direct visibility measured using geographic data only (Geo). For comparison, the system provides also routes searched on the road network (by Google Directions). Recommended route will be displayed on the map and its textual instructions will be displayed next to the map. In addition, the system offers two functions for visualizing visibility and popularity on the map. If SNS+Geo or Geo is selected and the searched route includes an area landmark extracted based on the direct visibility, the visible intersections of the area landmark will be highlighted in color ranging from blue (lower visibility) to red (higher

[5]Note that to simplify the explanation, the visible areas are illustrated as circles, although their actual shapes are more complex.

[6]Note however that other approaches could be selected instead.

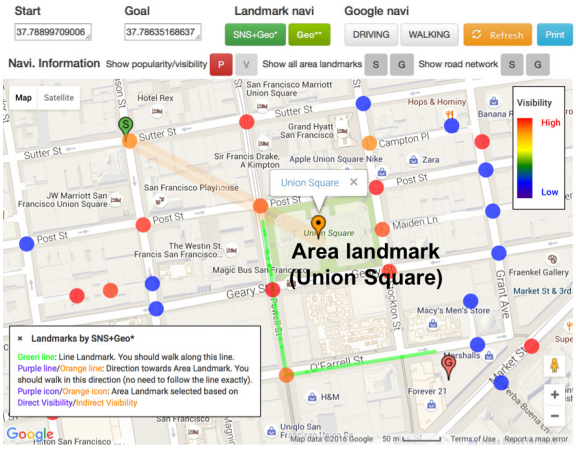


Figure 11: Popularity map of an area landmark.

visibility). On the other hand, if SNS+Geo is selected and the searched route has an area landmark extracted based on the indirect visibility, corresponding intersections of the area landmark will be colored ranging from blue (lower popularity) to red (higher popularity) as shown in Fig. 11.

Finally, the system also provides images of area landmarks. The user can view them when clicking landmark name as shown in right side of Fig. 2(c).

Datasets

We have constructed datasets for two geographically and culturally different cities: San Francisco, CA, USA (population: 840K) and Kagoshima, Japan (population: 610K). Table 2 shows the statistics of both the datasets.

San Francisco City Downtown

We extracted 600K tweets issued by 55K users in SF city between 2013/9/25 and 2014/1/17. We then filtered out tweets of potential bots or spammers by setting the upperbound limit on tweet count by a single user[7]. Consequently, the dataset contained 0.57M tweets. Fig. 4 shows the tweets' distribution within the SF downtown. We have also constructed the place DB for SF city by gathering basic attributes and crowd-sourced statistics of 25K places using the snapshot of Foursquare data as of the end of Sept., 2015. The remaining data concerning SF city was obtained from DataSF which officially provides open data of SF city [2].

Based on this dataset, we then extracted 179 point landmarks, 549 area landmarks, and 45 line landmarks consisting of 3,294 segments. Fig. 12 shows the route graph constructed with the landmarks. We indicate the extracted area landmarks in red and also present the visibility map of one of the area landmarks (represented by the black point).

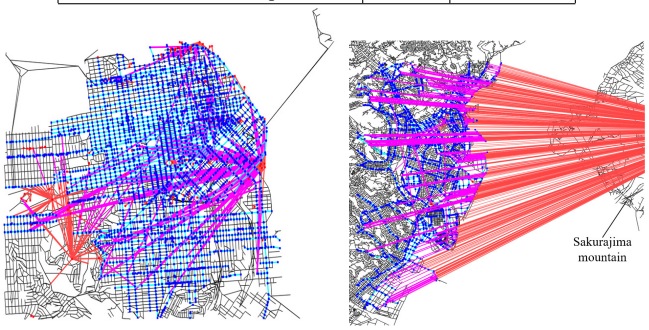
Kagoshima City

We used 98K tweets by 4.6K users in Kagoshima city between 2013/8/15 and 2014/1/14. The place DB of Kagoshima city consists of 0.4K places' basic attributes and crowd-sourced statistics. The geographic data contains building shapes from a residential maps of Japan by ZENRIN [26], while the road data has been obtained from the Geospatial Information Authority of Japan [7]. Using the dataset, we extracted 452 point landmarks, 11 area landmarks, and 290 line landmarks consisting of 3,224 segments. The map in Fig 12(b) gives the route graph. Note that the graph has quite different shape than the one for SF due to the key area landmark of Kagoshima city - Sakurajima volcano located on a nearby island, which is characterized by very high indirect and direct visibility.

[7]The threshold has been empirically set by checking the tweets issued by the top users ranked in the descending order of the tweet frequency.

Table 2: Datasets of two cities.

		SF	Kagoshima
Geographic data	streets	1,233	61,075
	intersections	9,649	30,703
	buildings	85,116	66,189
Geo-tagged tweets from Twitter		600K	98K
Places from Foursquare		25,256	383



(a) San Francisco downtown (b) Kagoshima

Figure 12: Route graphs overlaid on maps. (Blue points: point landmarks, cyan lines: line landmarks, red points: area landmarks, and magenta lines: virtual edges.)

EXPERIMENTAL EVALUATION

In this section we describe experiments conducted to evaluate our system [14]. We have set the experiments such that users walk through areas unfamiliar to them. Note that finding a large number of strangers to perform experiments in a given city is however not easy. Thus, for simplicity, in one case (San Francisco) we employ a virtual space approach to evaluate the system by simulating the real-world using Google Street View (SV) [9]. In the other case (Kagoshima), we perform an experiment both in the real space and in the virtual space.

Evaluation on SF Dataset

Settings

To evaluate the quality of recommended routes we compare our proposed system with an existing route search system: Google Directions (*GR*) [8]. Furthermore, in order to confirm the benefit of using crowd footprint data, we generate two types of routes. The first one (*LR*) uses landmarks extracted by considering both indirect and direct visibility measured using both tweets and geographic data, while the second one (*VR*) uses landmarks extracted by only considering direct visibility measured using geographic data. Our target user group are pedestrians or bicyclists. Hence, we set relatively short distances between the start and destination points. *GR* baseline was also set to use the walking mode.

We then randomly selected three pairs of start and destination points. As a result, we tested 9 routes (3 routes $\times$ 3 methods). The distances of the routes are between 0.8km and 2.0km. Fig. 13 shows examples of three routes between $start_2$ and $goal_2$: LR_2 is a route with landmarks extracted by considering the three indicators of landmark's utility, VR_2 is a route with

Table 3: Examples of route directions in SF downtown. (<.>: Point landmarks, (.): Area landmarks, [.]: Line landmarks.)

(a) LR_2: Indirect + Direct visibility	(b) VR_2: Direct visibility	(c) GR_2: Google Directions
(1) Go straight on **[2nd St] towards (Stadium)** (2) Turn right onto **[King St]**	(1) Go straight on **[2nd St]** (2) Turn right onto **[Brannan St]** (3) Go straight towards **(a building)** around **<Zeke's Diamond Bar>** (4) Turn left onto **[Townsend St]**	(1) Head southwest on S Park St 112 ft (2) Turn right to stay on S Park St 0.1 mi (3) Turn right to stay on S Park St 171 ft (4) Turn left onto 3rd St 0.2 mi (5) Turn right onto Townsend St 459 ft (6) Turn left at Lusk St 194 ft

landmarks extracted by 3DCG-based method, which determines direct visibility for only tall buildings, and GR_2 is a route searched by Google Directions walking mode.

Primarily, we should perform the evaluation in the real space. However, these are too expensive and time-consuming tasks to carry out [12], especially, for participants who live far away. Therefore, as mentioned before, to evaluate the recommended routes on the SF dataset, we let users walk in a simulated real space using SV. However, we further confirm these results by investigating the results performed on both the virtual and real spaces of the same city (Kagoshima).

36 students (28 males and 8 females) in their 20's were recruited for the study. All of them have never been to SF. [8] After a brief training, upon receiving a route, each user had 2 min to read the route description and to memorize it. Then relying on her memory, the user tried to reach the destination in a simulated real space by operating SV[9].

The materials about the recommended routes shown to the users included textual route directions from a starting point *start* to a destination *goal* (see Table 3 for examples) and a route path displayed on a map as well as images of landmarks, if any were included in the routes (see Fig. 13). During the experiments, these materials and a small screen enabling each participant to check his/her current location on a map (which is typically shown in the corner of SV interface) were hidden from users unless the subjects asked to see them.

Users were asked not to check the route directions and their own locations as much as possible until they got lost. Note that the route directions based on landmarks include virtual paths, which allow users to freely move toward next landmarks from point/line landmarks connecting virtual paths. To evaluate the quality of routes, we record three evaluation items for each trial as follows.

i) time (min.): Time needed to reach the destination.
ii) route ref.: Number of times a user checks route directions on printed materials.
iii) self-position ref.: Number of times the self-positions were checked with Google Maps.

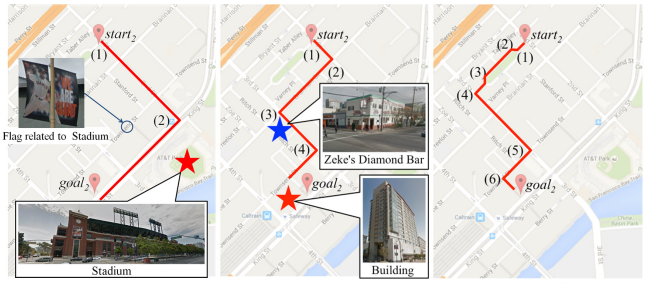
When analyzing results of the participants, we cut the best and worst score from each route and normalized the scores based on the distance of the route.

Results

First, we report the average route length in terms of the number of navigation points. On average, the routes generated by *LR*, *VR*, and *GR* have 3.0, 4.0, and 6.3 navigation points. This implies that *LR* allows obtaining the shortest routes in terms of points necessary to be remembered and, hence, *LR* outputs the most memorable routes.

[8]Note that their cognitive abilities were not considered in this experiments.

[9]We asked participants to use arrow keys for making the operation speed uniform.



(a) LR_2: Indirect + Direct visibility (b) VR_2: Direct visibility (c) GR_2: Google Directions

Figure 13: Examples of routes between $start_2$ and $goal_2$ by three methods (*LR*, *VR*, and *GR*). Red star indicates an area landmark, blue star is a point landmark and red line is line landmark. Note that a stadium, an area landmark with high indirect visibility, is not tall, and thus, is visible only from nearby locations, yet some cues can be noticed (e.g., flags indicated in (a)).

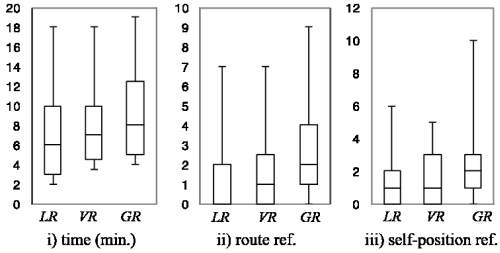


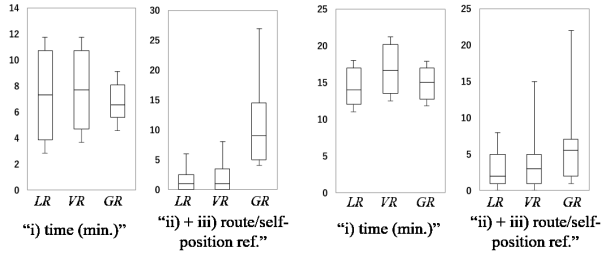
Figure 14: Boxplots of the evaluation items (SF).

Next, in Fig. 14 we show box plots that contain the other evaluation items of the three tested methods: the time spent from a start to a destination, ii) the route directions' reference count, and iii) the self-position's reference count. As mentioned above, these were recorded during the experiment. We can observe that the average time of *LR* was shorter than the ones for the other two methods. Moreover, its number of route references was smaller, too. Compared to *GR*, the frequency of checking the number of self-positions is also decreased. Furthermore, the deviations of the counts of references of routes and of self-positions for LR were smaller than the ones for the other two tested methods. We found significant differences in the counts of references to route directions and the numbers of self-position references between *LR* and *GR* based on the student's t-test (p<0.05).

Evaluation on Kagoshima Dataset

Settings

We performed the next experiments by hiring 30 students in their 20's to walk in both the virtual space using SV and



(a) Virtual space (b) Real space

Figure 15: Boxplots of the evaluation items (Kagoshima).

in the real space. Basically, the procedure and restrictions concerning these evaluations were almost same as in the case of SF. However, as for evaluation items, we merged ii) route ref. and iii) self-position ref. to a single item ("ii) + iii) route/self-position ref.") as it is difficult to distinguish them when walking in the real space.

We randomly selected six pairs of a start and a destination point. Then, we searched for routes by the three methods per every pair of a starting point and a destination. As a result, we evaluated 18 routes (6 routes $\times$ 3 methods) both in the virtual space and in the real space. The distances between the start and destination points range from 1.5km to 2.6km.

Results

The landmark-based routes were found to be simpler than the ones by Google Directions. The average number of navigation points of *LR* was again smaller (5.0) than for *VR* (5.3) and for *GR* (11.2). Fig. 15(a) and 15(b) show the distributions of the results of the evaluation items for routes by *LR*, *VR*, and *GR* in the virtual space and in the real space. From these graphs, we see that, on average, *GR* enabled to reach destinations earlier than *LR* and *VR*, though the significant differences were not confirmed by Student's t-test ($p \leq 0.05$). On the other hand, participants using *GR* more frequently checked route directions printed on their maps and self-positions with their devices. As for the count of route/self-position references, we could confirm significant differences (with $p \leq 0.05$) by comparing *LR* and *VR* with *GR*. This was observed in both the results for the virtual space and the real space. Since we could observe similar tendencies in terms of the route and self-position reference count, we believe that performing evaluations in a simulated space on SV makes sense.

Qualitative Evaluation

We have also conducted a brief survey after each experiment to obtain participants' opinions and to reason about the quality of the suggested routes. The questions were as follows: q1) "*Did you think that the route directions were easy to remember when you check them in the printed map at first?*", q2) "*Did you think that the directions you memorized were reliable?*", q3) "*Did you think that it is easy to follow the memorized route directions?*", and q4) "*Did you think that the route directions were useful as route navigation?*". The participants were required to provide answers on the 5-point Likert scale [16]: 1: not at all, 2: a little, 3: so so, 4: much, 5: very much.

Table 4 shows the average scores from all the experiments. We could confirm that the participants were more satisfied with route directions by the proposed method than those by the other two methods. The average scores for route directions by the proposed method were higher than by Google Directions. According to Student's t-test, the difference between *LR* and *GR* was statistically significant in terms of q1) related to *memorability* ($p = 0.00$), q3) concerning *easiness* ($p = 0.00$), and q4) related to *usefulness* ($p = 0.02$) under $p \leq 0.05$. As for the difference between *LR* and *VR*, we could not observe statistical significance.

Table 4: Questionnaire scores.

	q1	q2	q3	q4
LR	4.5	4.1	3.9	3.9
VR	4.2	3.8	3.6	3.8
GR	3.1	3.2	2.6	3.2

In addition, we asked the users about the advantages and disadvantages of different route directions. As for *GR*, it was hard for several participants to remember many decision points. Other users who participated in the experiments in the real space mentioned that it is difficult to identify streets in route directions, as most streets in Japan do not have names.

Discussion

n this study, evaluations were performed in the simulated real space using SV (SF and Kagoshima cities) and in the real space (Kagoshima city). We could observe similar patterns of results in both the spaces, that is, *LR* is better than the other methods for all the three types of the evaluation criteria. We believe that the results obtained in the virtual space on SF dataset are correct considering that the ones obtained in real and virtual space on Kagoshima dataset are quite similar. Therefore, an evaluation method using SV can be regarded as one of alternative testing means, especially, when it is difficult to perform full scale experiments in the real space. Note that SV has limitations such that it is impossible to test the system during evening hours, poor weather conditions, or during occurrences of unusual events like demonstrations. Therefore, in the experimental evaluation, we tested the performance of our system only under typical conditions of daytime hours and sunny/cloudy day. The other limitation is that eye level of SV is different from the one in the real space. In the experiments, however, the eye level is aligned for all the subjects.

Interestingly, our method could recommend a stadium as an area landmark (Fig. 13(a)), though the stadium was not detected by 3DCG-based method which only considered the height of a building. Actually, there were flags (contextual clues) near the stadium which allowed wayfinders to guess they are near the stadium (Fig. 13(a)), even if they could not directly see it. We believe that such type of landmarks could be useful during night time navigation in cities since they are frequently mentioned on microblogs in night time, whereas high tall buildings may be less visible at night.

Overall, our system could achieve memorable route recommendation enabling users to move without the need for GPS and maps by appropriately extracting the three types of useful landmarks exploiting crowd footprints data over LBSNs and geographic data for continuous visual feedback. On average, the paths recommended by our method may not be always the shortest; however, usually, only a small number of landmarks are used, few enough to be remembered by an average user. Note that according to psychological and cognitive studies

[20] most adults can store between 5 and 9 distinct items in their short-term memory (7 ± 2).

User interviews showed also that some participants tried to find landmarks in Google Directions' paths by themselves, as it is difficult to remember lots of decision points. This is likely due to the ability of landmarks to increase users' spatial awareness by informing them about surroundings and by decreasing the effort needed for constructing mental representation of cities.

Although we focused on two cities, our method can be applied to other cities. Geographic data like road networks, intersection and terrain information are usually provided by public sector and are free or relatively cheap to be acquired. When it comes to 3D data, DEM (digital elevation map) tend to be also inexpensive or free. The polygon data for buildings are mainly constructed by private companies (Zenrin in Japan, Google and Microsoft worldwide) and are usually not free. However, like in the SF case, the importance of such data is being increasingly acknowledged and we believe it will finally be made by public sector, too.

Lastly, we provide an online system [14] for generating routes by the proposed approaches and for comparing them with the ones by Google Directions. Figs. 2 and 11 demonstrate the snapshot of its interface. It shows a recommended route from a given start to a given destination point as well as the degree of indirect/direct visibilities of a landmark, which belongs to the route, at its nearby intersections. These help to understand from where the landmark is visible (e.g., in case users would get lost). In Fig. 11, when "Union Square" is selected as an area landmark, all the nearby intersections can be colored based on the landmark's visibility from these intersections.

RELATED WORK

Landmark Extraction

Landmarks are important not only for route recommendation, but also for facilitating understanding of a city to support decision making and urban planning. To automatically extract landmarks, most of prior approaches used NLP techniques over textual content. Furlan *et al.* [6] presented schemes for automatically extracting and classifying landmarks based on the way that they are used in space-related sentences. Due to the popularity of LBSNs, several methods for ranking and recommendation of POIs have been proposed [15, 27]. All these works however do not focus on route recommendation.

When it comes to route selection, Raubal *et al.* [22] introduced measures to formally specify the landmark saliency of a feature for generating wayfinding instructions equipped with landmarks. They presented three types of attractions for landmarks: visual, semantic, and structural, and quantified them using city maps, street graphs, geo-referenced photos and content. To extract landmark's characteristics, Kallioniemi *et al.* [13] discussed the attractiveness of landmarks based on their visual, structural and semantic properties. These methods are related to our work in terms of the purpose to extract and measure landmark's utility. However, we define and utilize not only local and global landmarks but also line landmarks. In addition, unlike the above works, our method measures the landmark's utility by calculating three indicators based on SNS data and geographic data.

Automatic Route Recommendation

Various navigation systems have been developed to satisfy users' diverse requirements for reaching destinations in unfamiliar environments. For example, Shao *et al.* [25] performed easiest-to-reach neighbor search. Sacharidis *et al.* [23] tried to find routes as simple and as fast as possible on road networks.

Few route recommendation systems exploiting crowd-sourced data have been developed so far. The one proposed by Quercia *et al.* [21] suggests routes that are not only short but also pleasant (beautiful, quiet, and happy) and stress-free. Fu *et al.* [5] developed an approach enabling to find convenient travel itinerary that avoids high crime areas. Hile *et al.* [10] automatically generated landmark-based pedestrian navigation using an online collection of geo-tagged photos. Unlike all these works, our method recommends user-friendly routes consisting of fewer navigation points. These points include selected landmarks enabling wayfinders to more freely move and to better orienteer themselves in unfamiliar terrains.

Urban Dynamics Monitoring

There have been various studies for observing or predicting ever-changing urban cities. Among GPS trajectory-based approaches, Shan *et al.* [24] demonstrated automatic map update by analyzing unmatched trajectories with road networks that had not been frequently updated on a digital map. Fan *et al.* [4] performed a short-term prediction based on the recent human movement observations.

CONCLUSIONS

One motivation behind this research is to facilitate navigation in unfamiliar cities and to decrease the need for users to look at smartphones while walking or cycling. We demonstrate an effective route search system that recommends memorable routes, which, at the same time, are short. For this we define *useful landmarks* and introduce a novel methodology for detecting them. Specifically, we extract three types of landmarks by measuring *visit popularity*, *indirect visibility*, and *direct visibility* and by exploiting geo-tagged tweet data, places and check-ins data from Foursquare, as well as geographical data from digital city maps. In this aspect our work is an example of *effective combination of real space data with the one harvested and computed from online social media*. We also propose line landmarks as a new type of landmarks to smoothly "connect" point (local) and area (global) landmarks for efficient navigation. In the experiments, we confirm that routes with landmarks extracted using both the indirect visibility and direct visibility are easier to remember and to follow than the routes with landmarks extracted considering only the direct visibility and the ones by Google Directions.

In the future, we plan to extend our experiments, investigate temporal patterns of visit popularity and visibility by exploiting timestamp information of SNS data, and classify the data in terms of time and weather conditions. We will also utilize place category to help users locate landmarks.

Acknowledgments

This research was supported in part by JSPS KAKENHI Grant Numbers 16K16057, 16H01722, 15K00162, and MIC/SCOPE 150201013.

REFERENCES

1. E. Antoine, A. Jatowt, S. Wakamiya, Y. Kawai, and T. Akiyama. Portraying collective spatial attention in twitter. In *KDD '15*, pages 39–48, 2015.
2. DataSF. `https://data.sfgov.org`.
3. M. Duckham, S. Winter, and M. Robinson. Including landmarks in routing instructions. *J. Locat. Based Serv.*, 4(1):28–52, Mar. 2010.
4. Z. Fan, X. Song, R. Shibasaki, and R. Adachi. Citymomentum: An online approach for crowd behavior prediction at a citywide level. In *UbiComp '15*, pages 559–569, 2015.
5. K. Fu, Y.-C. Lu, and C.-T. Lu. Treads: A safe route recommender using social media mining and text summarization. In *SIGSPATIAL '14*, pages 557–560, 2014.
6. A. Furlan, T. Baldwin, and A. Klippel. Landmark classification for route directions. In *SigSem '07*, pages 9–16. Association for Computational Linguistics, 2007.
7. Geospatial Information Authority of Japan. `http://www.gsi.go.jp/ENGLISH/page_e30031.html`.
8. Google Directions. `https://developers.google.com/maps/documentation/directions/`.
9. Google Maps Street View. `https://www.google.com/maps/views/streetview?gl=us`.
10. H. Hile, R. Vedantham, G. Cuellar, A. Liu, N. Gelfand, R. Grzeszczuk, and G. Borriello. Landmark-based pedestrian navigation from collections of geotagged photos. In *MUM '08*, pages 145–152, 2008.
11. J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1992.
12. S. Janarthanam, O. Lemon, and X. Liu. A web-based evaluation framework for spatial instruction-giving systems. In *ACL '12*, pages 49–54. Association for Computational Linguistics, 2012.
13. P. Kallioniemi and M. Turunen. Model for landmark highlighting in mobile web services. In *MUM '12*, pages 25:1–25:10, 2012.
14. Landmark Navi (online DEMO site). `http://www.ibe.kagoshima-u.ac.jp/~lmnavi/index_en.html`.
15. X. Li, G. Cong, X.-L. Li, T.-A. N. Pham, and S. Krishnaswamy. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In *SIGIR '15*, pages 433–442, 2015.
16. R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 140(55), 1932.
17. K. Lynch. *The Image of the City*. MIT Press, 1960.
18. P.-E. Michon and M. Denis. When and why are visual landmarks used in giving directions? In *Spatial Information Theory*, volume 2205, pages 292–305. 2001.
19. J. L. Nasar and D. Troyer.
20. L. R. Peterson and M. J. Peterson. Short-term retention of individual verbal items. *Journal of experimental psychology*, 58(3):193–198, 1959.
21. D. Quercia, R. Schifanella, and L. M. Aiello. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *HT '14*, pages 116–125, 2014.
22. M. Raubal and S. Winter. Enriching wayfinding instructions with local landmarks. In *Geographic Information Science*, volume 2478, pages 243–259. 2002.
23. D. Sacharidis and P. Bouros. Routing directions: Keeping it fast and simple. In *SIGSPATIAL '13*, pages 164–173, 2013.
24. Z. Shan, H. Wu, W. Sun, and B. Zheng. Cobweb: A robust map update system using gps trajectories. In *UbiComp '15*, pages 927–937, 2015.
25. J. Shao, L. Kulik, and E. Tanin. Easiest-to-reach neighbor search. In *GIS '10*, pages 360–369, 2010.
26. ZENRIN CO., LTD. `http://www.zenrin.co.jp/english/`.
27. J.-D. Zhang and C.-Y. Chow. Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *SIGIR '15*, pages 443–452, 2015.