Aurigo: An Interactive Tour Planner for Personalized Itineraries

Alexandre Yahi*, Antoine Chassang*, Louis Raynaud*, Hugo Duthil*, Duen Horng (Polo) Chau
Georgia Institute of Technology
{alexandre.yahi, antoine.chassang, l.raynaud, hduthil, polo}@gatech.edu

ABSTRACT

Planning personalized tour itineraries is a complex and challenging task for both humans and computers. Doing it manually is time-consuming; approaching it as an optimization problem is computationally NP hard. We present Aurigo, a tour planning system combining a recommendation algorithm with interactive visualization to create personalized itineraries. This hybrid approach enables Aurigo to take into account both quantitative and qualitative preferences of the user. We conducted a within-subject study with 10 participants, which demonstrated that Aurigo helped them find points of interest quickly. Most participants chose Aurigo over Google Maps as their preferred tools to create personalized itineraries. Aurigo may be integrated into review websites or social networks, to leverage their databases of reviews and ratings and provide better itinerary recommendations.

Author Keywords

User Interfaces; Visualization; Recommendation; Tour itinerary planning

ACM Classification Keywords

H.3.3 Information storage and retrieval: Information Search and Retrieval; H.5.2 Information interfaces and presentation (e.g. HCI): User interfaces

INTRODUCTION

The Internet has become the leading source of information for trip planning [18], dramatically transforming how travel planning would be performed, prompting users to shift from using printed materials like travel guides and brochures, to leveraging more online resources, like travellers' comments and suggestions on online forums, review websites, and blogs.

Trip planning is a challenging problem that has been widely studied. From a mathematical point of view, trip planning can be viewed as an optimization problem under constraints (e.g., in time, costs, popularity of places), its simplest version being the famous travelling salesman problem (TSP). However,

*The four first authors contributed equally to this work

IUI 2015, March 29–April 1, 2015, Atlanta, GA, USA.

http://dx.doi.org/10.1145/2678025.2701366

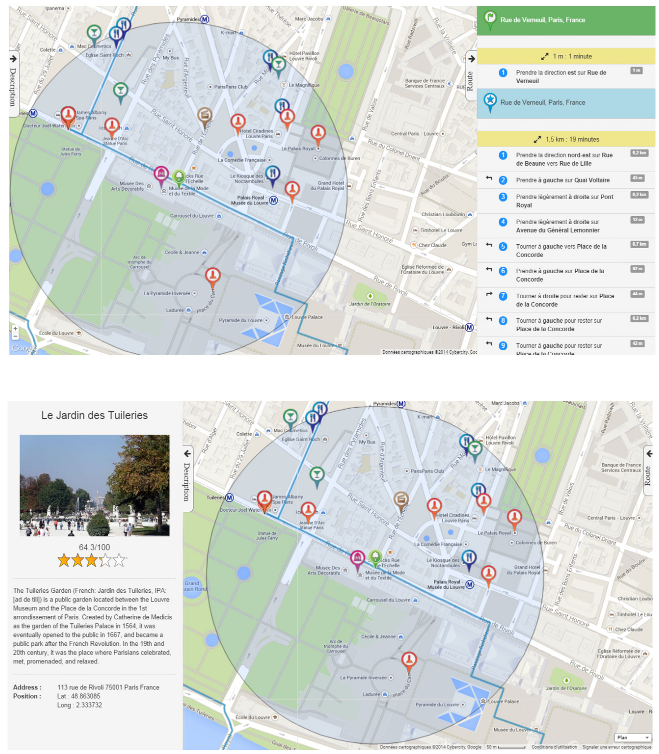

Figure 1. Screenshots of Aurigo showing the path of an itinerary generated by its recommendation algorithm. Points of Interest (POIs) are displayed within the blue area of the *Pop Radius*, a novel feature provided by Aurigo for interactively selecting POIs to fine-tune a path. The collapsible right panel (top image) shows the directions of the itinerary. The left panel (bottom image) describes a POI ("le jardin des Tuileries") with a picture, rating and text description.

this optimization problem is NP hard [3], which led to two predominant streams of research: 1) developing algorithmic solutions that try to find the best possible itineraries, often through approximation and heuristics; and 2) learning from human behaviors. The latter appears as a reasonable solution to mitigate this parameter-intensive optimization problem.

In the tourism industry, it has been widely acknowledged that tour planning tools should be designed to allow users to develop *memorable experiences* (ME) while building their itineraries [16], which means that besides considering quantitative data and objective requirements like price, duration, or distance, a desirable itinerary building process should also take into account the user's cognitive, behavioral and affective experiences. The Internet and online social media have made the sharing of such information possible and often enjoyable, where users' participation is often driven by altruism or community-related motivations [12, 17]. Nowadays, people often share their personal experiences through microblogs (e.g., Twitter), blogs (e.g., Tumblr), social network sites (e.g., Facebook, Google+), media-sharing sites (e.g., Instagram), and review sites (e.g., Yelp, TripAdvisor) [12].

Not surprisingly, some research aimed to leverage this wealth of social media data to tackle the trip planning problem, from mining geotagged photos and check-ins on social networks [8, 4, 9, 6] to crowdsourcing recommendations by using the Amazon Mechanical Turk and other related systems [20, 14], combining human contributions to collectively build better itineraries.

In this paper, we present Aurigo (Figure 1), a novel system for tour planning that aims to strike a balance between automated approaches and purely manual approaches. Aurigo combines data-driven recommendations, interactive visualization, and personalization features that are highly configurable to help the user quickly build desirable itineraries.

Aurigo's recommendation engine suggests itineraries based on the user's personal preference (e.g., prefer museums) and data such as reviews and ratings extracted from review websites. Using Aurigo, the user can fine-tune a recommended itinerary's route using an interactive user interface (UI). The user may also build his or her own itinerary incrementally and interactively through the same UI. We designed Aurigo to tackle two common types of tour planning:

- **The efficient itinerary problem**: *How to efficiently go from point A to point B, visiting points of interest (POIs) along the way, while having the option to fine-tune and select which POIs to visit?*

- **The assisted exploration problem**: *How to rapidly create my own itinerary, assisted by an interface that relevantly displays POIs and information about my trip?*

Aurigo's major contributions include:

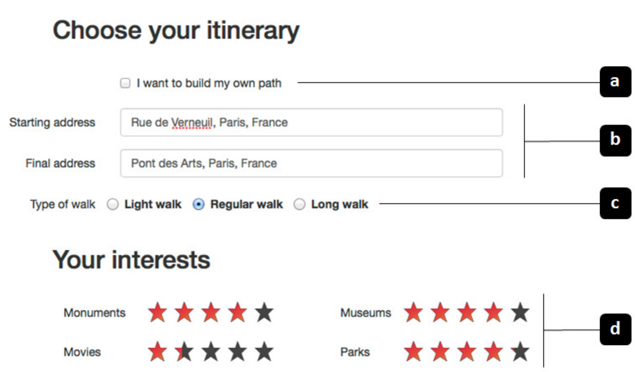

Figure 2. Aurigo's homepage with (a) the "I want to build my own path" option linking to the S2S (step-by-step) mode when checked, (b) the starting and ending address for the E2E (end-to-end) mode, (c) the style of walk indicating the qualitative length of the tour, and (d) the user's preferences towards the four types of attractions offered.

- Addressing two fundamental issues of trip planning with: an *end-to-end* mode (E2E) where Aurigo constructs a recommended itinerary based on a start and end points provided by the user; and an *step-by-step* mode (S2S) that allows the user to build a personalized itinerary interactively and incrementally, step by step.

- An original itinerary recommendation algorithm (used in E2E mode) that incorporates the user's *quantitative* preferences (e.g., desired tour length, preferred types of POIs, start and end points).

- An interactive user interface for building and fine-tuning the itinerary path based on the user's personal *qualitative* tastes and desires, with novel features such as the *Pop Radius* overlay (see Figure 1) which is a lightweight interaction technique for the user to interactively explore and select POIs within a short walking distance from a point on the map.

Currently, Aurigo uses the Yelp API to obtain data about POIs, the Google Maps API to localize them, and the Wikipedia API to obtain photos and descriptions about POIs. However, our general approach is database-agnostic and could be generalized to work with other databases POIs, reviews and ratings.

INTRODUCING AURIGO

Aurigo's main interface

Aurigo's user interface is composed of two main pages: a *homepage* (Figure 2), and an *exploration page* (Figure 3). The *homepage* allows the user to choose between two modes:

- *End-to-end* (E2E), where the user provides the start and end points and Aurigo's recommendation algorithm generates an itinerary based on the user's preferences.

- *Step-by-step* mode (S2S), where the user interactively build an itinerary by adding POIs incrementally, beginning from a start point.

In the E2E mode, two fields are available by default under "Choose your itinerary" to specify the starting address and the destination address (Figure 2, at b) and to indicate personal preferences through the style of walk: light, regular or long (at c). Under "Your interests", the user can specify his or her preferences for the four types of POIs that Aurigo supports (Monuments, Museums, Movie filming locations, and Parks) through a 5-star rating scale (at d). Alternatively, the user can enter the step-by-step mode by checking the box "I want to build my own path" (at a), where the user would only need to enter the starting address before moving on to the next page (i.e., the exploration page).

The *exploration page* (Figure 3) shows a map of the city, centered at the address the user has provided. The user can zoom in and out of the map. The current total distance of the itinerary is displayed at the top of the page ("). The map is the main interactive area through which the user would construct his or her itinerary (at e). It contains two collapsible panels: the *description panel* on the left (at d), and the *route panel* on the right (at f).

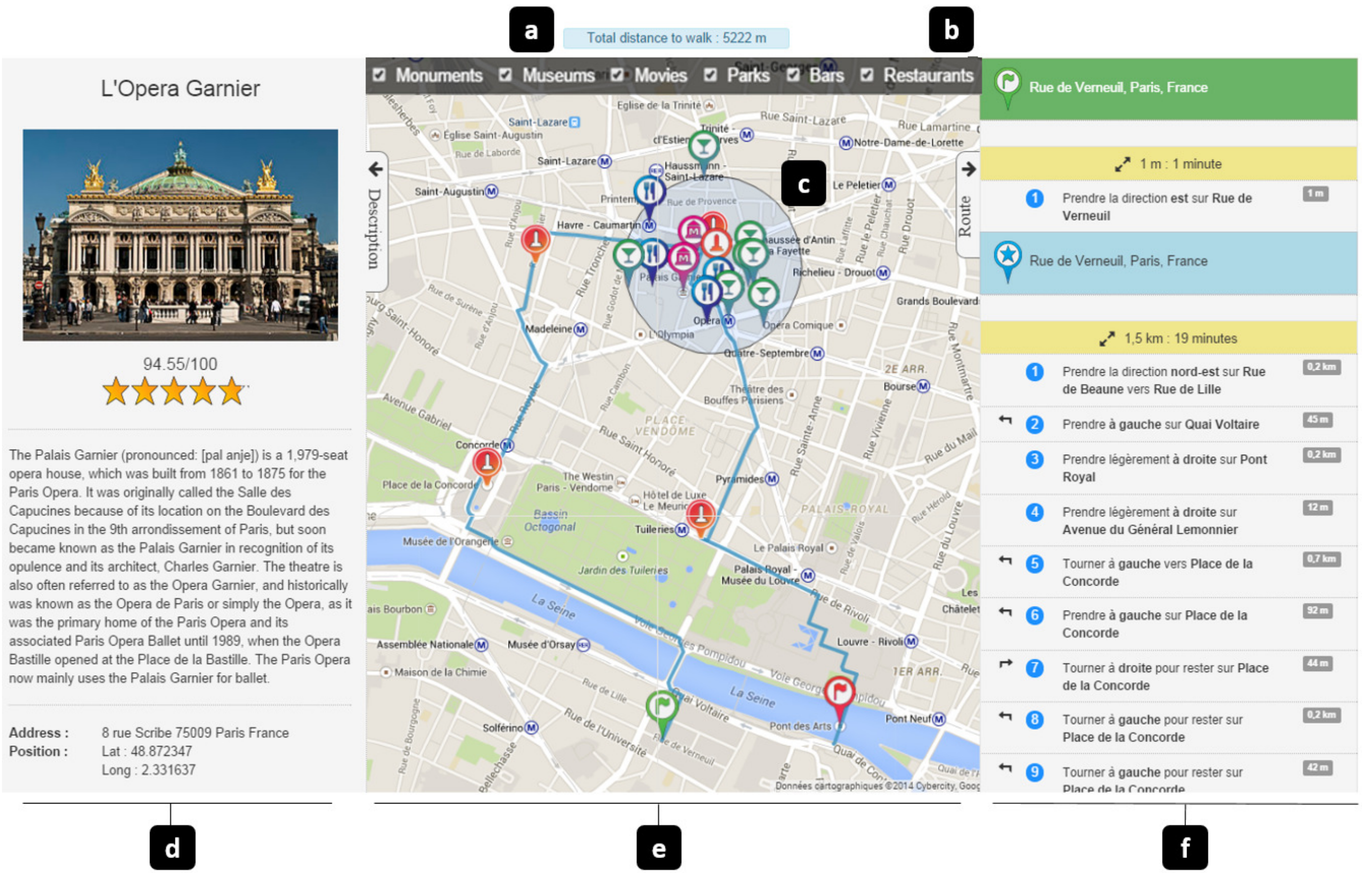

Figure 3. Aurigo's exploration page, with a tour constructed with the end-to-end mode (E2E, where the user provides the start and end points), composed of (a) the total distance, (b) the filter bar, (c) the Pop Radius, (d) the Description panel, (e) the map and (f) the Route panel.

A top filter bar (at b), right below the total distance, allows the user to select the types of POIs to display on the map. We included *bars* and *restaurants*, in addition to the four POI types listed on the homepage, since the user may want to find them on an ad hoc or on-demand basis.

POIs are displayed on the map as colored icons (see Table 1, and Figure 3, at c). The route panel (at f) shows the directions of the itinerary and is collapsible. The description panel (at d) provides a small paragraph about the selected POI with its photo (extracted from Wikipedia), its popularity score on a 0-to-100 scale, its address and its geographical location with latitude and longitude. We describe how all this information is extracted or derived from online websites and databases in a later section.

The map area (Figure 3, at e) offers multiple interactive features for the user to explore the surroundings of the itinerary and to personalize it by adding and removing POIs. Mousing over a POI will show its name in a pop up label (Figure 4, at c). Clicking a POI will cause a *Pop Radius* to show up as a transparent blue disk, centered at the clicked POI (at a). Within the Pop Radius, Aurigo displays all POIs of types that are selected in the filter bar. The user can then add or remove POIs in the Pop Radius and the itinerary will automatically adapt to the modifications and update the Route in the right panel.

Bar	Monument	Filming Location
Museum	Park	Restaurant

Table 1. Icons for each POI on the map

The Pop Radius represents the core feature of the step-by-step mode: beginning at a starting location, through consecutive invocations of the Pop Radius, the user can incrementally build an itinerary by adding or removing POIs, while exploring attractions in the city. The route shown on the right automatically updates after each user interaction with the Pop Radius.

ILLUSTRATIVE SCENARIOS: AURIGO IN ACTION

Below, we describe two scenarios to illustrate each of the two modes that Aurigo supports in tour planning.

Scenario 1: Gina does not know anything about Paris

Gina is visiting Paris for the first time. She prefers to walk around to better immerse herself in the city rather than joining a bus tour. She only knows that she will start her walk near her hotel and finish it near a cafe where she will meet her friend Kat who lives in Paris. She will use the end-to-end mode (E2E). While Gina enjoys walking, we does not

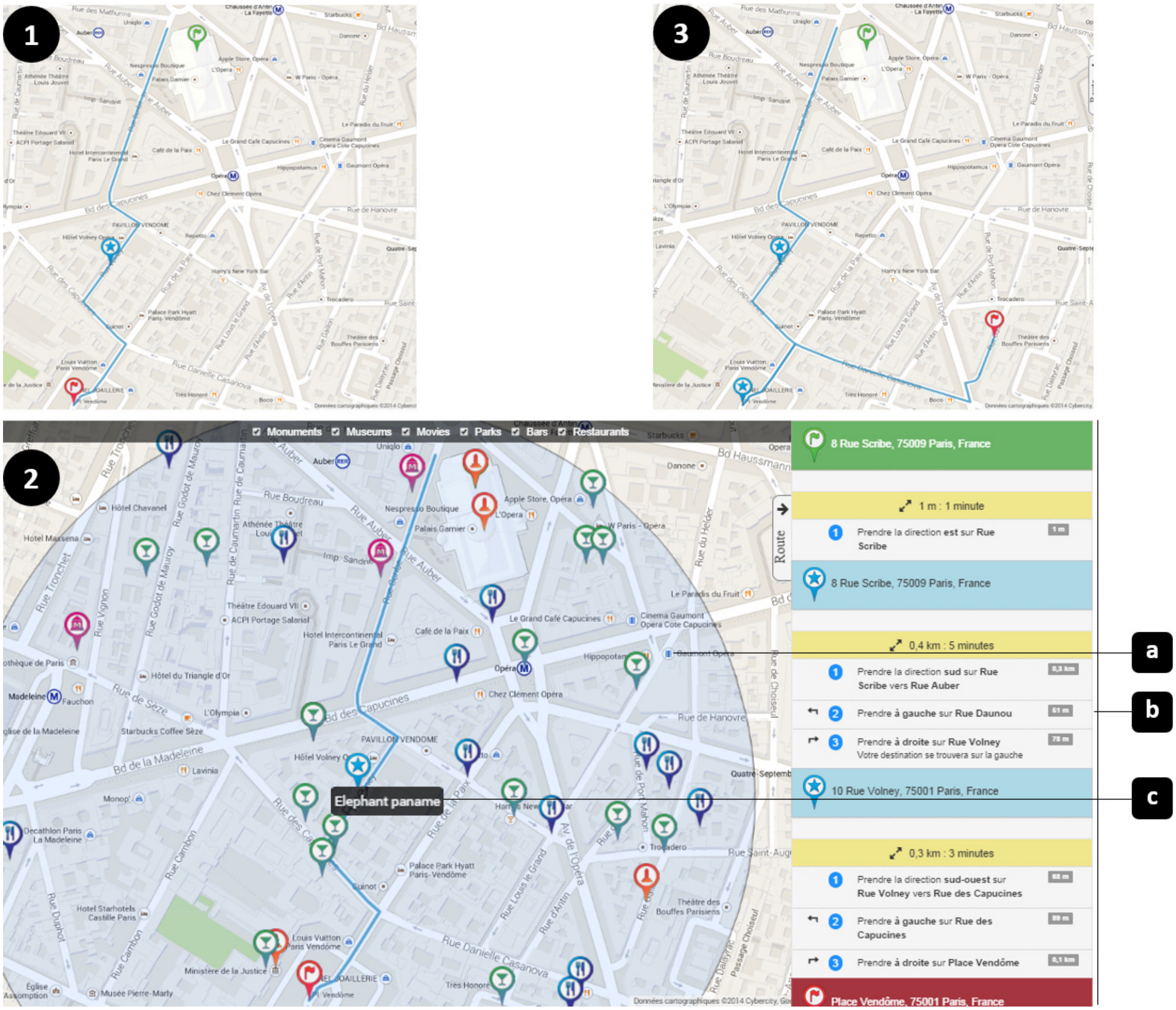

Figure 4. Illustration of the S2S mode (step-by-step) and the use of the Pop Radius: (1) the user has selected a starting point, shown by a green flag, and added two POIs. The last POI added is a temporary end point, shown as a red flag. (2) The user clicks on the POI indicated by a blue star to display a Pop Radius shown as a transparent blue circle (at a) that reveals all POIs within its radius; the route panel on the right (at b) shows directions for the current itinerary; mousing over a POI shows its names (at c). (3) The user has selected a new POI which becomes the new end point and the route is update automatically.

want to overdo it, so she chooses a *regular* walk on Aurigo's homepage (Figure 2).

Gina prefers to spend her day outdoors to visit historic sites and sees monuments rather than staying inside museums and buildings. The problem is that there are just too many monuments in the city. To indicate what she prefers, Gina rates each POI type (i.e., museums, parks, monuments, movie filming locations) using a 5-star scale on the homepage. Then she presses "Go" and Aurigo brings her to the exploration page.

Now she sees the map of the city on which Aurigo has displayed the recommended itinerary in blue, with all the selected POIs. The start and end points are indicated with a green and a red flag respectively. The path has been drawn using Google Maps API and each POI has a type-specific icon. Now Gina can use Aurigo's interactive visualization features to personalize this itinerary further.

On the recommended itinerary, Gina sees the Louvre museum as a suggested POI (Figure 5). She does not want to visit it yet since she promised to go there with Kat, so she clicks on the POI to remove it from the path. (If she wants to undo that, she can double click on it to add it back.) Instantly, the path is updated by the Google Maps API, which now excludes the Louvre museum from her itinerary.

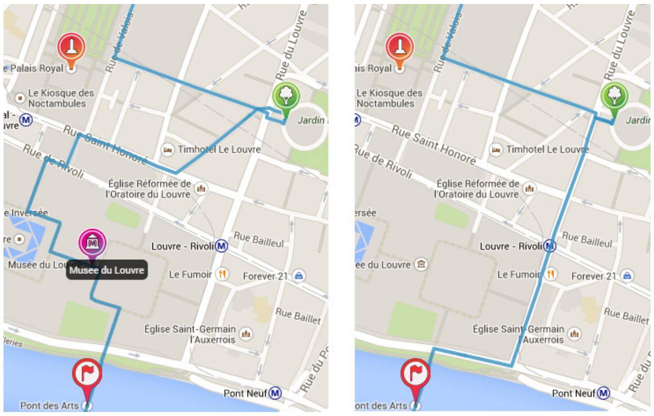

Figure 5. Gina does not want to go to the Louvre museum, but it is part of her recommended itinerary. She double clicks on it to delete it.

Gina can iterate this process until she is satisfied with her itinerary. Starting with an itinerary close to her expectations, it takes very little work for her revise the itinerary. She can easily revise it by integrating her subjective preferences. With Aurigo, Gina can craft her itinerary on a single-page user interface, instead of browsing many different sources of information, which can be difficult to navigate and summarize.

Scenario 2: Patrick knows a little bit about Paris and wants to create his own tour

Patrick has been reading a lot about Paris, and he knows the in which areas that he wants to spend his weekend with his wife. However, he is not sure how to organize his tour and is also not sure if he has enough places to visit for his two-day visit. Since he wants to begin their trip near their hotel, he enters the hotel's starting address on Aurigo's homepage, and starts building his path. Patrick selects "I want to build my own path" and enters "L'hotel Paris" as the starting point.

Figure 6. The Pop Radius is centered on the POI that has been clicked on. All the POIs inside this area are shown with an icon according to their category (or type of attraction). The starting point has a green flag. By double clicking on a POI, it adds it to the POI list and makes it the temporary ending point.

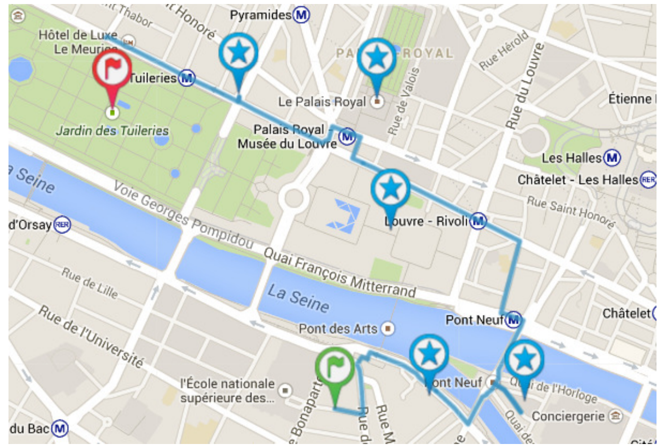

Figure 7. Candidate itinerary for Patrick's tour. The green flag is the starting point, the red flag is the ending point, and the blue stars are the intermediate POIs that Patrick has selected.

Once he clicks the "Go" button, a map is displayed on the exploration page. Thanks to the Pop Radius that has appeared at his starting location (Figure 6), he can choose his first place to visit. Since Patrick spent his previous day visiting monuments, he is eager to try something new. We wants to visit some movie filming locations. Within the Pop Radius, he looks for the POIs with the movie icons, and makes his choice. Patrick chose "Le Pont Neuf" because it would please his wife (he forgot their last wedding anniversary).

As he selected "Le Pont Neuf", it becomes the center of a new Pop Radius, which allows Patrick to select a new checkpoint. He wants to mix things up a little; he wants to visit the Louvre museum. Right after adding the museum to his itinerary, the new Pop Radius centered at the museum shows a restaurant which he recalls is a very nice one, with a great romantic setting, which his wife would surely enjoy dining in.

After a few minutes and some descriptions reading, Patrick has built a perfect itinerary step by step. Patrick takes note of the route, and gets ready for a great trip in Paris. He is fully prepared this time!

AURIGO'S KEY FEATURES

Popularity Function

Our popularity function combines the ratings and the number of reviews for each POIs. We used two different sources: (1) Yelp for museums, monuments and parks; (2) a movie database website for the filming locations (see *Data sources and implementation*). Movies are rated out of five points so we scaled them to a 0-to-100 scale. Concerning the Yelp POIs', we used a weighted scoring approach to balance the two quantitative parameters we extracted (i.e., ratings and number of reviews):

- 80 points derived from the star rating (1 to 5);
- 20 points derived from the review count (normalized).

The popularity formula for a POI p is the following:

$$popularity(p) = (stars(p) - 1) \times 20 + \ldots$$
$$\ldots + [\frac{nb_{reviews}(p)}{max_{reviews}(cat(p))}] \times 20$$

Where:

- $stars(p)$ is the number of stars (1 to 5) for the POI p (i.e., ratings);
- $nb_{reviews}(p)$ is the number of reviews for this POI;
- $max_{reviews}(cat(p))$ is the maximum number of reviews for this number of stars over all the POIs sharing the same category as p (for normalization).

In addition, a POI p with $stars(p) = 0$ (i.e., no ratings or null ratings) will return a zero popularity score. The formula above ensures that there is no score overlap between places with different ratings due to the number of reviews: 1 star will return a score in the interval $(0, 20]$, 2 stars $(20, 40]$, 3 stars $(40, 60]$, 4 stars $(60, 80]$ and 5 stars $(80, 100]$.

The Recommendation Algorithm

For the end-to-end mode (E2E), we designed a recommendation algorithm that constructs an itinerary path that goes through a sequence of POIs that are selected based on the users preferences. The algorithm first identifies and orders the candidate set of POIs, and then uses the Google Maps API to find a route to connect these POIs.

Our algorithm is fast and scalable. Its runtime is linear in the number of POIs specified. Empirically, it takes less than 0.5 second to run on more than 2,000 POIs. In the algorithm, the most computationally expensive step is the creation of the path itself (database accesses are instantaneous). The algorithm takes into account the following parameters from the user:

- Start and end points (both are addresses)
- Style of walk: *light walk*, *regular walk*, or *long walk*
- POI Preferences: expressed as 5-star ratings for *Monuments*, *Museums*, *Movies* (filming locations), and *Parks*
- POIs' popularity, computed with our popularity function

Other parameters concerning POI data include: POIs' geographical locations and popularity. We designed a scoring function to compute the popularity for each POI, which takes into account the reviews and ratings that the POI is associated with.

$$Score(p, path, taste, walkFactor) = distToPath(p, path)$$
$$-\lambda \times popularity(p) \times taste(cat(p)) \times walkFactor$$

Where:

- $distToPath(p, path)$ is the shortest distance of a point to the path (i.e., minimum of the distances from the orthogonal projections to each segment of the path using euclidian norm in the latitude-longitude plan);
- λ is empirical coefficient to balance popularity versus distance in the score function. We selected $\lambda = 0.0002$ after iterative testings;
- $popularity(p)$ is the score of the place given out of 100 as explained above; $taste(cat(p))$ is the the number of stars given to the corresponding category (higher means more preferred by the user);
- $walkFactor$ is a distance criterion, it allows us to tune the importance of the popularity of the place with respect of the distance. Long walks have higher values than short walks. By doing so, a path can include new points that are further away. The associated coefficients are 1 for a short walk, 2 for a medium walk and 3 for a long walk.

Since we do not expect reviews and ratings to change frequently, Aurigo pre-computes the popularity scores of all POIs. These popularity scores are then normalized using a scale of 0 to 100, with 100 before the score for the most popular place in the POI database.

Our algorithm is designed to find an itinerary in the form of a **natural** path connecting a list of POIs. For example, we would not want the user to walk through the same place twice, and we also would not want to an itinerary to consist of lengthy detours from the shortest path between the start and end points.

Given these design rationale, the trajectory obtained by linking the selected POIs from a birds-eye view should ideally be smooth. The first step of the algorithm (see Algorithm 1) is to draw a line between the start and end points. As we add POIs, we break the path down into Piecewise Linear (PwL) segments. From the initial straight line, the algorithm iteratively computes the Euclidean distance of all the POIs relative to this trajectory, selects the POI that minimizes the distance and maximizes a custom weighted objective function (with the parameters described above). The algorithm loops through the above steps (Algorithm 1, line 4-8), computing distances and picking a new POI in each iteration, until the path has 8 POIs, a limit imposed by the Google Maps API that the algorithm uses to get the turn-by-turn directions between consecutive POIs. We plan to investigate methods to overcome this limitation in our future work (e.g., break the itinerary into sub-itineraries).

The Pop Radius

The **Pop Radius** is one of Aurigo's key interaction techniques for locally exploring POIs. It allows the user to select any point along a path to display all POIs within a context-dependant radius, visually shown as a circular translucent blue overlay. It also allows the user to add and remove POIs from the tour itinerary, based on subjective preferences.

The area of the Pop Radius is computed based on the style of walk selected, using the following formula.

$$r = d(2i + 1)$$

Where r is the radius in meters, d a scaling factor (constant) set to 60m, and i depends on the walk: $i = 1$ for light, 2 for standard, and 3 for a long walk. If there are fewer than 5 POIs within the circle of radius r, it will be expanded with a new radius r' as follows:

$$r' = r + 2d$$

Clicking a POI within the Pop Radius selects it and displays its description on the Description panel. If the user clicks on it once more, it will be added to the path. Similarly, the user can remove a POI that is already on the path by clicking on it. The tour itinerary will be instantly updated.

DATA SOURCES AND IMPLEMENTATION

The current Aurigo prototype focuses on the city of Paris. We extracted a total 2204 POIs from various online websites and databases: from Yelp, we extracted 626 bars, 236 monuments, 187 museums, 150 parks, 714 restaurants; and from an open-source website[1], we extracted 282 filming locations. For each POI, Aurigo extracted its name, address, number of reviews, and its ratings using the Yelp API. For movie filming locations, we derived their popularity scores from ratings of the movies filmed at those sites by extracting data from *the movie database* website[2]. We used the Google Maps API to find the geolocation of each POI, and Wikipedia for the POI's description and photos.

EVALUATION

Hypothesis

Aurigo aims to assist users during the creation of tour itineraries in cities where there is a high density of POIs. We conducted a user study to assess how users would take advantage of the various features provided by Aurigo to build their own itineraries, and to test the usability of Aurigo's features and the system overall. Our study was designed to compare Aurigo with currently available solutions and evaluate the impact of Aurigo's unique features on its performances. We hypothesized that Aurigo is easier to use and provides more personalized itineraries in similar time constrains. We selected Paris for this user study, since it is one of the most visited city in the world, and it has a wide variety of monuments, museums, parks and other featured POIs types to visit.

[1] http://opendata.paris.fr/

[2] https://www.themoviedb.org/

Algorithm 1: Building an itinerary (a path of POIs), with a *Score* function that determines how much a user may like it

Data: path, POIs, Score, Insert
Result: path
$p_{min} \longleftarrow POIs[0]$;
while $POIs.length < 8$ **do**
  $score_{min} \longleftarrow +\infty$;
  for $p \in POIs$ **do**
    $score = Score(p, path)$;
    if $score < score_{min}$ **then**
      $score_{min} \longleftarrow score$;
      $p_{min} \longleftarrow p$;
  $path = Insert(p_{min}, path)$;

Design

We used a within-subjects design with two conditions: (1) *Aurigo* and (2) *Google Maps* where participants used Google Maps to find interesting places and build their itineraries. This design allowed the participants to compare their experiences with both tools. We decided to only evaluate Aurigo's end-to-end mode (with recommendation algorithm) against Google Maps for two reasons: (1) Aurigo's step-by-step mode (S2S) does not have a counterpart in Google Maps; (2) Aurigo's E2E mode includes most interaction and algorithmic functionality used in the S2S mode.

Participants

We recruited 10 volunteers for this user study by word-of-mouth from our professional and academic networks. The first inclusion criterion for a user to participate is that he or she should have travelled at least once in his or her life. There were 4 males and 6 females, ranging from 18 to 27 years old, averaging at 23 years old. Most of them were graduate students, a few were consultants, and one was an undergraduate student. Every participant had used Google Maps before, from once a week and more than once a day. However, most participants had not used Google Maps for planning tours.

Every participant had at least travelled once in the past 3 years, domestically or internationally. Only one participant was not used to preparing his or her own trip. The other participants used tour guides or web searches to help them prepare (80%) (Figure 8). The participants commented that they usually needed about an hour to design a satisfying itinerary. Each session lasted for about 75 minutes, and each participant was paid $10 in compensation for his or her time.

Apparatus and Materials

All participants used laptops that we provided. Every session was conducted in a "Incognito" window of the Google Chrome web browser (i.e., private browsing mode) so as not to keep any history or preferences from a previous participant. When using Google Maps and Aurigo, all Google accounts were disconnected, so that all participants will be using the tools in their default settings. The database Aurigo used from Yelp and open source websites contained 2,204 POIs. Aurigo was run on a localhost (e.g., the laptop the participant was using).

Procedure

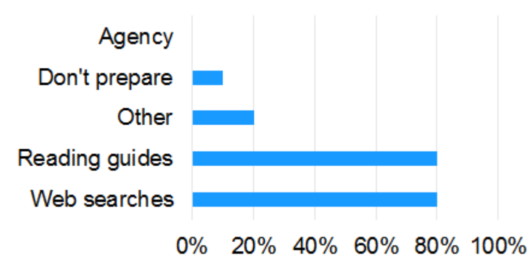

Figure 8. Types of tools used by our participants for tour planning. Tour guides and web searches are most frequently used. None of the participants consulted travel agency for tour preparation.

We first asked all the participants to fill out a background questionnaire to evaluate how much they were used to travelling and itinerary building. Then, we explained the main purpose of this study to the participants. Next, we asked the participants to imagine that they were tourists about to arrive in Paris, or that they were already in the city, but they do not know much about the city. With this scenario, they wanted to visit the city by walking and they would use the Internet to find interesting information about the attractions available. They would focus on where they wanted to start, and their preferences considering their walking style (e.g., short walk, moderate walk, long walk) but also the type of places they wanted to visit (e.g., museums, parks, monuments, famous movies shooting locations).

All participants were free to choose their own start and end points. In fact, if we were to tell all participants to do exactly the same things, user preferences would become irrelevant, and it would be unlikely for them could build satisfying itineraries. Before starting with each condition, we briefly explain how to use the tool by showing a demo. Finally, we randomly assign half of the participants to start with the Aurigo condition, to counter balance for condition order.

For each condition, participants were given about 15 minutes to find interesting places and design their itineraries. Then they were asked to fill out another questionnaire that asked for their subjective opinions about Aurigo, how it compared to Google Maps, and any feedback and suggestions that they might have about Aurigo.

Results

Quantitative results

For each condition, the participants used the tool for a maximum of 15 minutes. If the participant was satisfied with her itinerary, she could let us know and we would stop the timer. The number of places chosen for the itineraries was interesting to us because it showed us how complex the itineraries might be. This number has an upper bound however: Google Maps only allows up to 10 places for an itinerary, while Aurigo supports up to 8 (due to limitations of the Google Maps API which Aurigo uses). We recorded these numbers for each participant; if a participant gave up for a condition, the time was noted and the number of places set to 0 (Figure 9).

In our evaluation, we saw that participants had similar behaviors using Google Maps and Aurigo. We believed Google-Maps based Aurigo's interface might not be the only factor that contributes to this, but that we also believed it was due to participants quickly becoming familiar with Aurigo's usage. Before the study, we had the hypothesis that Aurigo participants would need to spend more time than Google Maps participants. However, using a one-tailed test, we found that this hypothesis was not supported ($p > .1$). Despite the similarity in timing results, we observed an important difference between the Aurigo and the Google Maps conditions: whereas the Aurigo participants spent most of their time on designing itineraries by adding or removing points to adjust locally the itinerary, Google Maps participants spent most of their time on finding places, often leaving little time in the end on building their itineraries. This observation implies that the itineraries built by the Google Maps participants might not be satisfactory.

We also observed that the number of places chosen for Aurigo were really close to the maximum allowed. Indeed, every participant chose at least 7 places, and over half of them stopped at 8 places, whereas we only suggested 6 places at the beginning of each itinerary designed with Aurigo's algorithm. Before the study, we hypothesized that Aurigo participants would be able to find more places than Google Maps participants since Aurigo suggested multiple places to start 9. Using a one-tailed test, we find that this effect to be statistically significant ($p < .002$), proving our hypothesis. Nevertheless, we could not find any correlation between the time spent and the number of places chosen. Indeed, every participant has his or her own rhythm when it comes to exploring the map, reading descriptions and choosing his or her POIs. Moreover, a satisfactory itinerary does not necessarily require a lot of POIs.

Subjective results

We asked our participants to compare Aurigo with Google Maps (Figure 10), and to rate Aurigo on several criteria (Figure 11). Aurigo was rated favorably. All participants, except one, would choose Aurigo over Google Maps when asked which tool they would prefer to use in the future for tour planning. Most Aurigo participants seemed to enjoy using it.

POIs and paths seem to be more easily found using Aurigo (80% of participants agree), and Aurigo's user interface seems more enjoyable to use as well. Knowing that Aurigo's design is not only based on building an itinerary but

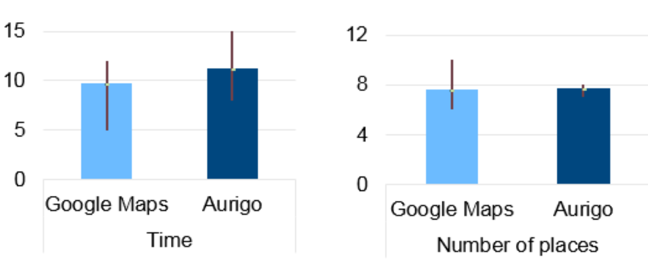

Figure 9. Time taken and number of places found on average for each condition with 95% error bar. Results were similar for Aurigo and Google Maps in term of the average number of places.

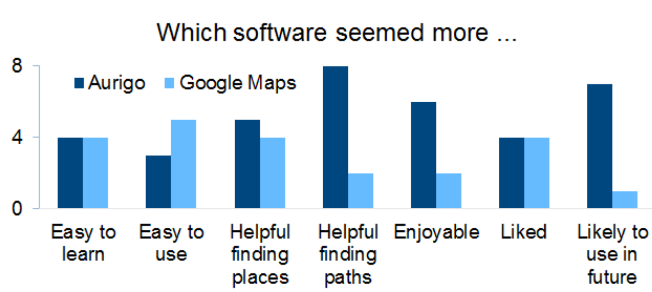

Figure 10. Impressions about the different conditions after testing them. Aurigo was rated favorably comparing to Google. Only one user chose Google Maps over Aurigo, when asked which tool they prefer to use in the future.

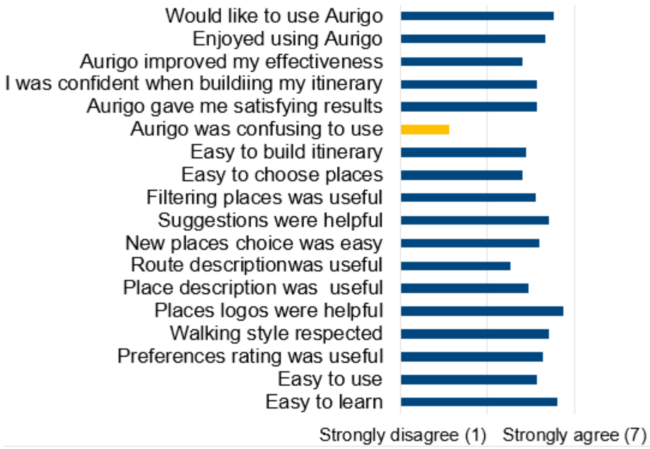

Figure 11. Subjective ratings of Aurigo. Participants rated Aurigo favorably. It didn't seem confusing, the best features for the participants being the places logos, the walking style and the suggestions.

also on finding interesting places to put in it, participants seem to have preferred it over Google Maps. One participant said "choosing my preferences and being able to modify my itinerary by picking some new places were the features I liked the most about Aurigo".

An exciting piece of feedback from the participant was that Aurigo did not seem to be confusing to use. Participants were able to learn to use its features in a very short amount of time. The top three features that interested participants were the icon design for the POIs, the choice of their walking style and the suggestions given by the "Pop Radius". These features have been developed to help users find POIs that best match their preferences, and they seem to play their part very well.

The overall impression about Aurigo seems to be very positive. However, some technical details like a wider "Pop Radius" has been identified as wishes by some of them and we plan to improve on them. The description content was also discussed. Some participants suggested that the POI descriptions should also include opening hours, and ticket prices.

RELATED WORK

Tour planning, a practical everyday activity and challenge, has received a lot of attention in research. Initially considered as an optimization problem under multiple constraints, a wide variety of studies have tried to take advantage of data and metadata available on social networks such as Flickr, Panoramio, Facebook or FourSquare.

While most approaches focus on displaying only points of interest (POIs) close to the user's location, INTRINGUE represents an early foray of intelligent information systems in tourism [2] by offering trip recommendations personalized for single individuals or groups with a user interface on Web browers and WAP minibrowers. Stimulated by those early prototypes, the tour planning problem has aroused a significant interest within the intelligent user interface community. Trip-Mine has introduced the travel time constraints and scalability of travel region to offer a broader tour planning experience [10]. The trip map, the set of potential POIs considered, was modeled by a weighted graph and Lu et al. proposed three optimization mechanisms for their cost function. Taking time constraints into account has been further developed with a study dealing with transfer connections in public transportation [7], taking into account the network togy, the schedules and the passengers' distribution over time to develop a trip planning system based on this algorithm for passengers. Tour planning problem has evolved to a Multi Constrained Team Orienteering Problem with Time Windows (MCTOPTW) by combining time with all other quantitative parameters [15]. Sylejmani et al. adopted the local search meta-heuristics of "tabu" planning, which keeps track of moves that cannot be performed to ensure high constraints on the neighborhood candidates to the trip schedule and facilitate the diversification of search process into unexplored part of the map. Aurigo's planning approach seems to be the next step of these evolutions, taking quantitative parameters (users' preferences, POIs' score) but also time restriction (type of walk desired by the users) into account to propose the best itinerary within a known map.

But those planning algorithms are NP hard [3], hence justifying the need for mining human behaviors to recommend natural trips more than mathematically optimal ones. Geotagged photos have represented a valuable and reliable source of metadata to track millions of tourists on their itineraries. However, these metadata appeared to be noisy and needed efficient filtering methods [13]. Efficiently collecting those metadata allows to leverage temporal information regarding tourist attractions. By mining trips on these large scale geotagged photos databases, it was possible to segment photo collections into trip patterns [1], and label them using tags under themes such as landmarks, events, nature or food tasting. However, to fully take advantage of those metadata (e.g., location, time, tags), Chareyron et al. developed methodologies to rebuild a photographer's path from Flickr (a hosting website allowing people to tags and share their photos and videos) [5], hence recreating the photographer's spatial and temporal trace. Aurigo also uses raw data in order to build itineraries and propose the best one with the close difference that the building part of the itineraries is achieved by the algorithm. The itineraries stay as natural as possible with the consideration of the user's preferences but also the "Pop Radius" functionality allowing the user to tune his or her itinerary. Using Panoramio, a geolocation-oriented photo sharing website, another initiative called Photo2Trip [11, 19], indexed paths generated from photos' metadata and computed a distance to evaluate the similarity between two paths. Then, using path enrichment technique, it suggested to the user itineraries resulting from the merge of several paths to ensure a longer and denser tour. Therefore, this approach offers tours that are more representative of people behavior.

Studies have been combining these behavioral trip mining with recommendation algorithms to provide knowledge-based tour planning tools. Among them, Antourage [8], proposed a distance-constrained recommendation algorithm relying on the mining of large volumes geotagged photos databases like Flickr and Panoramio. Given a starting point, which is usually the user's hotel, the algorithm applied max-min ant system (MMAS) derived from classic "ant colony optimization" (ACO) meta-heuristic to suggest tours looping back to the starting point. Aurigo presents another approach where our recommendation algorithm selects an ordered list of POIs and Google API trace the route between them. Another framework, TripBuilder [4], mined itineraries from Flickr and matched the POIs available on Wikipedia. Then it applied classic generalized maximum coverage developing its own cost function for path recommendation.

Other studies have exploited users' "check-in" activities on Location-Based Social Networks (LBSN) such as Gowalla, FourSquare or Facebook. For example, Lu et al. [9] used data from these 3 social networks non-distinctively to attribute popularity and temporal-based properties to points of interest (POIs). Then they combined a recommendation algorithm based on parallel computing with these data to extract social relationships and temporal properties. In addition to using these "social breadcrumbs" [6], De Choudhury et al. proposed to validate the itineraries with Amazon Mechanical Turk (AMT) for better recommendations.

The use of AMT was fully exploited in Mobi [20], a crowd-powered trip planning application using natural language input and quantitative constraints to ask the crowd for asynchronous contributions. With Crowdcierge [14], Rafidi et al. suggested the application of the retainer model to keep a pool of workers and therefore offer synchronous AMT where workers tag ideas to plan the itinerary and re-plan in response to problems that can arise during the trip. Aurigo uses large pools of workers with the scores being extracted from popular touristic social networks, that let the data respect to the extent possible the tourists' opinion of every places.

DISCUSSION, LIMITATIONS AND FUTURE WORK

We believe that interactive visualization is complementary to planning algorithms for providing efficient and personalized tour itineraries. The interface we built provides features to empower the user: (i) information and visualization to guide his or her choices and (ii) means of action to locally modify his or her itinerary. Thanks to those elements, only a small amount of quantitative preferences is required from the user in the departure-arrival mode. Our approach merges smoothly the crowd-sourcing by taking into account popularity in POIs' scores without depriving completely the user from making his or her own choices.

Due to the limited amount of participants in the user study, it has been difficult to establish significant results. However, Aurigo appears to take significantly more advantage of the maximum number of places possible to visit. Users also spent significantly more time on the Aurigo condition, mainly due to the interactive path they liked to tuned as much as much. This feature and Aurigo's interactive interface including the Pop Radius function were overall appreciated by the majority of participants. A User study at a larger scale would help us evaluate further the quantitative performances.

Aurigo currently relies on POIs' ratings, reviews and location, and displays pictures, descriptions and ratings. In the future we would like to enable our system to use more data and allow users to visualize more information on the interface. Aurigo does not take yet into account the time of departure nor provides a time estimation in addition to the total distance outside of the Route in the right panel. We could integrate the opening timetables of museums, restaurants and other time dependant types of POIs to recommend our path. With appropriate data, we could also integrate prices as a user's parameter and estimate the expends of the tour. Aurigo does not integrate either a public transportation module that would allow a user to assemble interactively several tour itineraries by taking recommended route by train or bus within the same city.

In terms of interface, we consider to improve the Pop Radius in the future by letting the user customize in some extends the size of the disk, along with allowing the creation of a Pop Radius anywhere on the map. We have also been working on creating a heat map that would be superimposed on the plan indicating the density of parameters such as popularity by type. This heat map could also be the first brick of a system that not only suggests POIs for a tour itinerary but also adapts the path to walk in streets that are more likely to interest the user between each POI (e.g., famous avenue instead of small street even if it increases the distance walked).

However those new features strongly depends on the provided data sources. We do not aim to create our own user platform but we want to demonstrate the potential of Aurigo in this proof of efficacy. Aurigo is an interactive system fed by data that can be integrated by a review website or a social network that references POIs, and uses reviews and ratings.

CONCLUSIONS

This paper introduces Aurigo, a novel system for tour planning. Aurigo helps the user create personalized itineraries through a mixed-initiative approach, by combining data-driven recommendations, interactive visualization, and customizable features. We illustrated the utility of Aurigo through two scenarios: the *end-to-end* mode (E2E) where Aurigo constructs an itinerary based on a start and end points provided by the user; and the *step-by-step* mode (S2S) that allows the user to build an itinerary interactively and incrementally.

We performed a user study with 10 participants, where most users found Aurigo and its features, such as the algorithmically-suggested POIs, and the Pop Radius, highly enjoyable and easy to use. In the current prototype, Aurigo uses data from Yelp and Wikipedia, and the Google Maps APIs. We look forward to integrating Aurigo with other crowd-sourced POI databases and deploying Aurigo for more people to try it out.

REFERENCES

1. Arase, Y., Xie, X., Hara, T., and Nishio, S. *Mining people's trips from large scale geo-tagged photos*. ACM, New York, New York, USA, Oct. 2010.
2. Ardissono, L., Goy, A., Petrone, G., Segnan, M., and Torasso, P. Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices. *dx.doi.org 17*, 8-9 (Nov. 2010), 687–714.

3. Bao, J., Yang, X., Wang, B., and Wang, J. An Efficient Trip Planning Algorithm under Constraints. In *Web Information System and Application Conference (WISA), 2013 10th*, IEEE (2013), 429–434.
4. Brilhante, I., Macedo, J. A., Nardini, F. M., Perego, R., and Renso, C. Where shall we go today?: planning touristic tours with tripbuilder. In *CIKM '13: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, ACM Request Permissions (New York, New York, USA, Oct. 2013), 757–762.
5. Chareyron, G., Da-Rugna, J., and Branchet, B. Mining tourist routes using Flickr traces. In *ASONAM '13: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ACM (New York, New York, USA, Aug. 2013), 1488–1489.
6. De Choudhury, M., Feldman, M., Amer-Yahia, S., Golbandi, N., Lempel, R., and Yu, C. *Automatic construction of travel itineraries using social breadcrumbs*. ACM, New York, New York, USA, June 2010.
7. Guo, J., Jia, L., Xu, J., and Qin, Y. An Algorithm for Trip Planning with Constraint of Transfer Connection in Urban Mass Transit Network. *Distributed Computing and Applications to Business, Engineering & Science (DCABES), 2012 11th International Symposium on* (2012), 341–344.
8. Jain, S., Seufert, S., and Bedathur, S. *Antourage: mining distance-constrained trips from flickr*. mining distance-constrained trips from flickr. ACM, New York, New York, USA, Apr. 2010.
9. Lu, E. H.-C., Chen, C.-Y., and Tseng, V. S. *Personalized trip recommendation with multiple constraints by mining user check-in behaviors*. ACM, New York, New York, USA, Nov. 2012.
10. Lu, E. H. C., Lin, C.-Y., and Tseng, V. S. Trip-Mine: An Efficient Trip Planning Approach with Travel Time Constraints. In *Mobile Data Management (MDM), 2011 12th IEEE International Conference on*, IEEE (2011), 152–161.
11. Lu, X., Wang, C., Yang, J.-M., Pang, Y., and Zhang, L. *Photo2Trip: generating travel routes from geo-tagged photos for trip planning*. generating travel routes from geo-tagged photos for trip planning. ACM, New York, New York, USA, Oct. 2010.
12. Munar, A. M., and Jacobsen, J. K. S. Motivations for sharing tourism experiences through social media. *Tourism management 43* (Aug. 2014), 46–54.
13. Popescu, A., and Grefenstette, G. Deducing trip related information from flickr. *Proceedings of the 18th international conference on World wide web* (2009).
14. Rafidi, J. *Real-time trip planning with the crowd*. ACM, New York, New York, USA, Apr. 2013.
15. Sylejmani, K., Dorn, J., and Musliu, N. A Tabu Search approach for Multi Constrained Team Orienteering Problem and its application in touristic trip planning. In *Hybrid Intelligent Systems (HIS), 2012 12th International Conference on*, IEEE (2012), 300–305.
16. Tung, V. W. S., and Ritchie, J. R. B. Exploring the essence of memorable tourism experiences. *Annals of Tourism Research 38*, 4 (Oct. 2011), 1367–1386.
17. Xiang, Z., and Gretzel, U. Role of social media in online travel information search. *Tourism management 31*, 2 (Apr. 2010), 179–188.
18. Xiang, Z., Wang, D., O'Leary, J. T., and Fesenmaier, D. R. Adapting to the Internet: Trends in Travelers' Use of the Web for Trip Planning. *Journal of Travel Research* (Feb. 2014), 0047287514522883.
19. Yin, H., Wang, C., Yu, N., and Zhang, L. Trip Mining and Recommendation from Geo-tagged Photos. In *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, IEEE (2012), 540–545.
20. Zhang, H., Law, E., Miller, R., Gajos, K., Parkes, D., and Horvitz, E. *Human computation tasks with global constraints*. ACM, New York, New York, USA, May 2012.