Automatic Construction of Travel Itineraries using Social Breadcrumbs

Munmun De Choudhury*
Arizona State University
Tempe, AZ, USA
mdechoud@asu.edu

Moran Feldman*
Technion - Israel Inst. of Tech.
Haifa, Israel
moranfe@cs.technion.ac.il

Sihem Amer-Yahia
Yahoo! Research
New York, NY, USA
sihem@yahoo-inc.com

Nadav Golbandi
Yahoo! Research
Haifa, Israel
nadavg@yahoo-inc.com

Ronny Lempel
Yahoo! Research
Haifa, Israel
rlempel@yahoo-inc.com

Cong Yu
Yahoo! Research
New York, NY, USA
congyu@yahoo-inc.com

ABSTRACT

Vacation planning is one of the frequent—but nonetheless laborious—tasks that people engage themselves with online; requiring skilled interaction with a multitude of resources. This paper constructs intra-city travel itineraries automatically by tapping a latent source reflecting geo-temporal breadcrumbs left by millions of tourists. For example, the popular rich media sharing site, Flickr, allows photos to be stamped by the time of when they were taken and be mapped to Points Of Interests (POIs) by geographical (i.e. latitude-longitude) and semantic (e.g., tags) metadata.

Leveraging this information, we construct itineraries following a two-step approach. Given a city, we first extract photo streams of individual users. Each photo stream provides estimates on where the user was, how long he stayed at each place, and what was the transit time between places. In the second step, we aggregate all user photo streams into a POI graph. Itineraries are then automatically constructed from the graph based on the popularity of the POIs and subject to the user's time and destination constraints.

We evaluate our approach by constructing itineraries for several major cities and comparing them, through a "crowd-sourcing" marketplace (Amazon Mechanical Turk), against itineraries constructed from popular bus tours that are professionally generated. Our extensive survey-based user studies over about 450 workers on AMT indicate that high quality itineraries can be automatically constructed from Flickr data.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-Data mining; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

*Part of this research was performed while visiting Yahoo! Research.

General Terms

Algorithms, Experimentation

Keywords

Flickr, geo-tags, mechanical turk, orienteering problem, social media, travel itinerary.

1. INTRODUCTION

Travel itinerary planning is often a difficult and time consuming task for a traveler visiting a destination for the first time. It involves substantial research to identify points of interests (POIs) worth visiting, the time worth spending at each point, and the time it will take to get from one place to another. Without any prior knowledge, one must either rely on (1) travel books, (2) personal travel blogs, or (3) a combination of online resources and services such as travel guides, map services, public transportation sites, and human intelligence to piece together an itinerary.

All these options have shortcomings. Travel books do not cover all cities/locations and, perhaps more importantly, are not free. Personal travel blogs reflect a single person's view, with no guarantees provided over the writer's experience or the amount of preparation invested in planning the trip. Finally, compiling an itinerary by selecting individual POIs and researching their to's and fro's is a task which is both time consuming and requires significant search expertise.

Fortunately, with the advancement of digital photography and the rapid rise of rich media sharing sites such as Flickr (http://www.flickr.com/), millions of travelers are now sharing their travel experiences through rich media data such as photos. More interestingly, users are increasingly associating shared media with rich contextual information. Flickr photos, for example, are usually time-stamped by the date and time of when they were taken. Furthermore, they are often tagged with geographical information (i.e., latitudes and longitudes), which can be easily mapped to the POIs. Even more frequently, the photos are associated with textual metadata such as tags, titles, notes, and descriptions.

Such shared photos can be seen as billions of geo-temporal breadcrumbs that can promisingly serve as a latent source reflecting the trips of millions of users. Our goal is therefore to *automatically* construct travel itineraries at a large scale from those breadcrumbs. More specifically, by analyzing these breadcrumbs associated with a person's photo stream, one can deduce the cities visited by a person, which

POIs that person took photos at, how long that person spent at each POI, and what the transit time was between POIs visited in succession. By aggregating such *timed paths* of many users, one can construct itineraries that reflect the "wisdom" of touring crowds. Each such itinerary is comprised of a sequence of POIs, with recommended *visit times* and approximate *transit times* between them.

The tasks described above raise several key challenges that are tag-related, geo-related, or time-related. Tags are used to capture different user intents. For example, a photo of a person named Paris taken in NYC may be tagged by the person's name. City and POI names have different variants. For example, "NYC", "Manhattan" and "The Big Apple" all relate to NYC. Geo-location information can be misleading. For example, pictures of a landmark can be taken from afar such as a picture of the Brooklyn Bridge taken from atop the Empire State Building in NYC. In this case the latitude and longitude information may not match the pictured landmark. As for time-related challenges, some travelers try to maximize the number of POIs they visit, while others like to leisurely tour fewer POIs. Backpackers move between places faster than a family of four can. In summary, the association of photos to cities/POIs needs to be assessed carefully, and the construction of travel itineraries from photos must address all the challenges raised above. In addressing these challenges and others, we make the following contributions:

1. We introduce a novel end-to-end approach that starts with the analysis of latent information reflected in social media sharing sites, and ends with the synthesis of practical information in the form of travel itineraries.
2. As an initial implementation of our approach, we apply a pipeline of multiple heuristics that together extract reliable granular evidence of individual tourists' trips to a destination from Flickr photos.
3. We aggregate the individual trips to form a graph representing collective touristic behavior, and adapt a solution of the Orienteering problem to efficiently generate intra-city travel itineraries from the graph.

An extensive survey based user study eliciting feedback from 450 users on Amazon's Mechanical Turk platform validated our system's ability to generate high quality travel itineraries for popular touristic cities.

The rest of the paper is organized as follows: Section 2 surveys related work. Section 3 presents some basic terminology and details how we process Flickr photos to derive timed travel paths of many users. Then, in Section 4, we aggregate those paths and generate itineraries. Section 5 reports on our user study, conducted on Amazon Mechanical Turk. Section 6 presents future challenges and conclusions.

2. RELATED WORK

Our work integrates the two emerging fields of touristic data analysis and touristic information synthesis, and is therefore related to various works in these two fields. For the former, there are a number of studies on analyzing landmark (i.e., POI) visitation patterns from geo-spatial and temporal evidences left by travelers. However, those works generally avoid synthesizing or recommending new paths and instead focus solely on the analysis itself. We survey those works in Section 2.1. For the latter, a number of other works construct and recommend tourist itineraries at various granularities. They rely, however, on structured and cleansed data on landmarks and their attributes, and do not deal with the challenge of analyzing and extracting from noisy data. We survey those works in Section 2.2. Our work is tangentially related to several vast fields such as visualizing geo-spatial databases, tracking movements based on sensor networks, and constraint optimization. Due to space limitation, we choose not to survey them here.

2.1 Touristic Data Analysis

Many works mine geo-spatial and textual metadata associated with Flickr images. Rattenbury et al. [16] analyze the geo-temporal dynamics of Flickr tags and distinguish between tags describing places and events. Ahern et al. [1] plot aggregated textual metadata associated with georeferenced Flickr images on a map interface, thereby exposing how Flickr users at large describe landmarks. Crandall et al. [8] explore the association of Flickr photos to physical locations, and apply their techniques to extract landmarks at various granularity levels that correspond to a geo-spatial hierarchy. Popescu and Grefenstette [15] deduce visit times at landmarks based on timestamps of Flickr photos.

A large body of related work was done by Girardin et al. [11], who analyzed dynamics of people moving through urban spaces. In [12], the authors study digital footprints, explicit (e.g. Flickr photos) or implicit (e.g. cell association in a mobile communication network), that people "leave behind" while traveling through a city; whereas, the focus of [13] was to tap tourist dynamics for better urban planning and deployment of location-based services.

Note, apart from tags and geo-information, the visual features of photos can also be analyzed to reveal clues about its content [16, 8]. However the visual features alone do not lend us any temporal information. Hence we chose to rely on solely the textual tag and geo-information.

2.2 Touristic Information Synthesis

An early work [7] developed a "tourist guide" system that used mobile computing technology on wireless infrastructure to present tourists with tour-related information that could be tuned to fit multitude of circumstantial contexts. In two more recent works, Leake and Powell tackle itinerary planning with Case Adaptation methodology within the framework of the Case Based Reasoning (CBR) by building the WebAdapt system [14]. WebAdapt taps knowledge bases of formalized knowledge, such as Wikipedia, and a geographical gazetteer in helping users to modify and personalize existing itineraries.

Dunstall et al. [9] developed ETP (Electronic Travel Planner), a system that constructs an entire vacation by piecing together structured components of types "tour", "lodging", and "transportation". Tours typically contain places to visit and activities to perform within a single day and general area, i.e. between lodgings and transports. The INTRIGUE system [2] recommends sightseeing destinations and itineraries while taking into account preferences of individuals or members of a group.

2.3 Integrating Analysis and Synthesis

Tai et al. [17] is the only prior work that we have found to be addressing both the mining of itinerary data and its synthesis. Built as an itinerary recommender system for Flickr users, they treat scenic landmarks photographed by a user as defining that user's interests. Given the last few landmarks photographed by a user, the system recommends a sequence of landmarks the user hasn't photographed (nor, presumably, visited) yet based on sequences of landmarks visited by other users. However, the system does not address the challenges involved in constructing an itinerary from scratch, nor does it address the temporal dimension (visit and transit times) of the proposed itinerary.

In summary, there is a lack of integration between touristic data mining *and* synthesizing those mining results to form itineraries such that the users can easily adopt and leverage. Our work aims to fill this void by proposing the first end-to-end system for automatically constructing full itineraries from the analysis of geo-temporal data available in large scale rich media sharing sites.

3. CONSTRUCTING TIMED PATHS

We begin by introducing a few basic notations. We have a set of photos $\mathcal{P}$, their owners $\mathcal{U}$, and a set of cities $\mathcal{C}$, each with a set of POIs $\mathcal{L}_C$ extracted from leading web sources.

Each photo $p \in \mathcal{P}$ is described by its attributes: u^p identifies the photo owner; tt^p and tu^p indicate when it was taken and uploaded (to Flickr), respectively; g^p_{lat} and g^p_{long}, if given, indicate where (i.e., latitude and longitude) it was taken; and finally $\{\theta^p_i | i = 1...m\}$ is the set of tags associated with the photo. For example, a photo of the Eiffel Tower in Paris may be tagged with "Tour Eiffel, Eiffel Tower, Architecture, Paris, Travel."

One interesting feature provided by Flickr is to allow users organize their photos into *photo-sets*. Observations indicate that the users often group travel-related photos into such photo-sets, with each set devoted to a particular trip or destination. Since descriptive tags attached to the photo-set apply to all photos within the set, we propagate those tags to all the photos within the set.

POIs for each city C are obtained from various sources, including Yahoo! Travel (http://travel.yahoo.com/) and Lonely Planet (http://travel.lonelyplanet.com/). Each POI is then described with the following attributes: *pname* uniquely identifies the POI; *city* is the city it belongs to; and g^ℓ_{lat} and g^ℓ_{long} are its latitude and longitude. Examples including museums, parks, historical sites, and religious places.

Given those basic building blocks, the first step is to convert the raw user photos into individual *timed paths* for a given city C. Intuitively, these paths, which connect various POIs, are constructed from individual photo streams and describe the movements of individual tourists. The process has three main challenges: (i) pruning irrelevant photos that are not associated with the city of interest or not owned by a *tourist*; (ii) mapping photos to the POIs, and (iii) constructing individual *timed paths*. Each timed path is a sequence of POIs traversed by a user, annotated with the time spent by the user at each POI and the transit times between pairs of successive POIs. Figure 1 gives an overview of the entire process, which is described in the rest of the section.

We emphasize here that: 1) while our study focus on leveraging information from a particular rich media sharing site, Flickr, the work is easily extensible to any other social repository, where uses can share semantically and geo-temporally tagged rich media; 2) while we process the internal Yahoo! Flickr data repository, the same protocol can essentially be followed by using the open Flickr API.

3.1 Constructing User Photo Streams

Given a city, pruning away irrelevant photos involves several tasks. The first task is to identify photos that are likely to be taken within the city. The second task is to identify users who are likely to be tourists of the city (as opposed to city residents). Finally, since the ultimate goal is to construct travel itineraries with the prediction of visit and transit times, we must also remove photos whose stored taken time may be inaccurate.

Identifying photos of the city. To identify photos of the city, we mainly leverage the semantic tags associated with users' photos. We start by collecting the set of names of the city, including its proper name and various popular variants, denoted as N_C. We then use the following rule to associate photos with the city:

RULE 1 (PHOTO-CITY ASSOCIATION). *A photo p is associated with the city C if p's set of tags (including any tags propagated from p's photo-set) contain at least one tag matching a name variant in N_C.*

For example, New York City can be referred to as "NYC", "Manhattan", etc., and any photo whose tags include one of those variants is associated with New York City. Note that we do not tap the geo information of the photos at this stage, as we found that it does not significantly improve the city-photo association and is far more costly to compute.

Filtering residents of the city. City residents exhibit different visit patterns from typical tourists. For example, they are not under pressure to visit many POIs within a time constraint. Travel itineraries generated from patterns derived from residents are not likely to be useful for tourists. To address this problem, we adopt the technique of [13] and implement the following heuristic rule:

RULE 2 (TOURIST USER). *A user u is considered as a tourist of the city C, if the span of the taken times between u's first and last photos in C is no more than N days. We empirically set N to 21 in this work.*

The assumption here is that while most tourists concentrate their visits within a short time period from several days to a couple of weeks, residents will take pictures of the city over a much longer period of time. We also enforce that a user visits at least two POIs of C to be considered as a tourist. Once a user u is identified as a non-tourist, all of u's photos associated with city C are eliminated.

Photo taken time verification. Constructing itineraries with accurate predictions of visit and transit times requires the photos to have reliable timestamps, which we verify by the following rule:

RULE 3 (ACCURATE TAKEN TIME). *A photo p is considered to have an accurate taken time tt^p if its minutes and seconds are different from those of its upload time tu^p (both timestamps are at a resolution of 1 second). If the minutes and seconds do match, p is considered to have an accurate taken time if tt^p and tu^p are more than 24 hours apart.*

Intuitively, differences in the seconds or minutes eliminate the possibility that the taken time is set by default to the upload time (a practice adopted by Flickr whenever the taken time info is missing). The 24 hour rule is used to recover photos mistakenly eliminated in the first round due to the time zone differences. In the end, if a photo does not have

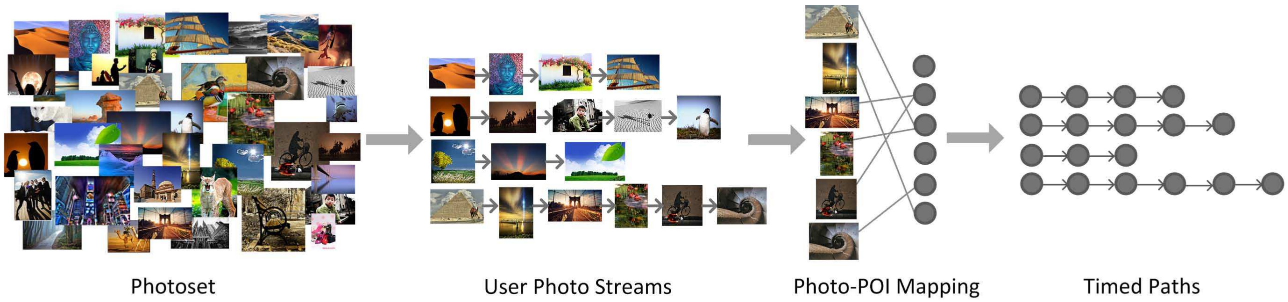

Figure 1: Schematic Diagram for constructing *timed paths*. Given Photosets $\mathcal{P}$, a city C and its POI set $\mathcal{L}_C$, we first construct user photo streams $\mathcal{S}_C$. Second, we map the photos to different POIs to get POI-associated streams $\mathcal{S}_C^{\mathcal{L}}$. Finally from $\mathcal{S}_C^{\mathcal{L}}$, we generate timed paths, TP_C.

an accurate taken time according to the above rule, it is ignored for the rest of the process.

Finally, we group all photos that satisfy all three rules by owner, and within each owner, sort the photos by their taken time. The result is a collection of city photo streams $\mathcal{S}_C$, one for each user.

3.2 Mapping Photos to Points of Interest

The next phase maps photos to POIs. It involves city-specific POI extraction, followed by photo-POI association.

3.2.1 *Extracting Candidate POIs*

In this study, we rely on Lonely Planet to extract the set of popular landmarks ($\mathcal{L}_C$) for a given city C. Furthermore, we employ the publicly available Yahoo! Maps API[1] to extract the geo-locations (i.e., latitudes and longitudes) of these POIs. Geo-locations are returned when querying the Yahoo! Maps API with the names of the POIs.

3.2.2 *Photo-POI Association*

Algorithm 1 Algorithm for Associating Photos with POIs

Require: City-relevant photo streams $\mathcal{S}_C$; a city C;
$\mathcal{L}_C =$ `getPOIs`(C);
for $(p \in \mathcal{S}_C)$ **do**
  for $(\ell \in \mathcal{L}_C)$ **do**
    if (`geoMap`(p, ℓ) || `tagMap`(p, ℓ)) **then**
      `associate`(p, ℓ);
    end if
  end for
end for
return Photo streams with photos associated with city POIs

Given geo information of the POIs, there are two main alternatives to map a photo to a particular POI: geo-based or tag-based. The former relies on matching the photo's geo location to the POI's geo location, while the latter relies on matching the photo's tags to the names of the POIs. Specifically, for the former, we associate a geo-located photo p to a POI $\ell \in \mathcal{L}_C$ whenever ℓ is the POI closest to p, and p was taken within $\delta = 100$ meters of ℓ. This is our preferred method for geo identification, especially for large and distinctive POIs. Such POIs are often photographed from afar (e.g., the Golden Gate Bridge in San Francisco), and therefore the POI extracted from their tags may not match the physical location of where they were taken.

When a photo lacks associated geo information, we apply tag-based matching as a secondary measure. Given a photo tag and the name of a POI, we compute the similarity between the two based on their trigram set similarity. We thus associate a photo p to a POI ℓ whenever ℓ has the highest similarity with any tag of p among all the POIs, with that similarity being above an empirically set threshold $\sigma = 0.3$.

The overall POI association process is depicted in Algorithm 1. It augments the previously identified individual photo streams ($\mathcal{S}_C$) with associated POI information to produce the POI photo stream, $\mathcal{S}_C^{\mathcal{L}}$.

3.3 Generating Timed Paths

Finally, we describe the process of constructing individual timed paths, TP_C, from $\mathcal{S}_C^{\mathcal{L}}$. As summarized in Algorithm 2, it involves two main steps: time segmentation and path construction.

3.3.1 *Segmentation of Photo Streams*

So far, a single stream contains all photos of a single user in a single city. This is not very useful, as two photos might be adjacent in the stream despite being taken (and their corresponding POIs visited) on different days. To address this issue, we segment each stream into sub-streams using a simple heuristic: we split the stream whenever the time difference between two successive photos, $tt^{p_{i+1}} - tt^{p_i}$ is greater than some threshold τ (we use τ=8 hours in our experiments). Subsequently, each sub-stream containing photos from a single POI, or containing less than η=3 photos overall, was discarded. Such sub-streams cannot reliably contribute to the computation of visit and transit times.

3.3.2 *Construction of Timed Paths*

In constructing timed paths we rely on the notion of a *timed visit*, defined as follows.

DEFINITION 1 (TIMED VISIT). *Let $\ell \in \mathcal{L}_C$ be a POI of*

Algorithm 2 Algorithm for Generating Timed Paths

Require: POI-associated photo streams $\mathcal{S}_C^{\mathcal{L}}$; a city C; a time threshold τ;
for $(s \in \mathcal{S}_C^{\mathcal{L}})$ **do**
  $SS =$ `segmentStream`(s, τ);
  for $(ss \in SS)$ **do**
    `pruneNonTourists`(ss);
    `addPaths`(TP_C, ss);
  end for
end for
return Timed Paths TP_C;

[1]http://developer.yahoo.com/maps/

city C. A timed visit *at ℓ is the triplet (ℓ, t^s, t^e), where t^s is the* start time *and $t^e \geq t^s$ is the* end time *of the visit.*

We construct timed visits at ℓ from maximal subsequences of photos associated with ℓ in a photo stream. The time stamp of the first photo in the subsequence determines t^s, while that of the last photo determines t^e. A timed visit implies a lower bound on the actual time spent by the particular user at that POI, since the start and end times represent the earliest time and the last time that a photo was taken at the POI, and not the actual times of arrival at and departure from the POI.

DEFINITION 2 (TIMED PATH). *A sequence of timed visits, $P_C = \{(\ell_1, t_1^s, t_1^e), \ldots, (\ell_k, t_k^s, t_k^e)\}$ is called a* timed path *for city C whenever $t_j^e < t_{j+1}^s$ for $j = 1, \ldots, k-1$. The difference $t_{j+1}^s - t_j^e$ is called the* transit time *from ℓ_j to ℓ_{j+1}.*

Timed paths are induced by the sequence of timed visits derived from a photo stream. Transit times imply an upper bound on the time it took for the particular user to move from one POI to the next.

4. FROM TIMED PATHS TO ITINERARIES

Given the set of timed paths, our goal is to aggregate the actions of many individual travelers into coherent itineraries while taking into consideration POI popularity. To this effect, we define a full undirected graph $G_C = (V = \mathcal{L}_C, E = \mathcal{L}_C \times \mathcal{L}_C)$ on which the following predicates are defined:

$T(\ell \in \mathcal{L}_C)$ is the visit time at each POI ℓ in $\mathcal{L}_C$. Different people might stay for different durations at a POI for many reasons, some of which transcend the value of the landmark itself (e.g. they may stop for a meal there). Moreover, a single user may visit a POI multiple times, each with a different visit time. We take the longest visit of each user u at a POI as u's visit time and set $T(\ell \in \mathcal{L}_C)$ to be the visit time closest to the 75th percentile among all users. This heuristic overcomes much of the noise and compensates for the fact that the visit times we measure are only lower bounds on the real visit times.

$T(e \in E)$ is the median transit time between each pair of POIs in $\mathcal{L}_C$. We allow each timed path to contribute multiple transits between the same pair of points. The median, then, is calculated over the set of all transit times collected from all timed paths. Edges with a single (or no) transit in the data are assigned a transit time of infinity.

$V(\ell \in \mathcal{L}_C)$ is the *prize* or value that an itinerary gets from visiting each POI ℓ in $\mathcal{L}_C$, and is a function of the popularity and visit duration of ℓ. We define the popularity of ℓ as the number of distinct users who visited it (POIs visited by less than 10 users are removed). The dependency on the visit duration is required, to prevent bias towards POIs with short visit durations. We set the prize to be the product of the popularity and the visit duration. Alternative definitions of the prize function may also factor in the number of photos taken at ℓ. Examining the effectiveness of different prize functions is part of our future work.

Note that while (theoretically) transit times should obey the triangle inequality, this may not hold with the transit times we empirically collect. To overcome this difficulty and enforce the triangle inequality, we apply metric completion on G_C. The drawback of metric completion is that an erroneous (short) transit duration at one edge can propagate to many other edges. To mitigate this risk, we ignore transit times of edges for which only a single transition was recorded.

4.1 Itineraries and the Orienteering Problem

An *itinerary* is a path in the graph G_C, where a node (POI) in the path may be visited more than once. Let $\mathcal{I}$ be an itinerary; its prize $V(\mathcal{I})$ is defined as the sum of prizes of the unique set of POIs (i.e., a POI's prize is counted only once even if it is visited multiple times) along the path. The time $T(\mathcal{I})$ of the itinerary is the sum of visit times to the unique set of POIs along the path, plus the transit times along all edges on the path (including those that are traversed more than once). The intuition behind counting prize and visit time only once for POIs visited multiple times is that one might pass through a place several times, always paying the transit time, but spending time there to view the place only once. However, since G_C satisfies the triangle inequality, we can assume without loss of generality that an itinerary is a simple path. Under this assumption, $T(\mathcal{I})$ becomes the sum of visit times of its nodes plus the transit times of its edges.

Given the definitions above, we formulate the Itinerary Mining Problem (IMP) as follows.

Instance: A graph $G = (\mathcal{L}_C, E)$ with edge costs (=transit times) $\{T(e) | e \in E\}$ obeying the triangle inequality, (time) budget B, node prizes $\{V(\ell) | \ell \in \mathcal{L}_C\}$, node costs (=visit times) $\{T(\ell) | \ell \in \mathcal{L}_C\}$ and two nodes $s, t \in \mathcal{L}_C$.

Objective Find an itinerary in G from s to t of cost (=time) at most B maximizing total node prizes.

We note here that, the time budget B is typically set to whole days. And s and t can either be provided by the user or be implicitly set by the itinerary recommendation application based on prior knowledge such as the POI popularity.

PROPOSITION 1. *The Itinerary Mining Problem is NP-Hard.*

The proposition can be easily proved by a reduction from the Hamiltonian Path problem [10].

Note, IMP is closely related to the well-studied, NP-Hard *Orienteering* problem. In the Orienteering problem there are only edge costs (no node costs), and its classic formulation is over undirected graphs. Many polynomial-time approximation algorithms for the classical Orienteering problem and its variants are available in the published literature [3, 5].

A common extension of the Orienteering problem specifies a time window for visiting each node. A node v, then, contributes to the prize collected by a path only if the path reaches v within its time window. Time windows allow us to model recommended visit times of POIs. In this regard, Chekuri et al. [4] give an incomparable approximation ratio of $O(\max\{\log OPT, \log \frac{\mathcal{L}_{\max}}{\mathcal{L}_{\min}}\})$ for Orienteering with time windows, where $\mathcal{L}_{min}$ and $\mathcal{L}_{max}$ are the lengths of the shortest and longest time windows, respectively. Besides, an $O(\log^2 n)$-approximation algorithm for Orienteering with time windows is also given by [5].

Note, typically, the above algorithms all have large time complexity in practice–$O(n^8)$ or worse. However, Chekuri and Pál [6] take a different approach and propose a quasi-polynomial recursive greedy approximation algorithm for Orienteering, whose approximation ratio is $\lceil \log k \rceil + 1$, where

k is the length of the optimal path (in terms of nodes). This approximation ratio is appealing in our context, as we expect a reasonable itinerary to visit only a few POIs each day. By imposing an upper bound $\bar{k}$ on k, the time complexity of the algorithm can be reduced to $O((2 + nA \log B)^{\log \bar{k}})$. Moreover, the algorithm is highly extensible.

4.2 Approximating IMP

As mentioned above, the major difference between IMP and the Orienteering problem is that IMP includes node costs. Hence we reduce IMP to the directed Orienteering problem by adding $T(\ell)$ to the cost of each edge entering ℓ.[2] Algorithm 3 is a restatement of Chekuri and Pál's algorithm [6], which we include here for completeness.

Algorithm 3 Recursive greedy algorithm for Orienteering (RG-QP) [6]

Require: Graph G, source s, destination t, budget B, a set X of nodes that cannot be used and the number of internal nodes allowed on the path k.

If$(\text{dist}_G(s,t) > B)$, **return Infeasible**
$P \leftarrow s, t$
If$(i = 0)$, return P
for (each $v \in V[G]$) **do**
  for (each possible prize a) **do**
    $B_1 \leftarrow \min_b\{\text{RG-QP}(s, v, b, X, \lfloor (k-1)/2 \rfloor) \geq a\}$
    If$(B_1 = \infty)$, continue
    $P_1 \leftarrow \text{RG-QP}(s, v, B_1, X, \lfloor (k-1)/2 \rfloor)$
    $P_2 \leftarrow \text{RG-QP}(v, t, B - B_1, X \cup V(P_1), \lfloor k/2 \rfloor)$
    If$(\text{prize}(P_1 \cdot P_2) > \text{prize}(P))$, $P \leftarrow P_1 \cdot P_2$
  end for
end for
return P

The idea of the recursive greedy algorithm is to guess the middle node v of the path and the amount of prize collected by the path in the first half (i.e., up to node v). The algorithm then determines how much budget has to be invested in the first half to collect the guessed prize, and calls itself recursively on both halves of the path.

4.3 Multi-Day Itineraries

An easy extension of Algorithm 3 allows it to produce multi-day (actually, multi-part) itineraries. The idea is to supply the algorithm with multiple triplets comprised of start-point, end-point, and time allowance. Each triplet represents a "part" and corresponds to the parameters s, t and B of the IMP formulation above. Typically, the end-point of triplet j would be the start point of triplet $j+1$, representing the location where a tourist might spend a night. In a multi-day stay in a city, those might all be the tourist's hotel. A multi-day itinerary is considered *valid* if it is the concatenation of sub-itineraries that connect the source and destination nodes of the triplets while respecting the corresponding time allowance.

A limitation of this approach for automatically constructing itineraries for road trips (as opposed to multi-day city stays) is the need to specify as input to the algorithm the "layover points". It would be interesting to either mine recommended layover spots from user data, or to have the algorithm reach "eligible layovers" at certain intervals (say, every 9-10 trip hours). We leave such extensions for future work.

[2]We omit a few additional details of the reduction that account for the visit time of the source node s.

Table 1: Sample POIs for the five selected cities.

City	#POIs	Sample POIs
Barcelona	74	Museu Picasso, Plaza Reial
London	163	Buckingham Palace, Churchill Museum, Tower Bridge
NYC	100	Brooklyn Bridge, Ellis Island
Paris	114	Tour Eiffel, Musee du Louvre
San Francisco	80	Aquarium of the Bay, Golden Gate Bridge, Lombard Street

Table 2: Data preparation statistics: only *valid* timed paths are counted.

City	#User Streams	#Timed Paths
Barcelona	6,530	6,087
London	31,351	19,052
NYC	6,375	3,991
Paris	14,438	10,651
San Francisco	13,089	12,308

5. EXPERIMENTAL EVALUATION

We evaluate the quality of travel itineraries constructed by our system in an extensive user study conducted through the Amazon Mechanical Turk (AMT)[3] system. Through the user study or the survey, we show that users perceive our *automatically* generated itineraries as being as good as (or even slightly better than) itineraries provided by *professional* tour companies. Furthermore, we show that users are satisfied with the recommended transit and visit times for the POIs within the itineraries. Finally, we discuss some interesting observations learned from the user study. Note that while our system is able to construct multi-day itineraries, these experiments focus on one-day itineraries.

5.1 Experimental Data Preparation

City and POI Selection. We generated itineraries for cities that are popular travel destinations, as reflected in about three years of Flickr data from the second half of this decade. Specifically, the popularity of a city is determined by the number of distinct users who have provided photos for that city (as described in Section 3.1). Five popular and geographically distributed cities were chosen: *Barcelona, London, New York City (NYC), Paris,* and *San Francisco.*

For each city, we obtained a list of POIs by pooling information from different sources (e.g., Lonely Planet) as described in Section 3.2. Table 1 illustrates the number of POIs obtained for each city, as well as some sample POIs for each one. Table 2 illustrates some statistics on the data we extracted Flickr for the chosen cities. Each user stream (Section 3.1) corresponds to an unsegmented sequence of photos by the same user for a single city. Each valid timed path corresponds to a segmented user stream, where the segmentation and validity checking are accomplished as described in Section 3.3.

[3]https://www.mturk.com/

Table 3: Ground truth itinerary sources.

City	Ground Truth Sources
Barcelona	www.barcelona-tourist-guide.com
London	www.theoriginaltour.com
NYC	www.newyorksightseeing.com
Paris	www.carsrouges.com
San Francisco	www.allsanfranciscotours.com

Time **09:00** : Start from **ground zero**
Time **09:00** : Spend 27 minutes at **ground zero**.
Time **09:27** : Transit to **empire state building** (estimated travel time: 52 minutes)
Time **10:19** : Spend 1 hour and 13 minutes at **empire state building**.
Time **11:32** : Transit to **new york public library** (estimated travel time: 15 minutes)
Time **11:47** : Spend 29 minutes at **new york public library**.
Time **12:16** : Transit to **radio city music hall** (estimated travel time: 24 minutes)
Time **12:43** : Spend 51 minutes at **radio city music hall**.
Time **13:34** : Transit to **central park** (estimated travel time: 23 minutes)
Time **13:57** : Spend 40 minutes at **central park**.
Time **14:37** : Transit to **rockefeller center** (estimated travel time: 33 minutes)
Time **15:10** : Spend 37 minutes at **rockefeller center**.
Time **15:47** : Transit to **grand central terminal** (estimated travel time: 22 minutes)
Time **16:09** : Spend 27 minutes at **grand central terminal**.
Time **16:36** : Transit to **chrysler building** (estimated travel time: 6 minutes)
Time **16:42** : Spend 31 minutes at **chrysler building**.
Time **17:13** : Transit to **brooklyn bridge** (estimated travel time: 32 minutes)
Time **17:45** : Spend 36 minutes at **brooklyn bridge**.
Time **18:21** : Transit to **statue of liberty** (estimated travel time: 21 minutes)
Time **18:42** : Spend 42 minutes at **statue of liberty**.
Time **19:24** : Transit to **little korea** (estimated travel time: 26 minutes)
Time **19:50** : Spend 31 minutes at **little korea**.
Time **20:21** : Transit to **ground zero** (estimated travel time: 38 minutes)

(a)

DAY 1

Time **09:00** : Start from **Ground Zero**
Time **09:00** : Transit to **Metropolitan Museum of Art** (estimated travel time: 1 hour and 43 minutes)
Time **10:43** : Spend 2 hours and 9 minutes at **Metropolitan Museum of Art**.
Time **12:52** : Transit to **Empire State Building** (estimated travel time: 1 hour and 30 minutes)
Time **14:22** : Spend 1 hour and 16 minutes at **Empire State Building**.
Time **15:38** : Transit to **New York University** (estimated travel time: 1 hour and 23 minutes)
Time **17:01** : Spend 18 minutes at **New York University**.
Time **17:19** : Transit to **Staten Island Ferry** (estimated travel time: 1 hour and 7 minutes)
Time **18:26** : Spend 1 hour and 10 minutes at **Staten Island Ferry**.
Time **19:36** : Transit to **Ground Zero** (estimated travel time: 56 minutes)
Time **20:32** : Reach **Ground Zero**

DAY 2

Time **09:00** : Start from **Ground Zero**
Time **09:00** : Transit to **American Museum of Natural History** (estimated travel time: 1 hour and 46 minutes)
Time **10:46** : Spend 2 hours and 25 minutes at **American Museum of Natural History**.
Time **13:11** : Transit to **Wollman Skating Rink** (estimated travel time: 1 hour and 2 minutes)
Time **14:13** : Spend 22 minutes at **Wollman Skating Rink**.
Time **14:35** : Transit to **Rockefeller Center** (estimated travel time: 1 hour and 2 minutes)
Time **15:37** : Spend 39 minutes at **Rockefeller Center**.
Time **16:16** : Transit to **Radio City Music Hall** (estimated travel time: 6 minutes)
Time **16:22** : Spend 30 minutes at **Radio City Music Hall**.
Time **16:52** : Transit to **Chelsea Art Museum** (estimated travel time: 34 minutes)
Time **17:26** : Spend 2 hours and 2 minutes at **Chelsea Art Museum**.
Time **19:28** : Transit to **Grand Central Terminal** (estimated travel time: 5 minutes)
Time **19:35** : Spend 17 minutes at **Grand Central Terminal**.
Time **19:52** : Transit to **St Paul's Chapel** (estimated travel time: 34 minutes)
Time **20:26** : Spend 26 minutes at **St Paul's Chapel**.
Time **20:52** : Transit to **Ground Zero** (estimated travel time: 4 minutes)
Time **20:56** : Reach **Ground Zero**

(b)

Figure 2: Sample itineraries constructed by our system for NYC: (a) one-day, (b) two-day itineraries.

Itinerary and Ground Truth Generation. For each city, we generate four itineraries using our system. We first select the city's four most popular POIs and designate them as ℓ_1 (most popular) through ℓ_4. The popularity of a POI is determined by the number of distinct users who have provided a photo associated with the POI. The four itineraries for each city are then constructed by setting the starting point and ending point as (ℓ_1, ℓ_3), (ℓ_1, ℓ_4), (ℓ_2, ℓ_3), (ℓ_2, ℓ_4), with a time budget of 12 hours. Each constructed itinerary is presented as an ordered list of POIs, along with the recommended visit time for each POI and the estimated transit time from one POI to the next. Figure 2 illustrates two sample itineraries for NYC. Note that an application based on our algorithm may either select automatically the starting and ending points of the itineraries (as in this experiment), or expose that degree of freedom to its users. Even users unfamiliar with a city would typically be able to designate such starting and ending points - these might be the more famous POIs of the city, or simply the users' hotels.

In order to compare our automatically constructed itineraries with baseline itineraries, we obtained itineraries provided by top tour bus companies for each city and considered them as ground truth itineraries (Table 3). Note that visit or transit times do not come with typical bus tour itineraries; hence we derive these times using our system and construct the ground truth itineraries.

5.2 Experimental Methodology

Because of the diverse geographical nature of the chosen cities, conducting on-site user studies is difficult as it requires finding enough users who are deeply familiar with foreign cities. Hence we design several user studies using the Amazon Mechanical Turk (AMT) based Human Intelligent Tasks (HITs) and seek feedback on various aspects of the itineraries constructed by our system from a large number of anonymous users.

5.2.1 Amazon Mechanical Turk

The concept of AMT is to provide a *crowd-sourcing* marketplace where *requesters* (i.e., individuals or institutions who have tasks to be completed) and *workers* (i.e., individuals who can perform the tasks in exchange for monetary reward) can come together. AMT provides a platform where the tasks (i.e. HITs) are hosted and executed, money is transferred securely, and the reputation of workers and requesters is tracked. The simplest HIT is often presented as a web form, where the worker answers the questions on the form and AMT transmits the answers to the requester for further analysis. The requester can also specify certain criteria that a worker must satisfy in order to perform the task. Identical HITs are grouped, and a single user can be limited to perform at most x HITs from each group, ensuring that results are produced by a diverse set of users.

5.2.2 User Study Design

For the purpose of our user study, AMT workers are now recruited to work on the HITs with the condition that the same worker can only work on a single HIT in a group (although the worker can work on multiple HITs across multiple groups). Further, to ensure only *reliable* workers are recruited, we enforce on AMT that only workers who have an *approval rate* (i.e., the percentage of a worker's HIT responses being accepted by the requesters) greater than 95% can undertake a HIT. Furthermore, we start each HIT with a qualification test to identify *expert* workers. In the test, the worker is presented with three photos, corresponding to "lesser-known" POIs of the city, and accompanied by multiple options of names of different POIs, only one of which is correct. These POIs (and photos) are chosen such that workers who are familiar with the city should recognize them with ease, while random users would typically not recognize them. For Paris, an example of such a POI is "Pont Neuf". We enforce that only the workers who correctly identify all three POIs qualify to proceed.

5.3 Comparative Evaluation of Itineraries

We first discuss user study results based on a "side-by-side" comparison between two given itineraries.

5.3.1 Survey Questionnaire

We design a survey questionnaire comprising the AMT HITs where we do not reveal to a worker whether an itinerary is the ground truth itinerary or one of those constructed by our system. The goal of this survey is to understand how the workers perceive our system-generated itinerary (say, itinerary A) and the ground truth itinerary (say, itinerary B) via direct comparison. There are two questions of interest in this survey: in Q_1, we ask the workers to rate the overall usefulness of the two itineraries via five different comparative measures: whether itinerary A is *significantly better*, *somewhat better*, *similar*, *somewhat worse* or *significantly worse* compared to itinerary B. The second question Q_2 deals with evaluating the appropriateness of the presented POIs in the two itineraries. This question also evaluates the effectiveness of itinerary A against B based on the five comparative measures discussed above.

In this survey, we have four system-generated itineraries and a ground truth itinerary for each city. Each itinerary is represented as 10 identical HITs. Hence, the total number of HITs in this survey over all five touristic cities is $40 \times 5 = 200$.

Evaluation Metric. To quantify the responses from the workers based on this questionnaire, we present an evaluation metric called *Mean Response Volume*. The metric estimates the usefulness of the itineraries from two aspects, such as the overall utility of the itineraries and appropriateness of POIs. That is, it measures the number of worker responses received per option[4] (in Q_1 and Q_2) in the survey questionnaire. Specifically, for a given option *opt* and a question q, it is given as:

$$\text{MRV}(opt, q) = \frac{1}{n_q(opt)} \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} \sum_{\mathcal{I}} n_q^{\mathcal{I},C}(opt), \qquad (1)$$

where $n_q^{\mathcal{I},C}(opt)$ is the number of workers who chose the option *opt* in question q for the HIT involving our system-generated itinerary I and city C; and $n_q(opt)$ is the total number of workers who responded to option *opt* for question q across all HITs.

5.3.2 Results

From the results in Figure 3, we observe that majority of the workers chose options first and second (i.e. *significantly better* and *somewhat better*) for the system-generated itinerary, against the ground truth one. As shown in the pie charts, we observe that 66% workers found our suggested itineraries better than the ground truth itineraries in terms of overall usefulness (only 14% found our itineraries to be not so useful). Whereas with respect to POI appropriateness, the pie chart indicates that 52% workers found our itineraries better (about 16% workers preferred the ground truth POIs). Hence to summarize, the results reveal that our proposed itineraries are able to improve overall satisfaction of itineraries by a large margin of 52%, while by 36% for POI utility, compared to the ground truth itineraries.

5.4 Independent Evaluation of Itineraries

In the second part of our experimental results, we seek the worker's feedback in order to independently evaluate the utility of a presented itinerary.

5.4.1 Survey Questionnaire

Table 4 illustrates the questionnaire geared towards this purpose. The first two questions (Q_1 and Q_2) evaluate the overall usefulness of the presented itinerary. The overall usefulness is likely to be dependent on the quality of the POIs, the order in which they are to be visited, the visit time of the

[4]To recall, the options are: *significantly better*, *somewhat better*, *similar*, *somewhat worse* and *significantly worse*.

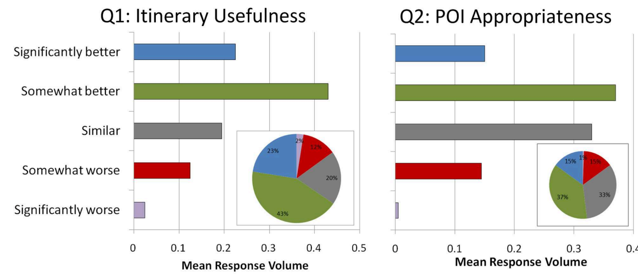

Figure 3: Mean Response Volume for experiments over survey questionnaire II. Two kinds of evaluation are shown: overall usefulness of itineraries (over the five cities) and the appropriateness of the POIs.

POIs, and the transit time between POIs. To obtain a better understanding, we design specific questions to evaluate the visit and transit times (Q_3 and Q_4). Those four questions are presented as multiple-choice questions to measure the feedback from the worker at four discrete qualitative levels.

The next three questions help us examine the itinerary in more detail. We judge the relevance of the recommended POIs in $Q_5(a)$ by asking the worker to tell us the POI(s) s/he finds undesirable in the itinerary. In $Q_5(b)$ and $Q_5(c)$, we seek feedback from the worker on which POIs s/he finds to have unreasonable visit times and transit times, respectively. For all three questions, a multiple-selection drop-down list of POIs (or POI pairs for transit times) from the itinerary is presented to the worker, and the worker is free to select as many as s/he finds appropriate.

For this survey, we construct 10 identical HITs as a group for each itinerary and each city, giving a total of $5 \times 5 = 25$ groups and a total of 250 HITs.

Evaluation Metrics. Our first metric is the *Mean Weighted Response*. Recall that Q_1 through Q_4 in the questionnaire seek workers' feedback on the itineraries in terms of overall usefulness and satisfaction. Each question has four possible responses, with the first reflecting complete dissatisfaction and the fourth reflecting strong satisfaction. We number those responses from 1 (worst) to 4 (best). To provide a quantitative measure, we aggregate the responses to each question q from the workers in the same group, into a single number, Mean Weighted Response (MWR), given as:

$$\text{MWR}(q) = \frac{1}{n_q} \sum_{i=1}^{4} i \cdot n_q(i) \qquad (2)$$

where $n_q(i)$ denotes the number of workers who chose response i to question q, and $n_q = \sum_{i=1}^{4} n_q(i)$ is the total number of workers who answered q. The responses are assigned linear weights, with the weight of each response being its ordinal number. Therefore, the higher MWR(q) is, the better the workers feels about an itinerary.

Our second metric is the *Mean Average Error Fraction* based on the responses for Q_5 in our survey questionnaire. We compute the percentage of the number of POIs ($Q_5(a)$), visit times ($Q_5(b)$), or transit times ($Q_5(c)$), that are considered *bad* or *inaccurate* by a particular worker, out of the total number of POIs, visit times or transit times in the itinerary, averaged over all workers working on the particular itinerary. Formally, for each itinerary $\mathcal{I}$, we have Mean

Table 4: Description of Independent Evaluation Questionnaire.

Q_1: Overall, would you rate the proposed itinerary as:
—Not at all useful to a tourist
—Not so useful to a tourist
—Somewhat useful to a tourist
—Very useful to a tourist

Q_2: How would you rate the set of points of interest included in the itinerary?
—Make no sense
—Mostly inappropriate
—Somewhat appropriate
—Mostly appropriate

Q_3: How would you rate the visit times at the landmarks, as proposed by the itinerary (from a tourist perspective)?
—Not accurate at all
—Somewhat accurate
—Mostly accurate
—Completely accurate
If you picked choices 3 or 4, did you find the visit times too short or too long?

Q_4: How would you rate the transit times between the landmarks, as proposed by the itinerary (from a tourist perspective)?
—Not accurate at all
—Somewhat accurate
—Mostly accurate
—Completely accurate
If you picked choices 3 or 4, did you find the transit times too short or too long?

$Q_5(a)$: Which landmarks you would rather not visit in this itinerary?
$Q_5(b)$: Which visit times are too long/short?
$Q_5(c)$: Which transit times are too long/short?

Table 5: Mean weighted responses for London.

London Itineraries	Q_1	Q_2	Q_3	Q_4
IMP Itinerary 1	3.1	2.9	2.7	2.8
IMP Itinerary 2	3.5	2.1	2.7	2.5
IMP Itinerary 3	3.4	2.5	2.8	2.7
IMP Itinerary 4	3.5	2.7	2.9	3.1
Ground Truth Itinerary	3.4	2.6	2.6	2.6

Error Fraction (MEF):

$$\text{MEF}(\mathcal{I}) = \frac{1}{|U(\mathcal{I})|} \sum_{u \in U(\mathcal{I})} \frac{b(u)}{|\mathcal{I}|} \tag{3}$$

where $U(\mathcal{I})$ is the set of workers responding on itinerary $\mathcal{I}$ and $b(u)$ is the number of POIs (resp. visit times, transit times) reported as *bad* by worker u.

5.4.2 Results

In this section, we describe our analysis on the user study of the above survey questionnaire. We observe the overall satisfaction of the workers for the itineraries. Table 5 illustrates the MWR for all five itineraries of London—the four IMP itineraries generated by our system, and the ground truth bus tour. Observe that the MWR values for all four IMP itineraries and all survey questions are close to (in fact, often better than) the ground truth itinerary. This indicates that the proportion of workers who liked (resp., disliked) the itineraries is the same for our system-generated itineraries and the expert-generated ground truth one.

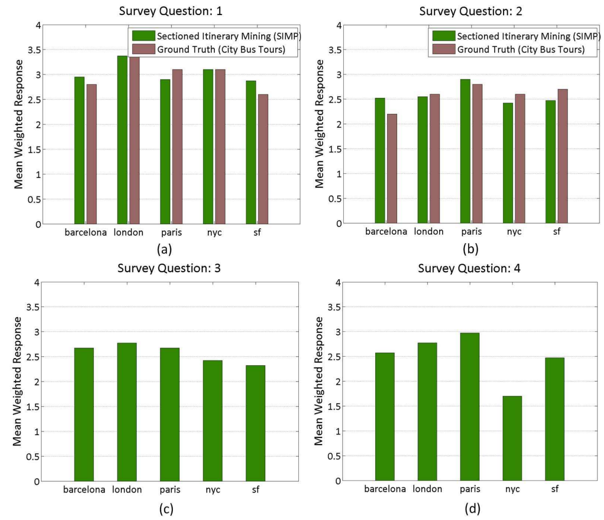

Figure 4: Mean Average Weighted Response from workers on itineraries over five cities—Barcelona, London, Paris, NYC, and San Francisco.

This observation is consistent across all five cities we examined, as shown in Figure 4. In this context, for simplicity, we take the mean of the MWRs of four IMP itineraries to compute a single number, *Mean Average Weighted Response* (MAWR), and compare the MAWR with the MWR of the ground truth itinerary. We observe that in terms of overall usefulness (Q_1) and POI satisfaction (Q_2), IMP itineraries are as good as professionally generated ground truth itineraries. The results also indicate that workers are generally happy with the visit (Q_3) and transit (Q_4) times that our system produces[5].

In our second part, we now perform an analysis of the worker responses on quality of POIs. We take the average of the MEF values over all IMP itineraries to derive a single number, called *Mean Average Error Fraction* (MAEF) and compare it against the MEF of the ground truth itinerary. The results of these "bad" POIs and "inaccurate" visit and transit times across different cities are shown in Figure 5. We observe that the error fractions are reasonably small.

6. CONCLUSIONS

This paper addressed the question of automatic generation of travel itineraries for popular touristic cities from large-scale user contributed rich media repositories. Our solution (1) generates per-user timed paths using geo-temporal attributes of each photo, (2) aggregates those paths into a graph, and (3) computes an approximate solution to a variant of the Orienteering problem to construct itineraries. Extensive user studies that evaluated the quality of the resulting itineraries yielded promising results. To the best of our knowledge, this is the first end-to-end work that leverages geo-temporal breadcrumbs to build travel itineraries. The following paragraphs highlight some of the challenges that we plan on addressing in the future.

Optimizing our parameters. The algorithms described in this paper use multiple parameters. Our concrete set-

[5]Since visit and transit times are generated by our system for ground truth itineraries, we do not provide comparison between them and IMP itineraries on Q_3 and Q_4.

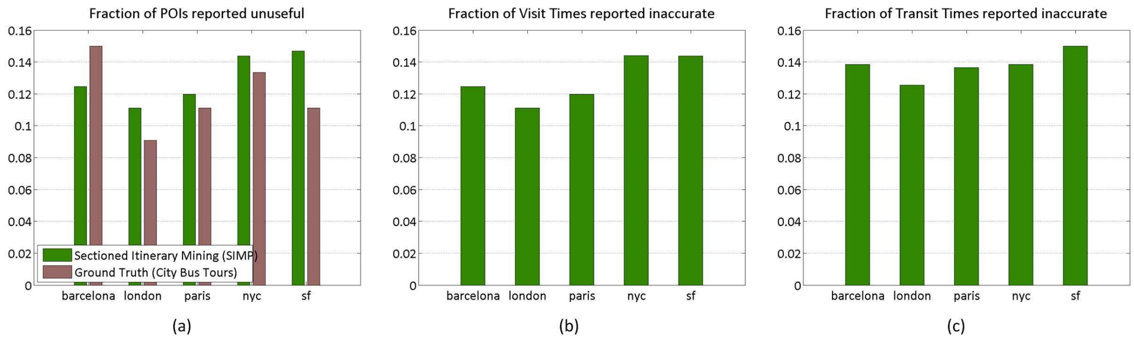

Figure 5: The mean error fraction of (a) POIs, (b) Visit Times, and (c) Transit Times.

ting of those parameters demonstrates the validity of the approach, but is not necessarily optimal. Fine tuning the parameters will require more extensive experiments.

Different strokes for different folks. Our generated itineraries cater mostly to a general tourist. In practice, travelers with different lifestyles, interests, traveling habits often plan different itineraries when visiting the same location, and display different behaviors while on vacation. It is challenging to apply different filtering and aggregation techniques to accommodate different types of travelers, and to construct "off the beaten track" itineraries that cater to niche audiences rather than mainstream crowds.

Time constraints. Itineraries may incorporate temporal constraints such as opening hours of museums, places especially nice at sunset, areas popular on weekends, etc.

Considering co-visitation patterns. Our current methodology may produce itineraries which include pairs of POIs that, despite their individual popularity, rarely appear together in timed paths. Low co-visitation might indicate that those POIs are either almost never simultaneously attractive to the same person, or are equivalent in some sense, rendering one redundant once visiting the other. This translates naturally to defining monotone sub-modular prize functions, as discussed in [6]. The challenge, however, is mining such complex prize functions from Flickr data.

Coverage. Our evaluation focused on 12-hour itineraries in five major cities. It will be interesting to extend our method and its evaluation to smaller and less popular cities, as well as to multi-day itineraries that go beyond city confines.

7. REFERENCES

[1] Shane Ahern, Mor Naaman, Rahul Nair, and Jeannie Yang. World explorer: Visualizing aggregate data from unstructured text in geo-referenced collections. In *JCDL'07*, pages 1–10, June 2007.

[2] Liliana Ardissono, Anna Goy, Giovanna Petrone, Marino Segnan, and Pietro Torasso. Intrigue: Personalized recommendation of tourist attractions for desktop and handset devices. *Applied Artificial Intelligence*, 17(8-9):687–714, 2003.

[3] Nikhil Bansal, Avrim Blum, Shuchi Chawla, and Adam Meyerson. Approximation algorithms for deadline-tsp and vehicle routing with time-windows. In *STOC*, pages 166–174, 2004.

[4] Chandra Chekuri and Nitish Korula. Approximation algorithms for orienteering with time windows. Archived at The Computing Research Repository (CoRR), 2007.

[5] Chandra Chekuri, Nitish Korula, and Martin Pál. Improved algorithms for orienteering and related problems. In *SODA*, pages 661–670, 2008.

[6] Chandra Chekuri and Martin Pál. A recursive greedy algorithm for walks in directed graphs. In *FOCS*, pages 245–253, 2005.

[7] Keith Cheverst, Nigel Davies, Keith Mitchell, Adrian Friday, and Christos Efstratiou. Developing a context-aware electronic tourist guide: some issues and experiences. In *Proc. SIGCHI'00*, pages 17–24, New York, NY, USA, 2000. ACM.

[8] David Crandall, Lars Backstrom, Daniel Huttenlocher, and Jon Kleinberg. Mapping the world's photos. In *Proc. WWW'2009*, pages 761–770, April 2009.

[9] Simon Dunstall, Mark E. T. Horn, Philip Kilby, Mohan Krishnamoorthy, Bowie Owens, David Sier, and Sylvie Thiebaux. An automated itinerary planning system for holiday travel. *Information Technology and Tourism*, 6(3), 2004.

[10] Michael R. Garey and David S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.

[11] Fabien Girardin. *Aspects of implicit and explicit human interactions with ubiquitous geographic information*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2009.

[12] Fabien Girardin, Josep Blat, Francesco Calabrese, Filippo Dal Fiore, and Carlo Ratti. Digital footprinting: Uncovering tourists with user-generated content. *IEEE Pervasive Computing*, 7(4):36–43, 2008.

[13] Fabien Girardin, Filippo Dal Fiore, Carlo Ratti, and Josep Blat. Leveraging explicitly disclosed location information to understand touristic dynamics: A case study. *Journal of Loc.-Based Servcs*, 2(1):41–54, 2008.

[14] David Leake and Jay Powell. Knowledge planning and learned personalization for web-based case adaptation. In *Proc. ECCBR 2008*, pages 284–298, 2008.

[15] Adrian Popescu and Gregory Grefenstette. Deducing trip related information from flickr. In *Proc. WWW'2009*, pages 1183–1184, April 2009.

[16] Tye Rattenbury, Nathaniel Good, and Mor Naaman. Toward automatic extraction of event and place semantics from flickr tags. In *Proc. SIGIR'07*, pages 103–110, July 2007.

[17] Chih Hua Tai, De Nian Yang, Lung Tsai Lin, and Ming Syan Chen. Recommending personalized scenic itinerary with geo-tagged photos. In *Proc. ICME'2008*, pages 1209–1212, 2008.