Parameter-Free Discovery and Recommendation of Areas-of-Interest

Dmitry Laptev
ETH Zurich, Switzerland
Yandex, Moscow, Russia
dlaptev@inf.ethz.ch

Alexey Tikhonov
Yandex, Moscow, Russia
altsoph@yandex-team.ru

Pavel Serdyukov
Yandex, Moscow, Russia
pavser@yandex-team.ru

Gleb Gusev
Yandex, Moscow, Russia
gleb57@yandex-team.ru

ABSTRACT

The task of discovering places of interest is a key step for many location-based recommendation tasks. In this paper we propose a fully unsupervised and parameter-free approach to deal with this problem based on the collection of geotagged photos. While previous papers are mostly devoted to discovering *points* (POI), we focus on *areas of interest* (AOI). Recommendation of areas better matches the traditional tourist goals and allows to robustly incorporate the interests of many users resulting in less subjective recommendations. The typical question that can be answered with the algorithm is formulated as *"Where can one spend T minutes/hours walking around to observe as many attractive places as possible?"*

The proposed method starts with estimating multiple density hypotheses and then partitions these densities with the watershed segmentation algorithm into regions. The implicit parameters are tuned automatically to fit tourist goals and constraints resulting in a *parameter-free* algorithm. In spite of the parameter optimization overhead, the method is computationally efficient as it employs fast Fourier transforms for convolutions.

We test our approach on 7 different cities and quantitatively show that the proposed method consistently outperforms the state-of-the-art DBSCAN algorithm and its modern modification P-DBSCAN, providing up to several times better recommendations in terms of time required for city exploration.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge acquisition*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

SIGSPATIAL'14, November 04 - 07 2014, Dallas/Fort Worth, TX, USA

http://dx.doi.org/10.1145/2666310.2666416.

General Terms

Algorithms, Experimentation, Performance

Keywords

Areas of Interest Discovery, Location-based Social Networks and Services, Geotagged Photos, Recommendation Systems, Density Estimation, Watershed Segmentation

1. INTRODUCTION

Travelling to new places always requires significant preparations if one wants to discover some tourist attractions or just looks where to spend some spare time. For larger cities it is becoming less of an issue, as they are usually covered by professional travel guides that list and describe their popular points of interest and walking routes. The problem arises more acutely for smaller cities with a few popular tourist attractions and low tourist traffic. In these cases, either no relevant sources of information exist at all or these sources represent only subjective opinions of a few city locals. This poses the problem that we address in this paper: *automatic discovery and recommendation of attractive areas* based on user-generated data. Using this data leads to less subjective recommendations as it incorporates interests of many users.

Location-based social networks and services allow their users to share locations and location-related content [1]. Digital cameras and smartphones with embedded Global Positioning System (GPS) make it possible to discover not only shared content itself, such as photos, notes and statuses, but also its location attributes. Our method for AOI discovery can work with any source of location-based information, however, we focus on geotagged photos due to the reasons discussed in Section 2.1.

In this work we propose an algorithm to build a location recommendation system given a collection of geotagged photo coordinates. The algorithm can recommend either discovered *Points of Interest (POI)*, or *Areas of Interest (AOI)*. Discovering and recommending POI is a well-studied problem in the literature [4, 15, 18]. Unlike these studies, we focus on the the problem of discovering and recommendation of Areas of Interest [5, 8, 9], which received considerably less attention in the literature. We argue that effective solutions to this problem have the potential to result into sought-after applications, especially useful for those tourists who visit off-the-beaten-path cities and regions. Particularly, solving AOI instead of POI discovery problem is capable to:

- Discover not only specific points on a map, but also nice parks and squares, embankments and streets.

- Provide less subjective recommendations for tourists than POI discovery methods, as any AOI contains a variety of POIs reachable within a walking distance and, hence, such a recommendation has a higher chance to satisfy a tourist than a recommendation of a single POI. Besides, not only tourists often prefer to visit certain areas rather than always focus on specific POIs, but also famous travel guides often group POIs into areas (neighborhoods, districts) of interest[1][2].

- As for the proposed method, formulating it as an AOI discovery algorithm, allows to exploit the desired region properties, resulting in a *parameter-free approach*, which, as we demonstrate in our experiments, advances the state-of-the-art methods for AOI discovery and recommendation.

AOI discovery and recommendation algorithms are designed to provide a tourist with a set of ranked AOIs that can be explored under a limited time budget. These AOIs should include as many attractive places as possible. In order to meet these tourists goals and constraints, we formulate the following quality measure, described in detail in Section 4.2. For a set of known points of interest (POIs), we compute the time required to explore a certain percent of these points *(coverage)* when following the recommendations produced by an algorithm. The smaller the time needed to achieve a certain level of coverage, the better the quality of an algorithm.

We use this metric to compare the algorithm that we propose in this paper with the baselines in Section 4. We show that our algorithm consistently outperforms the state-of-the-art approaches, i.e. recommendations of our algorithm can be explored faster than recommendations produced by baselines.

A very important feature of the algorithm that distinguishes it from others is that it requires *no parameters tuning*. The internal parameters are hidden from end user and are chosen automatically based on the desired properties of the outcome (the size of discovered AOIs, the number of AOIs to explore). In our experiments, the size of AOIs is chosen in such a way that a tourist spends about 10 minutes walking around one AOI to explore it. Fixing AOI size leaves only one degree of freedom: the number of regions to recommend, which is then linearly connected with the spare time available. The value of 10 minutes is just a quantization of the walking time: tourists are rarely interested in a more fine-grained time measurement.

For example, if one wants to spend 3 hours walking around the most attractive places in the city, the algorithm will release about 18 recommended AOIs. All of them will be ranked by popularity and each of them will take roughly 10 minutes to explore. In most cases the recommended AOIs are not separated from each other, but grouped into larger clusters (see Figure 1) and therefore they take longer to stroll around. If a tourist has less (more) time, then fewer (more) AOIs will be recommended to satisfy the tourist's constraints. Experiments in Section 4 show that 3 hours is enough to visit 70% of attractions in most of the cities in the dataset.

The rest of the paper is organized as follows: in Section 2 we discuss the related studies in the field of both POI and AOI recommendation; in Section 3 we describe the method and its improvements that make it computationally feasible; in Section 4 we describe the dataset and provide evaluation results together with the baselines comparisons.

2. RELATED WORK

Retrieving geospatial information from location-based user-generated data is a relatively new but a very fast-growing field. It was especially boosted with the recent development of mobile services for sharing location-related information. The variety of different problems in this field is also very broad: starting from a single point of interest discovery [4, 15, 18] and advancing to personalized recommendations based on social graphs joint with location information [1].

In this section we investigate the relations between the recommendation of AOI proposed in the current paper and other works from three different perspectives. Frist, we compare data sources used in different studies to find the one that better suits our needs. Second, we provide an overview of the state-of-the-art algorithms for attractive areas discovery and explain the advantages of our approach. And finally we relate the proposed method to other types of tasks and show how they can benefit from this algorithm.

2.1 Data sources

The first major difference between the studies in the field of geospatial recommendations lies in the very source of location-related information. Each data source represents only a specific subset of users or content, which introduces a bias and in turn affects further problem solving.

Some recommendation methods focus on the data on location points collected from GPS trackers that explicitly share the location information [19]. These approaches are based on the exact trajectories of every person walking with a GPS-enabled device. However, the obvious drawback of such type of datasets is their low availability caused by relatively sparse use of specialized GPS devices, while tracking with smartphones is still prohibitively energy consuming. Therefore, it is applicable to only very specific studies with a small number of people involved.

Another source of information is represented by the services where users explicitely share their location, such as Foursquare[3] [9, 15]. These services are widely used by many people around the world, so such data is much more representative than the data collected from GPS trackers. The method proposed in the current paper can be applied to these datasets with no additional modifications and it produces reasonable recommendations. However, locations gathered from, for example, Foursquare tend more to the venues relating to daily activities: restaurants, coffee shops, stores, airports, fitness, pubs [2]. Therefore, the recommendations that can be deduced from this dataset represent not *attractive places* (mostly touristic), but *activity venues* (attended mostly by locals). The task of activity venues discovery is also very important, but it is out of scope of this paper.

On the contrary, people usually share their discoveries of attractive places via sharing visual information, such as pho-

[1]http://timeout.com/paris/en/by-area/
[2]http://nycgo.com/neighborhoods/
[3]https://foursquare.com/

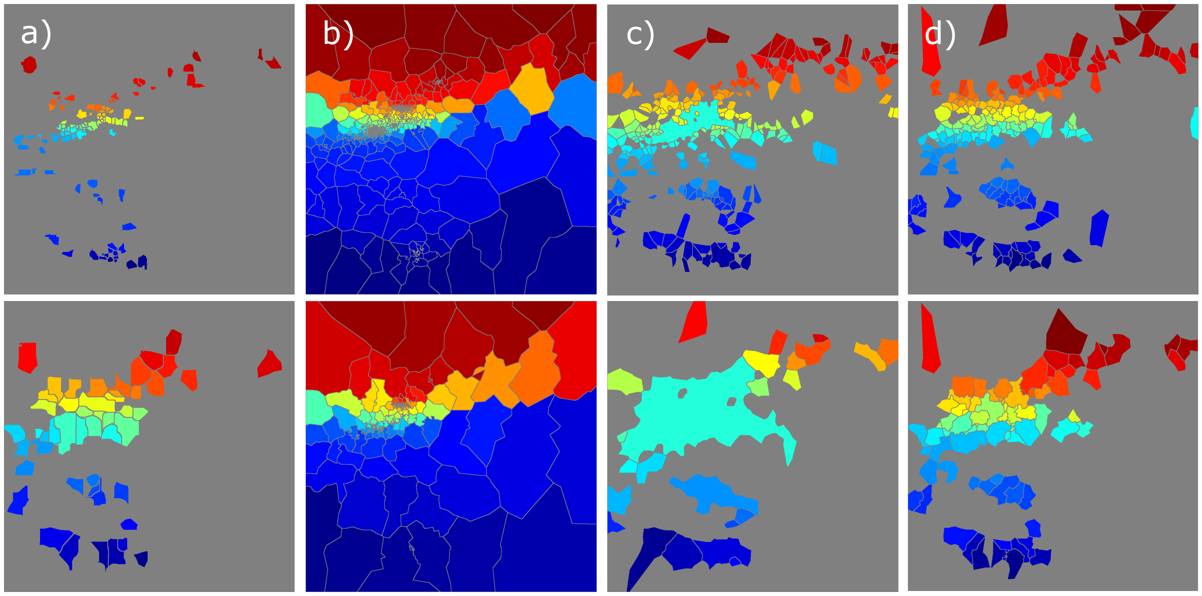

Figure 1: An example of regions discovered with our method (column a), K-means (b), DBSCAN (c) and P-DBSCAN (d). Here different colors correspond to different AOIs, grey areas correspond to the non-clustered regions with small density values (K-means does not support non-clustered regions). Top and bottom rows correspond to different values of the algorithm parameters. The input photo coordinates are shown in Figure 2. Images in column (a) visualize the resulting watershed segmentation $L(h)$ for two different h. Our method produces clusters of arbitrary shape and of approximately the same size. K-means results in many clusters of impractical size in the areas where the density is low (see Section 2.2). DBSCAN often discovers one large region that requires a very long time for exploration.

tos. And photos made with GPS-enabled cameras allow us to see the location where each photo was taken. Therefore, the most common source of information about attractive places of interest is the photo-sharing services, such as Flickr[4] [4, 16], Instagram[5], or Yandex.Photos[6]. The specific choice of the data source depends on which service is more popular in the specific part of the world. In this work we focus on Eastern Europe, so we use a collection of photos gathered from Yandex.Photos media sharing service (more than 4.8 million monthly user audience[7]).

2.2 Points and Areas of Interest discovery

In this section we review related methods for both POI and AOI discovery. We also show why only a small subset of POI discovery methods can be applied to the AOI discovery problem.

Given a collection of coordinates of geotagged photos, the task is to find some places of interest. Quite naturally (if no additional information is available), we consider some areas attractive, if there are many photos taken in/around these areas. All solutions to the problem differ in formalization of two concepts: how to define the areas, and how to count the points in/around the area.

K-means is a well-known mean-based clustering technique that was proposed for POI discovery [7], but that can be easily adapted to AOI discovery. K-means partitions the whole set of photos into closely connected subsets. From the spatial point of view it is equivalent to partitioning the city into Voronoi cells [6]. The center of mass of every cell can represent a specific POI, while a cell itself can represent AOI.

The major disadvantage of this approach is that it assigns every single point to one of the clusters constantly resulting in many large cells where the photo density is small (see Figure 1, b). The time required for AOI exploration in this case is very large and therefore the method appears to be impractical comparing to the method proposed in the current paper.

MeanShift [4, 10] instead looks only for the areas with high photo density instead of assigning every photo to a cluster. In a way similar to K-means it starts with some coordinates and iteratively shifts them in the direction of larger density values. By proceeding like this, MeanShift finds local peaks of density which can be interpreted as POI candidates. However, as it operates only with points, it cannot be directly applied to AOI discovery, unlike our method that finds AOI of arbitrary shape.

DBSCAN is the most commonly used density-based clustering method applied both for POI and AOI discovery [5, 17]. It adds a new photo to a cluster, if it is within a certain radius around the photos that are already in the cluster and if it has at least a certain number of nearest neighbours. Both the radius and the minimum number of neighbours are the parameters of DBSCAN. As well as our method, DBSCAN also introduces non-clustered regions (shown in grey in Figure 1, c) where the photo density is low. Photos in these regions are not assigned to any cluster, letting to

[4]https://flickr.com/
[5]http://instagram.com/
[6]http://photos.yandex.ru/
[7]TNS Web Index (http://tnsglobal.com/), April 2014

overcome the disadavantages of K-means described above.

The main disadvantage of DBSCAN that is also reported in [8] is that for any set of parameters it finds regions that are significantly larger than others in dense areas (Figure 1, c). Recommending these regions to a tourist makes certain sense, but, as we show in Section 4 would significantly increase the minimal walking time required to explore the city, as compared with our algorithm, which, on the contrary, meets any tourist constraints by design.

P-DBSCAN [8] is the recent modification of DBSCAN, that aims to solve this problem by introducing a new parameter: maximum possible density change when adding a new point. Because of it, large areas produced by DBSCAN are split into parts of almost constant densities (Figure 1, d). That partly overcomes the issue of DBSCAN for AOI discovery. P-DBSCAN was also compared to other methods for POI discovery and appeared to be the best when its parameters were tuned [18]. We choose both DBSCAN and P-DBSCAN as the baselines for our method and report the evaluation results in Section 4.

2.3 Applications of AOI discovery

In this section we provide a brief overview of different tasks that are related to AOI discovery. Beside recommending best areas for city exploration, which we focus on throughout the paper, there are many other important applications that require AOI discovery as a part of their pipelines. The proposed method outperforms other techniques for AOI discovery in terms of time required for recommendations exploration (see Secton 4). And therefore can contribute to the solution of these related applications.

The authors of [4] and [10] use AOI as a building block for a route recommendation system. They connect the regions by estimating photographers tracks. This can be done using the sequence of the coordinates of the photos taken by a photographer. Incorporating also the information on the time the photos are taken helps to provide season-specific recommendations, or recommend the best within-a-day schedule for a tourist.

Recent advances in computer vision allow to analyze photos visually in order to recognize objects and recommend them to users [11]. The photos can be automatically matched to the photos with known description in order to identify and rank some specific classes of objects and landmarks. However, visual matching is extremely computationally expensive. And preliminary AOI discovery is widely used to solve this problem. As the first step, photos are assigned to location-based regions. Then visual analysis is conducted only within the discovered AOI.

A different very fast-growing direction of research is geosocial factors analysis [3]. Combining social graph of user-to-user relations with the location graph of places that users visit improves the understanding of their interests and provides a support for personalized recommendations [1]. In most of the scenarios, these recommendations are also based on preliminary collected and ranked areas of interest.

The method proposed in the current study discovers and recommends better regions of interest (as discussed in Section 4). Therefore, every research direction described above can benefit from our algorithm with minimum effort as no tuning is required.

3. METHOD DESCRIPTION

Notation. Let us denote a set of photo coordinates by $x_i \in \mathbb{R}^2$, $i \in 1, \ldots, N$. Here N is the overall number of photos available in the dataset, $x_i = (x_{i,1}, x_{i,2})$, where $x_{i,1}$ denotes longitude and $x_{i,2}$ denotes latitude. We also map all the point x_i to a K by K square grid for the purpose of discretization (we discuss the choice of K in Section 3.1). On this grid we compute histogram representation G of the original dataset, which is then smoothed with a gaussian kernel with an internal bandwidth parameter h to produce density $D(h)$. This internal parameter is selected during the algorithm training and does not need to be set by a user.

The proposed method for tourist attractions recommendations consists of the following five steps that we describe in detail further in the paper:

- coordinate space discretization (grid mapping) and computing histogram representation G (Section 3.1);
- gaussian density estimation with different values of kernel parameter h resulting in density hypotheses $D(h)$ (Section 3.2);
- generating grid partition hypotheses $L(h)$ via watershed segmentation for each density hypothesis $D(h)$ (Section 3.3);
- relevant parameter selection h_{opt} based on the desired properties of partition $L(h)$ (Section 3.4);
- region popularity ranking (Section 3.5).

We choose Gaussian kernel density estimation, because Gaussian function is a commonly used kernel with infinite support. The motivation behind choosing a kernel with infinite support [14] is that it allows us to estimate density in the regions that are only *sparsely covered* with the photos. The most common examples of these regions are parks and lakes or rivers waterfronts. Every photo taken in a park, even though it is quite distant from its center, contributes to the local density maxima estimated by using Gaussian kernel.

Local peaks of the density show themselves possible POIs. The larger the density value at a point, the more photos were taken around it. However, we go further and find not individual points, but *areas of interest* with watershed segmentation of the estimated density. As discussed in the introduction, this segmentation allows us to find spatially distributed tourists attractions, make the method parameter-free, and gather important statistics for further ranking AOI for recommendation.

The approach to finding the optimal value of h is motivated by a high-level formulation of the tourist goals: "What are the best places in the city one can spend approximately 10 minutes walking around?". Here 10 minutes is an arbitrary time chosen to represent a conventional time unit that could be spend to explore an attractive area. This motivation in our terms directly transfers to the unit areas we would like to recommend: 10 minutes of walking is enough to explore, on average, an area of about 0.1 square kilometers. We pick the value of 10 minutes as it seems to be small enough for time quantization, but still can be large enough for a stroll.

All the steps of the proposed pipeline with examples of every stage output are sketched in Figure 2. In the following sections we describe details of our method, which is further summarized in Algorithm 1.

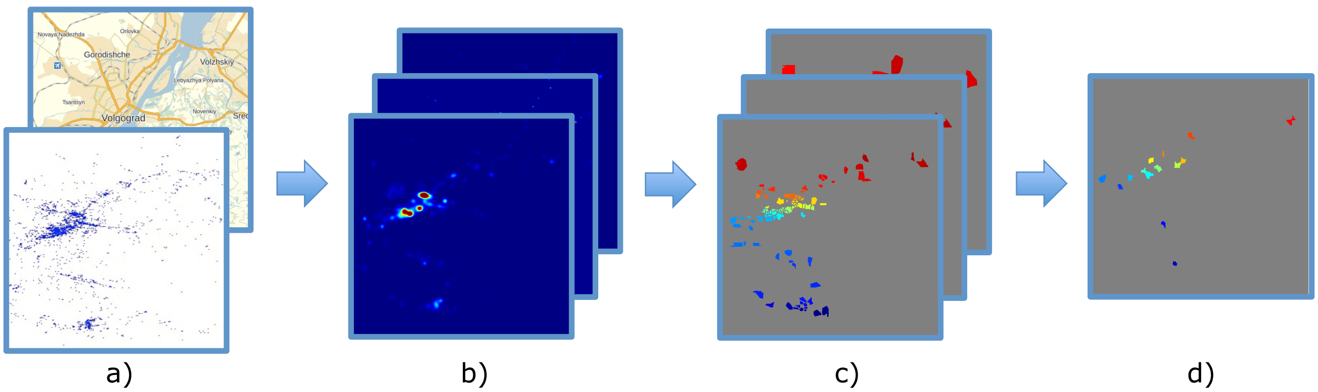

Figure 2: The pipeline of the proposed method. Coordinates of photos are mapped to a grid, and their histogram representation is computed (a). The histogram is then used for density estimation. Estimation is performed with different values of the internal bandwidth parameter and produces a collection of densities (b). Here larger values of density correspond to the red color and smaller values correspond to the dark blue. Watershed segmentation procedure is applied to the densities and results in partition hypotheses (c). Each color here represents an AOI candidate, grey color shows non-clustered regions. Among the partitions, the one is selected that corresponds to the tourist walking constraints, all of the selected areas are ranked, and a subset of top ranked areas (14 regions in this example) is released as a recommendation to a user (d).

3.1 Discretization

For the purposes of convenient numerical computations of the following density estimation and faster numerical integration for ranking, we first discretize the original space, where photo coordinates are represented, to a uniform grid with a mapping function $\mathbf{F} : \mathbb{R}^2 \longrightarrow \{1, \ldots, K\}^2$. We select $\mathbf{F}(x_i) = (\mathbf{F}_1(x_{i,1}), \mathbf{F}_2(x_{i,2}))$ to be the composition of linear scaling and ceiling function:

$$\mathbf{F}_j(x_{i,j}) = \left\lceil \frac{K}{10} + \frac{8K}{10} \frac{x_{i,j} - \min_t(x_{t,j})}{\max_t(x_{t,j}) - \min_t(x_{t,j})} \right\rceil,$$

where $j \in \{1, 2\}$, $\lceil \cdot \rceil$ is the ceiling function, that maps a real number x to the smallest integer not less than x. Incorporating bias term $\frac{K}{10}$ and scaling term $\frac{8K}{10}$ introduces 10% margins around the image of the set of points $\{x_i\}$ under $\mathbf{F}$.

For large values of K, function $\mathbf{F}$ can be approximately considered as a bijective function. For example, if we set $K = 2^{12}$ for a large city with the area of 1000 square kilometers, x_i can be recovered from $\mathbf{F}(x_i)$ with a deviation of at most 5.5 meters, which is negligible for our purposes.

Once the coordinates are mapped, we represent an original set of points by the histogram $G \in \mathbb{R}^{K \times K}$, defined by:

$$G_{i,j} = |\{t : \mathbf{F}(x_t) = (i, j)\}|$$

The method of AOI identification then proceeds by estimating the density of the distribution of the points $\mathbf{F}(x_i)$ in the grid and partitioning the grid into clusters, which are further used as AOI candidates.

3.2 Density estimation

Gaussian kernel density estimation approximates the density $D(h)$ at every node of the grid with the following equation:

$$D_{i,j}(h) = \frac{1}{N} \sum_{p=1}^{K} \sum_{q=1}^{K} \frac{G_{p,q}}{2\pi h^2} \exp\left(-\frac{(i-p)^2 + (j-q)^2}{2h^2}\right), \quad (1)$$

where h is the bandwidth parameter: the larger h, the smoother the estimated density [14].

The usual drawback of direct computation of $D_{i,j}(h)$ using Equation 1 is its computational cost. Computing $D(h)$ as described in Equation 1 requires to sum over all the nodes of the $K \times K$ grid to compute each entry $D_{i,j}$. Moreover, as we generate multiple hypotheses, we perform this procedure many times, which further increases the computational costs. In Section 3.6 we introduce some common computational tricks to deal with these efficiency issues.

3.3 Partitioning

Local density peaks carry little information about possible sizes and shapes of AOIs, so to perform region recommendations, we partition the grid according to the density hills with the watershed segmentation method [12]. Watershed segmentation finds the regions of the grid corresponding to the local maxima and takes into account regions compactness. It starts with individual nodes of the grid (also called markers) as small regions and then enlarges them iteratively until they either meet each other, or meet low-density regions. We initialize a marker at each point corresponding to a local maximum of the density $D(h)$. That results in the partitioning of the grid into small compact clusters in the areas with large density. Each cluster, depending on the density smoothness, corresponds to a specific area of interest around some local peak of interest. Some examples of watershed segmentation are shown in Figures 1, 2, and 4.

Formally, watershed segmentation produces an integer matrix $L(h) \in \{0, \ldots, R(h)\}^{K \times K}$ such that nodes of the grid with the same non-zero value r correspond to the same region r. Zero values denote the borders and the regions that were not partitioned due to very small density. Overall number of regions depends on the parameter h and is equal to $R(h)$. The details of the watershed algorithm can be found in [12], so we leave them out of the scope of this paper.

Figure 1 shows qualitative comparison of the proposed approach, K-means, DBSCAN, and P-DBSCAN (see Section

2.2) for one of the cities in the dataset. The proposed approach tends to discover regions of approximately the same size, while DBSCAN results in the small regions, where the density is low, and in one major region in a dense area. This one large region may require itself impractical time to explore. K-means also produces many large clusters of impractical sizes as discussed in Section 2.2.

3.4 Parameter selection

We perform watershed segmentation for each density hypothesis obtained by estimations with different values of the parameter h. As discussed at the beginning of the section, one needs to select h in such a way that the average area of the regions established is about 0.1 square kilometers. We first compute the average area $\mathbb{E}_h(\text{area})$ of a region for a given value of the parameter h:

$$\mathbb{E}_h(\text{area}) = \frac{1}{R(h)} C_{\text{long}} C_{\text{lat}} \delta_{\text{grid}} \sum_{r \in \{1,\ldots,R(h)\}} |\{(i,j) : L(h)_{i,j} = r\}|$$

where $|\{(i,j) : L(h)_{i,j} = r\}|$ is the number of nodes of the grid covered by region r. Here we use the constant δ_{grid} for the area of a domain in the Euclidean space $\mathbb{R}^2$ that is mapped onto a unit cell of the grid. Formally, we set

$$\delta_{\text{grid}} = \frac{1}{K^2} \frac{10^2}{8^2} \left(\max_t x_{t,1} - \min_t x_{t,1} \right) \left(\max_t x_{t,2} - \min_t x_{t,2} \right)$$

At last, we use C_{lat} and C_{long} for the real-world distance encoded in one degree of latitude and one degree of longitude respectively. Quantity C_{lat} is constant all around the world and is approximately equal to 111 kilometers. C_{long} depends on the value of latitude, but varies insignificantly within the area of one city and therefore can be considered constant.

Then the value of the parameter h that satisfies the walking constraints is given by the following equation:

$$h_{\text{opt}} = \max_h \{h : \mathbb{E}_h(\text{area}) \le 0.1\}$$

Here we just find the maximum value of the internal parameter h that gives the average region area smaller than 0.1 square kilometers.

3.5 Region ranking

The final goal, however, is not only to partition the city into regions, but to rank the regions according to the popularity and find the most attractive AOIs. We introduce the ranking of the regions according to the integral density over the whole region, which directly corresponds to the number of the photos taken in a region and in some area around it. So the rank of the region $r \in \{1,\ldots,R(h_{\text{opt}})\}$ is computed using the following equation:

$$\text{rank}(r) = \sum_{(i,j):\ L_{i,j}(h_{\text{opt}})=r} D_{i,j}(h_{\text{opt}}) \tag{2}$$

Here $D_{i,j}(h_{\text{opt}})$ is a density calculated by Equation 1 and $L_{i,j}(h_{\text{opt}})$ is the partitioning obtained with the watershed segmentation.

This metric is more robust than a single value of the density at one point as it takes the surrounding area into consideration. As the watershed segmentation produces regions of approximately the same size, we need no further renormalization when comparing our recommendation approach to the rankings of AOIs produced by other techniques.

3.6 Efficiency issues

Equation 1 is very hard to compute when the size of the grid is relatively large. Direct computation requires $O(K^4)$ operations, which is computationally infeasible for $K \le 10^3$. The common approach is to instead interpret density estimation on a grid as a convolution of the histogram G with the gaussian kernel. That allows one to apply convolution theorem that states that the Fourier transform of a convolution of two signals is the pointwise product of Fourier transforms of each of the signals [13].

Assume that fft and ifft denote direct and inverse fast Fourier transform respectively. Then the density can be also computed with the following formula:

$$D(h) = \text{ifft}\left(\text{fft}(G) * \text{fft}\left(g(h)\right)\right)$$

where $g(h)$ is a gaussian kernel with the parameter h and "$*$" stands for the component-wise multiplication of two matrices. That allows to compute the density with only $O(K^2 \log K)$ operations, which can be done within less than a second for reasonable values of K.

Another minor speeding-up is achieved due to the fact that $\text{fft}(G)$ does not depend on h and therefore can be computed only once and further used to compute $D(h)$ for all values of h. Moreover, $\text{fft}\left(g(h)\right)$ does not depend on the data, and therefore can be precomputed for different values of h in advance. Therefore, we need to apply fft procedure only once.

Our approach to recommendation AOIs is summarized in Algorithm 1.

Algorithm 1 Region discovery and recommendation

Require: photo coordinates $x_i, i = 1,\ldots,N$
Require: spare time available T (in minutes)
  $G_{i,j} = |\{t : \mathbf{F}(x_t) = (i,j)\}|\ \forall i,j \in 1,\ldots,K$
  $FG = \text{fft}(G)$
  load precomputed $fg(h) = \text{fft}\left(g(h)\right)$
  for h in $\{2^{-20},\ldots,2^{-1}\}$ **do**
    $D(h) = \text{ifft}\left(FG * fg(h)\right)$
    $L(h) = \text{watershed}\left(D(h)\right)$
    $R(h) = \max(L(h))$
    $\mathbb{E}_h(\text{area}) = \sum_{r\in\{1,\ldots,R(h)\}} |\{(i,j) : L(h)_{i,j} = r\}|$
    $\mathbb{E}_h(\text{area}) = \mathbb{E}_h(\text{area}) C_{\text{long}} C_{\text{lat}} \delta_{\text{grid}} / R(h)$
    if $\mathbb{E}_h(\text{area}) < 0.1$ **then**
      $h_{\text{opt}} = h$
    end if
  end for
  for r in $\{1,\ldots,R(h)\}$ **do**
    $\text{rank}(r) = \sum_{(i,j):L_{i,j}(h_{\text{opt}})=r} D_{i,j}(h_{\text{opt}})$
  end for
  return $\frac{T}{10}$ top ranked regions

4. EXPERIMENTS

As stated in Algorithm 1, we perform all the experiments varying the parameter h in the range from 2^{-20} to 2^{-1}. The size of a grid K is fixed in advance and equals to 2^{10} (that gives high enough resolution for all the cities explored). If the city is significantly larger and better resolution is required, it can be enlarged and will not affect any other part

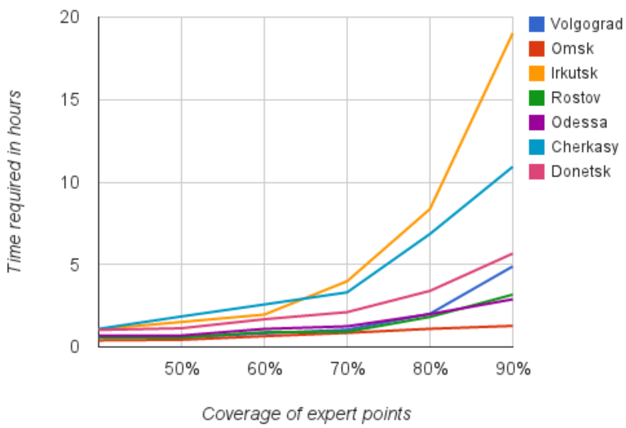

Figure 3: An illustration of the time-vs-coverage tradeoff for all the cities in the dataset and for our method: the larger the percent of points we want to cover, the more time we need to invest. And vice versa: the more time we have, the more places of interest we can explore.

of the algorithm. As discussed in Section 3, we use 10 minute walking distance to determine desired AOI size. Increasing it up to 20 minutes changes the results insignificantly, further increase introduces AOIs with larger surroundings and therefore results in exploration time overhead.

We compare with the two baselines: DBSCAN and P-DBSCAN (see Section 2.2 for the explanation of this choice). For both of them we use exactly the same ranking scheme as for the proposed method (see Section 3.5).

4.1 Dataset

All methods are tested on a dataset of photos obtained from Yandex.Photos[8] — one of the largest photo sharing services in Eastern Europe, most popular in Russia and Ukraine. We select 4 Russian and 3 Ukrainian cities varying in size and population, where also the number of geotagged photos available varies significantly: from 600 to 12000. The cities are: Volgograd, Omsk, Irkutsk, Rostov-on-Don, Odessa, Cherkasy, Donetsk.

To provide quantitative results we collect a database of attractive places for all seven cities. The database consists of points, carefully selected by local experts to answer the following question: "Where would you bring a friend of yours visiting your city for the first time?". Lists of POIs collected in this manner are not restricted to attractions that one can always find in a pocket travel guide, but also usually include parks, lakes and squares, unusual buildings, that are less common for guidebooks. Therefore, they contain a whole variety of areas potentially interesting to a tourist and places where people tend to take photos.

4.2 Time-vs-coverage trade-off

To evaluate how good the recommendation is, we simply count the percent of expert-selected points that fall into the AOIs recommended by the algorithm. We call this percent *"coverage"*. The larger the coverage – the better, as one can discover more interesting places.

[8]http://photos.yandex.ru

On the other hand, the larger the recommended area – the larger the coverage: if the algorithm returns the whole city as a recommendation, it will cover all the expert-selected points. But this algorithm is going to be useless for practical purposes, because we are also interested in the time that one spends exploring the recommended AOIs.

Therefore, we face the trade-off between the coverage and the time required to explore the recommended areas. Figures 3 and 6 show the trade-off pattern for all the seven cities in the dataset. The time required for exploration grows exponentially as the desired coverage increases both for our algorithm and for the baselines.

4.3 Results

Our approach is parameter-free, and therefore, requires no training dataset for tuning. As soon as we get the list of photo coordinates, we can analyze it immediately. For the state-of-the-art DBSCAN and P-DBSCAN baselines we perform parameter selection via cross-validation. In order to apply DBSCAN or P-DBSCAN to a city, we first select the best parameters using the datasets from six other cities, and then use the selected parameters to test the performance.

Some examples of the results are presented in Figures 4 and 5. The results prove that usually the proposed method can identify the points selected by the experts. At the same time it keeps the individual regions small and feasible to walk around within stated 10 minutes, as opposite to DBSCAN method that produces one large region. Figure 5 shows that the recommended regions can be subjectively interpretable, even in cases when expert-selected points do not coincide with the results of the algorithm.

Quantitative results are presented in Figure 6 in the form of time-vs-coverage curves for all the cities for our algorithm (blue curves), for DBSCAN (red curves) and for P-DBSCAN (orange curves). Specific numbers are provided for two fixed values of coverage (60% and 80%) in Table 1 (for DBSCAN) and Table 2 (for P-DBSCAN).

P-DBSCAN is based on a more flexible model than DBSCAN, and is in most cases not worse than DBSCAN, which confirm previous results (see Section 2.2). However, it is not always the case, and most probably happens because P-DBSCAN has more parameters and therefore it is more subject to overfitting.

Our method consistently outperforms the state-of-the-art DBSCAN and P-DBSCAN recommendations. Only in a very few cases P-DBSCAN produces recommendations that require less time to explore than the recommendations of our algorithm for some fixed coverage values: in Odessa and Cherkasy for only some values of coverage. In all these cases the difference is not more than 11%, which is usually negligible.

On the other hand, for another 4 cities our method produces recommendations that require up to 1.5-2 times less exploration time comparing to the both baselines. And, finally, for Volgograd, both DBSCAN and P-DBSCAN fail completely requiring about 10 times more time than the proposed method. That most probably happens because Volgograd stretches along Volga river and DBSCAN (as well as P-DBSCAN) enforces not elongated, but compact clusters.

As we perform areas recommendation, our algorithm naturally performs grouping of POIs into AOI. One AOI released as a recommendation by our algorithm, contains from 0 up to 5 POIs that were selected by local experts depend-

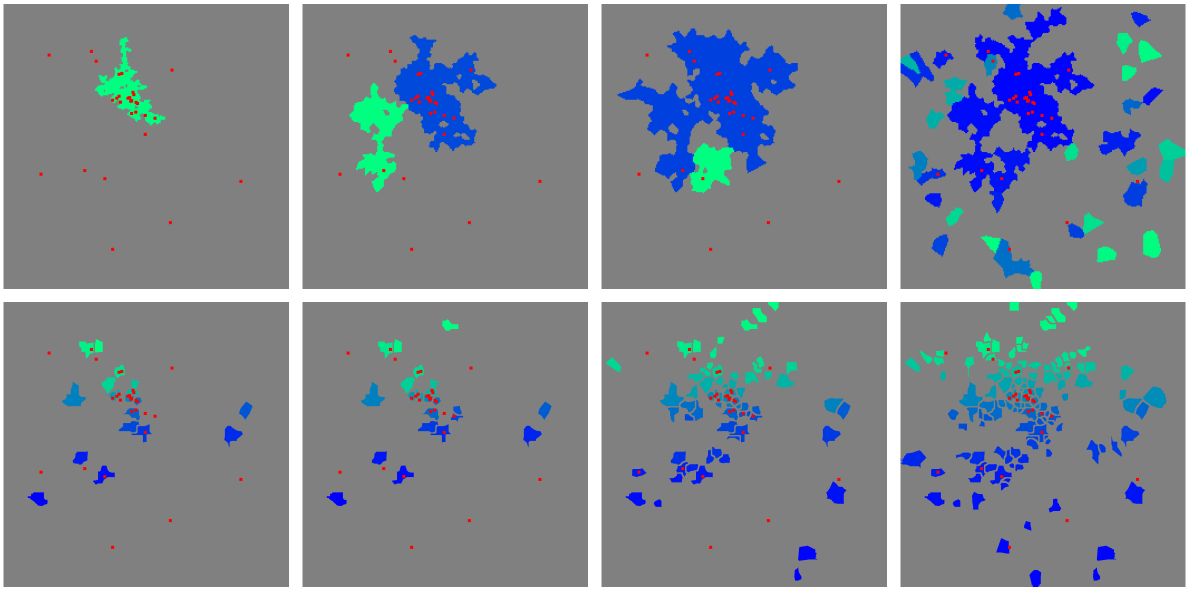

Figure 4: An example of the resulting recommendations from DBSCAN (first row) and our algorithm (second row) for Irkutsk. The points of interest selected by local experts are colored in red. The parameters of DBSCAN in this experiment are selected manually with no cross-validation. Each column from left to right corresponds respectively to a fixed coverage levels of 60%, 70%, 80% and 90% (the percent of red points that fall into the colored regions). The larger the colored area, the longer time is required to explore it. The third column corresponds to 16 hours for DBSCAN and 8 hours for our method.

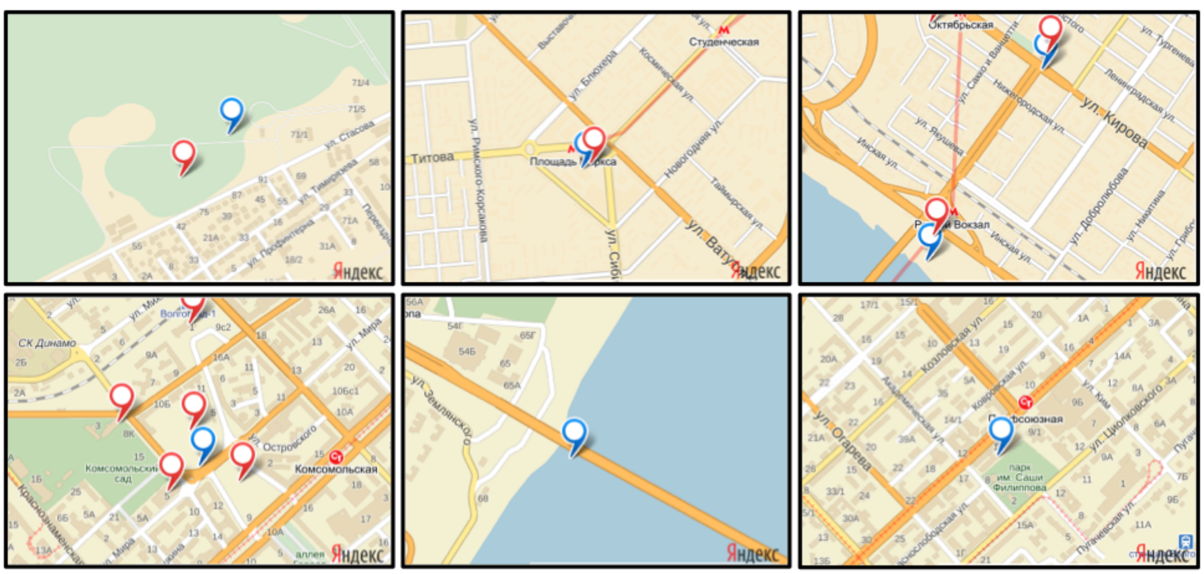

Figure 5: A closer look at the algorithm output projected to a map for different cities: red markers correspond to the POIs selected by experts, blue markers correspond to the geometrical centers of the AOIs recommended by the proposed algorithm. Top row presents direct correspondences found by the algorithm (one POI per AOI), bottom row shows cases with no direct correspondence between experts and the method. In the bottom left subfigure four expert points are joined into one area, as designed by our algorithm (all four POIs are included in one 10 minutes walking AOI). The bottom middle and bottom right subfigures show two examples of AOIs recommended, even though experts did not select any POIs in this area. One could still argue that something interesting is probably located there, as one of the points refers to a park (green area) and another one to a bridge across Volga river. Both areas are dense in terms of photos taken around.

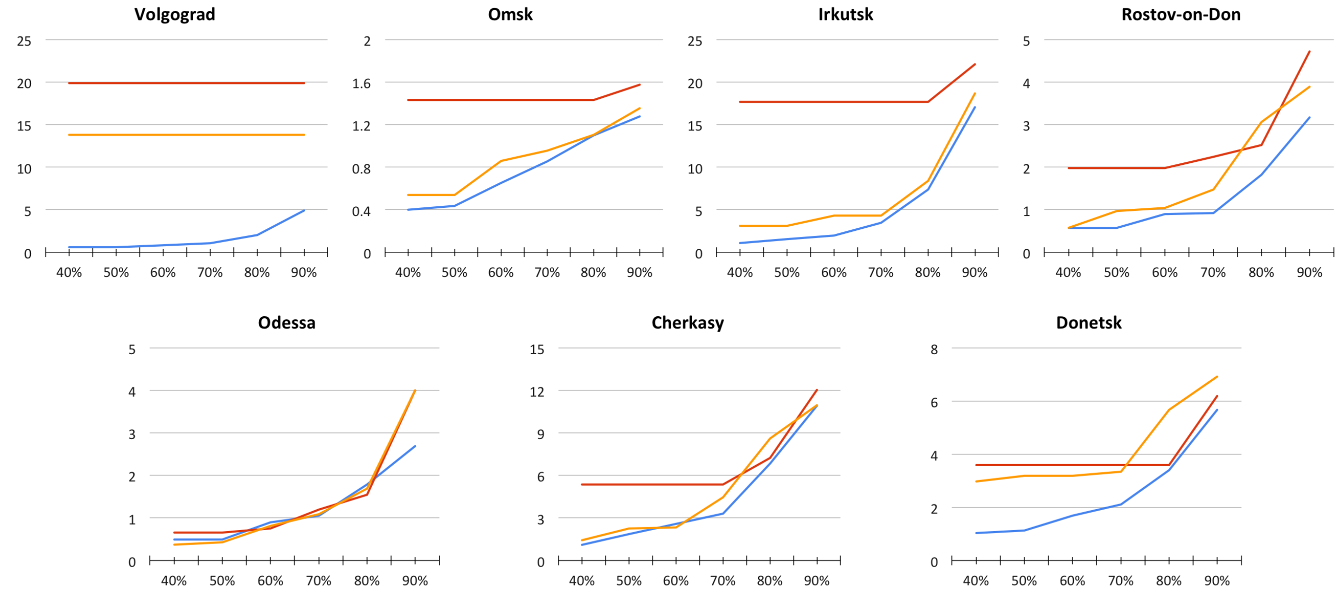

Figure 6: Quantitative comparison of DBSCAN, P-DBSCAN and our method. Vertical axes represent the exploration time (in hours) required for a tourist to cover a fixed percent (coverage) of expert-selected points (horisontal axes). The red line corresponds to the state-of-the-art DBSCAN results, the orange line corresponds to P-DBSCAN, the blue line corresponds to our results. The smaller the time for a fixed value of coverage – the better. As expected, P-DBSCAN often outperforms DBSCAN. Our method is not worse than both DBSCAN and P-DBSCAN in most of the cities, and in many cities significantly outperforms them for every fixed coverage level.

ing on the city. For P-DBSCAN these numbers are similar, as well as the approximate AOI size. DBSCAN, however, usually produces AOI that contain up to tens of POIs (see Figure 4). That happens because of the reasons discussed in Section 2.2.

The tables and the plots can be read, for example, as follows: if one wants to visit approximately 80% of the most interesting places in Rostov-on-Don (according to the expert selection), she needs to spend a little less than 2 hour walking around recommended regions (each region can be explored within at most 10 minutes of intense walking) plus some transportation overhead. 90% of the most interesting places can be explored within about 3 hours, which can be very convenient for a relatively large city (the area of Rostov-on-Don is about 354 square kilometers and the population is about 1.1 million people).

Of course, if one wants to see a larger number of attractive places, she needs to spend more time and to explore more areas recommended. For example, to increase the coverage from 70% to 90% in Irkutsk, 4.5 times more regions need to be explored, and therefore 4.5 times more exploration time spent (4 hours versus 18 hours for 90% coverage).

5. CONCLUSIONS

In this paper we present a method for discovering and recommendation areas of interest based on user-generated geotagged information. Our approach allows us to provide fully unsupervised recommendations in the cities with no expert-collected guidebooks and to incorporate the interests of many users resulting in less subjective conclusions.

The proposed method starts with kernel density estimation of observed location positions and then partitions their distribution with the watershed segmentation algorithm. Multiple partition hypotheses are computed at the same time with different values of the bandwidth parameter, but only one is chosen: the one with the regions of the size suitable for a ten minute stroll. This procedure results in a parameter-free recommendation algorithm. The only parameter that one need to provide for the algorithm is the spare time available. Therefore, the proposed non-parametric algorithm is driven only by individual tourist's requirements.

The main advantages of the proposed method include: more robust and often more relevant recommendations based not on the points, but on the regions; no parameter selection required as parameters are tuned to better respond to high-level tourist goals; computational efficiency.

We test the method on 7 Eastern European cities with different properties and different number of geotagged photos available. The method produces robust recommendations that coincide with the experts opinion on attractive places, while meeting the high-level goal to keep the overall walking time small.

Quantitative experiments show significant increase in recommendation efficiency as compared to the state-of-the-art DBSCAN and P-DBSCAN clustering methods, resulting in up to several times savings in time while exploring the same number of attractive places.

As discussed in Section 2, many different location-based recommendation tasks may start with solving the problem of attractive places discovery. The proposed method can be easily incorporated into their pipelines and therefore can contribute to further increase in their performance.

Table 1: Time required to achieve given coverage for our algorithm and DBSCAN in hours.

City	60% coverage			80% coverage		
	Ours	**DBSCAN**	**Gain**	**Ours**	**DBSCAN**	**Gain**
Volgograd	0.8	19.9	2309%	2	19.9	888%
Omsk	0.7	1.4	120%	1.1	1.4	30%
Irkutsk	2	17.7	801%	7.4	17.7	140%
Rostov-on-Don	0.9	2	122%	1.8	2.5	38%
Odessa	0.9	0.8	-11%	1.8	1.6	-11%
Cherkasy	2.6	5.4	108%	6.9	7.2	6%
Donetsk	1.7	3.6	113%	3.4	3.6	5%

Table 2: Time required to achieve given coverage for our algorithm and P-DBSCAN in hours.

City	60% coverage			80% coverage		
	Ours	**P-DBSCAN**	**Gain**	**Ours**	**P-DBSCAN**	**Gain**
Volgograd	0.8	13.8	1569%	2	13.8	584%
Omsk	0.7	0.9	32%	1.1	1.1	0%
Irkutsk	2	4.3	119%	7.4	8.4	14%
Rostov-on-Don	0.9	1	17%	1.8	3.1	68%
Odessa	0.9	0.8	-9%	1.8	1.7	-5%
Cherkasy	2.6	2.3	-9%	6.9	8.6	25%
Donetsk	1.7	3.2	89%	3.4	5.7	67%

6. REFERENCES

[1] J. Bao, Y. Zheng, D. Wilkie, and M. F. Mokbel. A survey on recommendations in location-based social networks. *ACM Transaction on Intelligent Systems and Technology*, 2013.

[2] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui. Exploring millions of footprints in location sharing services. *International Conference on Weblogs and Social Media ICWSM 2011*, pages 81–88, 2011.

[3] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090, 2011.

[4] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *18th international conference on World Wide Web*, pages 761–770, 2009.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 226–231, 1996.

[6] J. A. Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., 1975.

[7] L. S. Kennedy and M. Naaman. Generating diverse and representative image search results for landmarks. In *17th international conference on World Wide Web*, pages 297–306, 2008.

[8] S. Kisilevich, F. Mansmann, and D. Keim. P-DBSCAN: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. In *1st International Conference and Exhibition on Computing for Geospatial Research & Application*, page 38, 2010.

[9] J. Liu, Z. Huang, L. Chen, H. T. Shen, and Z. Yan. Discovering areas of interest with geo-tagged images and check-ins. In *20th ACM international conference on Multimedia*, pages 589–598, 2012.

[10] X. Lu, C. Wang, J.-M. Yang, Y. Pang, and L. Zhang. Photo2trip: generating travel routes from geo-tagged photos for trip planning. In *18th ACM international conference on Multimedia*, pages 143–152, 2010.

[11] J. Luo, D. Joshi, J. Yu, and A. Gallagher. Geotagging in multimedia and computer vision survey. *Multimedia Tools and Applications*, 51(1):187–211, 2011.

[12] F. Meyer. Topographic distance and watershed lines. *Signal processing*, 38(1):113–125, 1994.

[13] H. J. Nussbaumer. *Fast Fourier transform and convolution algorithms*. 1982.

[14] E. Parzen et al. On estimation of a probability density function and mode. *Annals of mathematical statistics*, 33(3):1065–1076, 1962.

[15] J. Sang, T. Mei, J.-T. Sun, C. Xu, and S. Li. Probabilistic sequential POIs recommendation via check-in data. In *20th International Conference on Advances in Geographic Information Systems*, pages 402–405, 2012.

[16] P. Serdyukov, V. Murdock, and R. Van Zwol. Placing flickr photos on a map. In *32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 484–491, 2009.

[17] M. Shirai, M. Hirota, H. Ishikawa, and S. Yokoyama. A method of area of interest and shooting spot detection using geo-tagged photographs. In *ACM SIGSPATIAL Workshop on Computational Models of Place*, 2013.

[18] Y. Yang, Z. Gong, and L. Hou. Identifying points of interest by self-tuning clustering. In *34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 883–892, 2011.

[19] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *18th international conference on World Wide Web*, pages 791–800, 2009.