

# Identifying Points of Interest by Self-Tuning Clustering

Yiyang Yang<sup>1</sup>

Zhiguo Gong<sup>2</sup>

Leong Hou U<sup>3</sup>

Department of Computer and Information Science  
University of Macau  
Macau

<sup>1</sup>ya97405@umac.mo <sup>2</sup>fstzgg@umac.mo <sup>3</sup>ryanlhu@umac.mo

## ABSTRACT

Deducing trip related information from web-scale datasets has received very large amounts of attention recently. Identifying points of interest (POIs) in geo-tagged photos is one of these problems. The problem can be viewed as a standard clustering problem of partitioning two dimensional objects. In this work, we study spectral clustering which is the first attempt for the POIs identification. However, there is no unified approach to assign the clustering parameters; especially the features of POIs are immensely varying in different metropolitans and locations. To address this, we are intent to study a self-tuning technique which can properly assign the parameters for the clustering needed.

Besides geographical information, web photos inherently store rich information. These information are mutually influenced each others and should be taken into trip related mining tasks. To address this, we study reinforcement which constructs the relationship over multiple sources by iterative learning. At last, we thoroughly demonstrate our findings by web scale datasets collected from Flickr.

## Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Clustering;  
H.2.8 [Database Applications]: Data Mining, Image Databases, Spatial Databases and GIS

## General Terms

Algorithms, Experimentation

## Keywords

Web Images, Spectral Clustering

## 1. INTRODUCTION

In Web 2.0, people are easily to share diverse types of resources with other people. Web album is one example that people can share their photos with others. In many web

album services, people not only share the photos, but also provide relevant information, such as tags, titles, description, taken time, and taken location. Such a large amount of user-contributed data constitutes spatial, temporal, textual, and visual information that can be used for different mining tasks. Deducing trip related information from web photos [3, 5, 7, 12, 13, 14, 19, 20, 21, 22] has been an active topic recently. These works include mapping the photos, identifying places of interest, predicting user travel behavior, and itineraries planning.

In this work, we consider the collection of geo- and time-tagged photos on Flickr. The geographical location was automatically captured by location aware camera, mobile phone, or manually input by some tool in Flickr. Also, the taken time was automatically recorded into the photo metadata. There are currently over 121 million geo-tagged photos on Flickr while there were only 40 million as reported by [13] in 2008. With the advent of more location aware devices, there is no doubt that the number of geo-tagged photos on Flickr and other sites grows rapidly. These involve in substantial research work for web scale mining process.

Our goal in this work is to identify high quality points of interest (POIs) using collections of geo- and time-tagged photos. In particular, point of interest is characterized by activities of photos based on the meta data, such as taken time and geographical location. Some results are recently reported in the community. Kennedy and Naaman [13] identify POIs in a city by applying  $k$ -means clustering on geo-tagged photos. The identified POIs are subsequently ranked by textual and visual features. Crandall et al. [7] study a similar problem on finding POIs in web scale datasets. The authors replace  $k$ -means with mean shift [6] that is a popular clustering approach in image segmentation. Mean shift clustering requires a *density radius* for clustering process instead of specifying the number of clusters in advance. In [7], the authors set the density radius to 100 meters by *subjective observation*. Nevertheless, we argue that such radius setting should not be set subjectively since the sizes of POIs could be immensely varying in different metropolitans and locations. For instance, the sizes of famous POIs, such as *Eiffel Tower*, *Musée du Louvre*, and *Arc de Triomphe*, in Paris are immensely varying as plotted in Figure 1. Besides  $k$ -means and mean shift clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [8] is recently studied for POIs identification in [14]. However, the parameters in DBSCAN are not nature either. The detail of these clustering approaches are discussed in Section 2.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR '11, July 24–28, 2011, Beijing, China.

Copyright 2011 ACM 978-1-4503-0757-4/11/07 ...\$10.00.

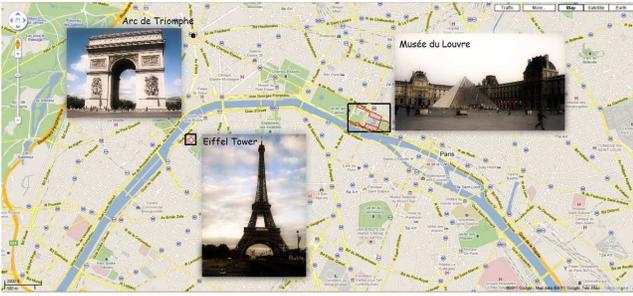


Figure 1: Famous POIs in Paris

Our first mission in this work is to remove *subjective* clustering parameters. Given a set of geo- and time-tagged photos, we propose a self-tuning clustering approach that can identify POIs neither knowing (1) the number of POIs, (2) the size of POIs, nor (3) the shape of POIs in advance. Our self-tuning clustering approach is based on *spectral* clustering [1, 23] which can be viewed as a grouping problem in graph theory. The set of objects in the features space are represented as a weighted undirected graph  $G$ , where the nodes are the objects and the edges are the relation between pairs of objects. In general, the weight on each edge is represented by the closeness in the features space. Spectral clustering then makes use of the spectrum of the graph similarity. It recursively seeks a bipartition of the dataset (i.e., done by various *cut* algorithms [1, 2, 4, 23, 24]) until all subsets fulfill the *termination criteria* (i.e., closeness of a dataset).

However, as discussed in the example (Figure 1), the sizes and densities of POIs are not necessarily identical in a city. There is no unified termination criteria for various POIs in different metropolitans or locations. We observe that the geo-tagged photos are Gaussian distributed in surrounding areas of an POI; and most likely the bi-partitions of the POI's photos follow the same distribution. Furthermore, we find that the costs to decompose a set of Gaussian distributed photos and its sub-sets are resembling. Therefore, if a cluster and its sub-clusters being created by spectral clustering have resembling costs, then the cluster is more likely an POI and should be preserved in the clustering result. The detail of our finding is introduced in Section 3.2.

Besides the parameters issues, the information being taken into the clustering process should be considered thoughtfully. Intuitively, web photos inherently have spatial and temporal information, such as taken location and time. Such information should be thoroughly participated in the clustering process. In this work, we attempt to integrate these information by *reinforcement* [15, 9] which is a well accepted method for analyzing richly structured data. In our reinforcement model, every pair of photos is appraised based on their distance (*spatial*), visit sequence (*temporal*), and region density (*spatial*). The detail of the model is discussed in Section 4.

We summarize our contributions as the following:

1. We attempt to identify POIs using non-subjective parameters. This is significant since the POIs features are immensely varying in different metropolitans and locations and there is no unified approaches to define the parameters appropriately.

2. We connect spatial and temporal information by reinforcement. Other types of information can be integrated into the reinforcement model after specific study.
3. We give a complete survey on POIs identification and thoroughly compare the state-of-the-art to our self-tuning approach by web scale datasets. In addition, we evaluate our reinforcement model and study its behavior in the experiments.

The rest of this paper is organized as follows. We first overview and compare the state of the art POIs identification in Section 2. Next, we discuss spectral clustering and our features similarity measurement in Section 3. Our reinforcement model is introduced in Section 4. It calculates the pairwise relationship of the photos based on their spatial and temporal features. Our approaches are thoroughly evaluated in Section 6 using web scale datasets collected from Flickr. Before we conclude our work and discuss possible future work in Section 8, we also introduce some related work in deducing trip related information in Section 7.

## 2. CLUSTERING TECHNIQUES FOR POIS IDENTIFICATION

Finding POIs from a collection of geo-tagged photos can be viewed as a clustering problem of identifying highly photographed locations. Kennedy and Naaman [13] use  $k$ -means clustering to identify famous locations using collections of geo-tagged photos.  $k$ -means clustering aims to partition  $n$  objects into  $k$  clusters such that each object belongs to the cluster with the nearest mean. The objective function can be formulated in the following equation.

$$\text{minimize } \sum_{1 \leq i \leq k} \sum_{o \in C_i} \|o - \mu_i\|^2 \quad (1)$$

where  $C_i$  is one of the  $k$  clusters and  $\mu_i$  is the centroid of  $C_i$ . Finding the exact solution of  $k$ -means is a NP-hard problem. The most common heuristic algorithm uses an interactive refinement technique. First the algorithm randomly selects  $k$  objects as initial means (i.e., the centroids of  $k$  clusters). Then the clustering proceeds by alternating between the following steps. (1) Every object is assigned to the closest mean. (2) Update the mean of each cluster to the centroid of the objects. The clustering process is terminated when the means no longer change. However,  $k$ -means clustering is a fixed-clustering approach which is problematic for POIs identification since the number and sizes of POIs are not known in advance. There is no appropriate rule to decide the value of  $k$  for different metropolitans and locations.

To address the problem of fixed-clustering approaches, Crandall et al. [7] study mean shift clustering [6] that supports arbitrary size and arbitrary number of clusters. Given a kernel function, mean shift locates the maxima by sampling discrete data from the function. It is a non-parametric feature-space technique since the clustering result depends on a kernel function other than some parameters. Broadly speaking, the mean shift clustering is an iterative process. At each iteration, every object  $o$  moves towards to a mean location where the mean is computed by a kernel function  $f_K$  and the nearby objects  $o_N$ . The kernel function  $f_K$  could be a typical Uniform kernel (i.e.,  $f_U(o_N - o) = 1$ ) or Gaussian

kernel (i.e.,  $f_G(o_N - o) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\|o_N - o\|^2}{2\sigma^2}}$ ). Accordingly, the mean is estimated by

$$\mu = \frac{\sum_{o_N \in N(o)} f_K(o_N - o) o_N}{\sum_{o_N \in N(o)} f_K(o_N - o)} \quad (2)$$

where  $N(o)$  is the neighborhood of  $o$  and  $\mu$  is the new mean estimated by the kernel function. We replace  $o$  by  $\mu$  and iteratively do the mean estimation until  $\mu$  converges to  $o$ . Finally, the objects share the same converge point should be grouped into a cluster.

However, mean shift is not completely parametric free while there is a bandwidth parameter for locating neighborhood (i.e.,  $N(o)$ ). The neighborhood of  $o$  is a set of objects for which  $k$  nearest objects (Uniform kernel) or  $\|o_N - o\| \leq \eta$  (Gaussian kernel), where  $k$  and  $\eta$  are the neighborhood bandwidth. The clustering quality is substantially affected by the bandwidth which can be viewed as the *influential area* of an object. Every object moves towards the mean in the influential area at each iteration.

Both  $k$ -means and mean shift are *mean*-based clustering approaches since they share the same thesis behind. The only difference is that  $k$ -means has fixed number of means; while the number is varying on the kernel function and the corresponding influential area in mean shift. Their differences are demonstrated in Figure 2. Suppose  $k$  is set to 2 and the initial locations of  $\mu_1$  and  $\mu_2$  are shown in Figure 2(a). The changes of  $\mu_1$  and  $\mu_2$  are illustrated by the arcs. Two clusters  $\{a, b, c\}$  and  $\{d, e, f\}$  are formed based on the converge locations of  $\mu_1$  and  $\mu_2$ . In contrast to  $k$ -means, mean shift does not know the number of means in advance. Every object is moved towards the mean according to the objects in their influential area, e.g., the influential area of object  $c$  is illustrated by a dash circle in Figure 2(b). In this example, all objects are converged to  $\mu$  and grouped into one cluster.

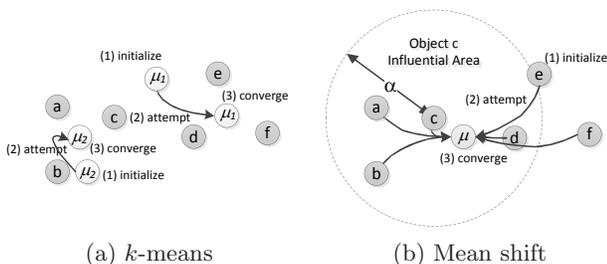


Figure 2:  $k$ -means versus mean shift

Besides mean-based clustering approaches, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [8] is another popular clustering technique. Kisilevich et al. [14] recently analyze places and events in a collection of geo-tagged photos using DBSCAN. Basically, DBSCAN is based on notion of density reachability. An object  $o$  is *directly* density reachable from another object  $o'$  if it is not farther away than a given density radius  $\varepsilon$  and  $o'$  is surrounded more than  $\theta$  objects.  $o$  is called density reachable from  $o''$  if there is a directly density reachable sequence of objects from  $o$  to  $o''$ . Consequently, the objects are grouped into a cluster if they are density reachable.

A common DBSCAN algorithm is demonstrated by Figure 3(a). First, the algorithm randomly selects an object and form a range search with radius  $\varepsilon$ . In this example, object  $c$  is selected and objects  $a, b$ , and  $d$  are discovered by the range search. Suppose  $\theta$  is set to 2, only  $d$  is directly density reachable from  $c$ . Consequently, we form a range search for  $d$  and iteratively discover subsequent density reachable objects. In this example, only one cluster is formed since all objects are density reachable from object  $c$ .

DBSCAN clustering works with generic points having a unified density threshold for all clusters; however, POIs may have varying sizes and density as discussed in Section 1. Kisilevich et al. [14] figure out this problem and propose P-DBSCAN where an adaptive density technique is added into the directly density reachable definition. In P-DBSCAN, an object  $o$  is directly density reachable from another object  $o'$  if it is not farther away than a given density radius  $\varepsilon$  and the ratio of surrounding objects between  $o$  and  $o'$  must be less than a density ratio  $\omega$ .

P-DBSCAN is demonstrated by Figure 3(b). Suppose  $\theta$  and  $\omega$  are set to 2 and 0.3, respectively, object  $c$  is the first selected object to construct a cluster. According to the revised definition, object  $d$  is no longer directly density reachable from  $c$  since the ratio of the surrounding objects between  $c$  and  $d$  is higher than  $\omega$ , i.e.,  $|(2 - 3)/3| \approx 0.33 \geq \omega$ . The cluster constitutes only  $\{a, b, c, d\}$ . Next, the algorithm selects another object (i.e.,  $k$ ) and iteratively find density reachable objects. For instance, object  $h$  is directly density reachable from  $k$ , i.e.,  $|(6 - 5)/5| = 0.2 < \omega$ . Accordingly, objects  $\{g, h, i, j, k, l, m\}$  are grouped into the second cluster. In this example, P-DBSCAN forms better clusters since it takes local density into account. However, the effect of parameters  $\varepsilon$ ,  $\theta$ , and  $\omega$  is never studied for POIs identification.

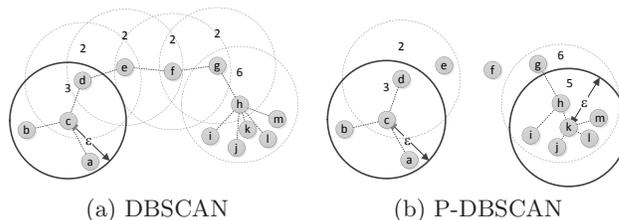


Figure 3: DBSCAN versus P-DBSCAN

### 3. CUT TECHNIQUES FOR POIS IDENTIFICATION

In recent years, spectral clustering [1, 2, 4, 23] has evolved into one of the most common clustering methods and has been widely applied in many applications. However, it has not yet developed for POIs identification. Roughly speaking, spectral clustering is a hierarchical clustering technique which iteratively seeks a bipartition of a set of objects until all sub-partitions fulfill the termination criteria. The bipartition can be computed by various *cut* techniques, such as ratio cut [24], normalized cut [11, 23], and Cheeger cut [1, 2].

In graph theory, a *cut* is to remove some edges from a graph such that the graph is isolated into two subgraphs subject to an objective function. Let  $G = (V, E)$  be an

undirected graph with edge weight  $w_{ij}$  for edges  $e_{ij}$ . For  $V_s, \bar{V}_s \subset V$ , let  $C(V_s, \bar{V}_s)$  be the weight sum of edges between  $V_s$  and  $\bar{V}_s$ , i.e.,  $\sum_{i \in V_s, j \in \bar{V}_s} w_{ij}$ . The objective and complexity of the *cut* techniques are summarized in Table 1.

Table 1: List of *cut* techniques

Name	Objective	Complexity
Ratio cut	$\min_{V_s \subset V} \frac{C(V_s, \bar{V}_s)}{ V_s }$	NP-hard
Normalized cut	$\min_{V_s \subset V} \frac{C(V_s, \bar{V}_s)}{C(V_s, V)} + \frac{C(V_s, \bar{V}_s)}{C(V_s, V)}$	NP-hard
Cheeger cut	$\min_{V_s \subset V} \frac{C(V_s, \bar{V}_s)}{\min\{C(V_s, V), C(V_s, \bar{V}_s)\}}$	NP-hard

The *cut* problem is known to be equivalent to a quadratic discrete optimization problem, *Rayleigh* problem, where the optimization objective is shown in Equation 3.

$$\min_{y^T Q \mathbf{1} = 0, y_i \in \{-b, 1\}} \frac{y^T \mathcal{L} y}{y^T Q y} \quad (3)$$

where  $y$  is a vector having  $n$  variables,  $\mathcal{L} = W^D - W$  is referred as the *Laplacian* of the graph,  $W$  is referred as the weight matrix of the graph,  $W^D$  is a diagonal matrix with  $W_{ii}^D = \sum_{(i,j) \in E} w_{ij}$ , and  $Q$  is a diagonal non-negative matrix which is set differently according to the *cut* technique. For example,  $Q$  is set to  $I$  in ratio cut and is set to  $W^D$  in normalized cut.

In general, *Rayleigh's* optimization is an NP-hard problem. A common approach to solve such optimization is to relax the integral constraints, such as replacing  $y_i \in \{-b, 1\}$  by  $y_i \in [-b, 1]$ . In spectral clustering, the relaxation is done by solving an eigenvector problem which is a looser relaxation than relaxing the integrality requirement. For each *cut* technique, there is a corresponding eigenvector problem for the relaxation. Due to the page limitation, we only introduce the normalized cut in the paper.

Algorithm 1 is a general normalized cut clustering framework. After we construct all necessary matrices in the first two steps, the second smallest eigenvalue  $\lambda$  and the corresponding eigenvector  $\mathcal{V}$  can be computed by Lanczos algorithm [10] in step 3. Note that this is the only step to be replaced if we change the cut technique. Based on the previous study [23], the second smallest eigenvalue  $\lambda$  can be viewed as the minimum *cost* to decompose the graph subject to the corresponding objective function. The graph is decomposed into two if the cost  $\lambda$  is smaller than the threshold  $\delta$ . The decomposition is done by picking a value  $v$  (e.g., medium) in the corresponding eigenvector  $\mathcal{V}$ . The elements in the eigenvector are accordingly split into two groups where the first group contains all elements having better value than  $v$  and the second group contains the remaining elements. Note that the selection of  $v$  is substantial to the decomposition. Thereby, we try multiple  $v$  and pick the best one (i.e., maximize the objective function) such that the decomposition is optimized. Even though spectral clustering is widely used and shown good result in many applications, it does not fulfill the requirement of POIs identification due to the subjective threshold  $\delta$ .

The input of spectral clustering is a weight matrix  $W$ . In general, we can construct  $W$  by Euclidean distances of photos. However, we observe that the photos are not uniformly distributed especially inside an POI. Figure 4(a) and 4(b) plot the photos distribution for *Eiffel Tower* and *Musée du*

---

### Algorithm 1 Spectral Clustering by Normalized Cut

---

- Algorithm**  $SC_{NC}(W)$ : the weight matrix of the graph
- 1: compute diagonal matrix  $W^D$
  - 2: compute Laplacian matrix  $\mathcal{L}$  by  $W$  and  $W^D$
  - 3: **compute the second smallest eigenvalue  $\lambda$  and the corresponding eigenvector  $\mathcal{V}$  of eigenvector problem  $\mathcal{L}y = \lambda Dy$**
  - 4: **if** the eigenvalue  $\lambda$  is smaller than threshold  $\delta$  **then**
  - 5:   decompose the graph into  $W_l$  and  $W_h$  according to  $\mathcal{V}$
  - 6:   call  $SC_{NC}(W_l)$  and  $SC_{NC}(W_h)$
- 

*Lowre*. In these examples, the photos are more likely in Gaussian distribution.

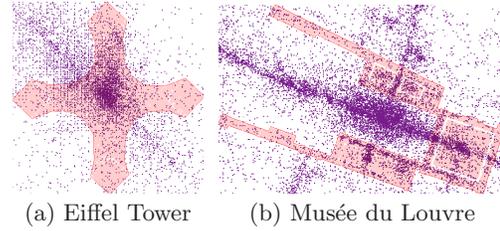


Figure 4: Visualization of photos distribution

We denote the distance relationship as  $D = [D_{ij}]_{n \times n}$ , where  $i$  and  $j$  are two different photos. According to above discussion,  $D_{ij}$  is defined by a Gaussian equation as shown in Equation 4.

$$D_{ij} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\|i-j\|^2}{2\sigma^2}} \quad (4)$$

where  $\sigma^2$  is the variance of the Gaussian function.

### 3.1 Comparison of Variant Clustering

In this section, we demonstrate the parametric issues using a real dataset collected by Flickr API.<sup>1</sup> The dataset is a Paris photos collection having 216,092 geo-tagged photos in total. The detail of our data preparation can be found in Section 6. All figures are generated by our modified version of Java OpenStreetMap Editor<sup>2</sup> which is a map editor for OpenStreetMap<sup>3</sup> written in Java. For the ease of presentation, we highlight the clusters by different colors such that the size and shape of the clusters are clearly illustrated in the figures. Besides, we also plot the minimum bounding rectangles (MBRs) of tourist attractions for reference, where the tourist attractions are collected from the metadata of OpenStreetMap.

The parameters of  $k$ -means, mean shift, and P-DBSCAN are set to their default values as reported in the state-of-the-art [13, 7, 14]. For  $k$ -means, we set  $k$  to 100. For mean shift, we use Gaussian kernel with  $\sigma^2 = 0.2$  and set  $\eta$  (influential area) to  $100m$ . Regarding P-DBSCAN, we set  $\omega$  (density ratio),  $\theta$  (minimum number of objects), and  $\varepsilon$  (density radius) to 0.1, 50, and  $100m$ , respectively. Also, we carefully study the effect of threshold  $\delta$  for spectral clustering and set  $\delta$  to the best value 0.6.

<sup>1</sup><http://www.flickr.com/services/api/>

<sup>2</sup><http://josm.openstreetmap.de/>

<sup>3</sup><http://www.openstreetmap.org/>

The results are shown in Figure 5. Some attraction areas are precisely identified by these clustering approaches. For instance, the world famous POIs Eiffel Tower and Musée du Louvre are identified very well by P-DBSCAN in Figure 5(c) and spectral clustering in Figure 5(d), respectively. By carefully tuning these parameters, the clustering approaches could provide better identification. However, the tuning time might take very long and it is immensely varying in different metropolitans and locations. Besides the default values, we also illustrate other settings in Appendix. In summary, the parameters  $k$  (the number of clusters),  $\eta$  (influential area),  $\varepsilon$  (density radius), and  $\delta$  (cut cost threshold) are quite significant and sensitive to the clustering quality of  $k$ -means, mean shift, DBSCAN, and spectral clustering, respectively.

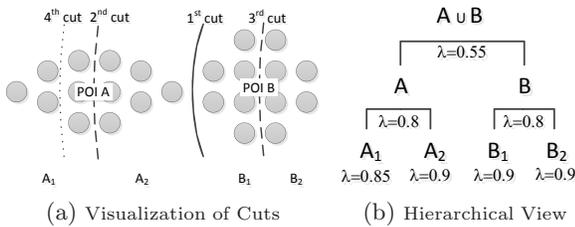
### 3.2 Self-Tuning Spectral Clustering

To address the parameters issue, we develop a self-tuning technique that can eliminate the effect of threshold  $\delta$  from spectral clustering. We attempt to refine spectral clustering since 1) the framework in Algorithm 1 is a hierarchical clustering so that each iteration is trackable and 2) the second smallest eigenvalue  $\lambda$  in each iteration can be viewed as the minimum *cost* to decompose the cluster. We do not have these properties in other clustering methods.

Consider the examples in Figure 4, the photos are Gaussian distributed in the POIs. Based on this observation, we create two sets of POI photos that are Gaussian distributed as shown in Figure 6(a). The photos are partitioned into 4 groups by spectral clustering and the *cuts* are shown by the solid and dash lines. We also show the hierarchical view of spectral clustering procedures in Figure 6(b).

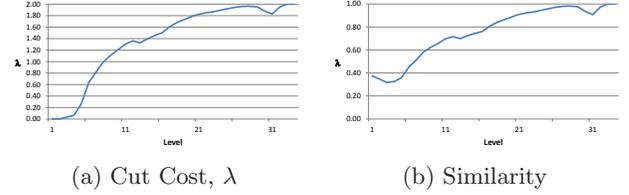
Roughly speaking, the *cut* in spectral clustering seeks a balance decomposition subject to an objective function. Suppose that the solid line is the 1<sup>st</sup> cut in spectral clustering and the cut cost  $\lambda$  is 0.55 and  $\delta$  is set to 0.5, the clustering procedure is terminated and perfectly identifies two POIs  $A$  and  $B$ . However, if  $\delta$  is set to 0.85, then spectral clustering will execute two more cuts such that the photos are decomposed into 4 small clusters,  $A_1$ ,  $A_2$ ,  $B_1$ , and  $B_2$ . It is obvious that  $\delta$  is substantial to spectral clustering. Besides, the cost to decompose  $A$  ( $\lambda=0.8$ ) is similar to the cost to decompose  $A_1$  ( $\lambda=0.85$ ), but different from the cost to decompose  $A \cup B$  ( $\lambda=0.55$ ). It is because  $A$  and  $A_1$  share resembling distribution. We summarize this finding in Observation 1.

**OBSERVATION 1.** *The cut costs of a cluster and its sub-clusters are resembling if the cluster is dense and Gaussian distributed.*



**Figure 6: Example of Spectral Clustering**

We verify the observation using our Paris photos collection. Figure 7(a) shows the cut cost  $\lambda$  at different levels<sup>4</sup> of spectral clustering. The cut cost  $\lambda$  becomes very high ( $>1.8$ ) and resembling after level 21. Also the costs drop a little bit after the peak due to the fluctuation of the cut costs of small clusters.



**Figure 7: Statistics of Spectral Clustering**

According to Observation 1, a cluster should be preserved in the result if the sub-clusters have resembling cut costs. Therefore, we propose a method to measure the costs similarity such that we can preserve the high quality clusters in the result. We first define a new concept, cut path  $p$ , in Definition 1.

**DEFINITION 1 (CUT PATH).** *Given the hierarchical structure of a spectral clustering, a cut path of a cluster  $C$  is the path from  $C$  to a leaf cluster.*

The cost of a cut path  $cost(p)$  is the average cut cost among all nodes in the path which is denoted by Equation 5.

$$cost(p) = \left( \sum_{i \in p} i.\lambda \right) / size(p) \quad (5)$$

In Figure 6, the cost of cut path  $(A, A_1)$  is 0.825. Based on the cost of cut path, we define cut costs similarity in Definition 2.

**DEFINITION 2 (CUT COSTS SIMILARITY).** *Given the hierarchical structure of a spectral clustering, the similarity of a cluster  $C$  is defined in*

$$sim(C) = 1 - \left( \sum_{p \in P} |cost(p) - \mu_C| \right) / size(P) \quad (6)$$

where  $P$  is the set of cut paths of cluster  $C$  and  $\mu_C$  is the mean cost of all cut paths in  $P$ .

In Figure 6, there are two paths from cluster  $A$  to the leaf clusters which include  $p_1 (= A \rightarrow A_1)$  and  $p_2 (= A \rightarrow A_2)$ . The average cost of these paths  $\mu_A$  is  $\frac{cost(p_1) + cost(p_2)}{2} = 0.8375$ . According to the equation in Definition 2, the cut cost similarity  $sim(A)$  is 0.9875.

The value of the cut cost similarity  $sim(C)$  can be viewed as the robustness of all sub-clusters under cluster  $C$ . Our strategy is to group the sub-clusters if  $sim(C)$  is strong. We plot the average value of  $sim$  at different levels in Figure 7(b) as a reference. We observe that the similarity value drops after a local peak and the end levels are not stable due to the fluctuation of the cut costs of small clusters. Therefore, we identify the clusters using a top-down execution paradigm. Given a spectral clustering execution tree, a node is chosen

<sup>4</sup>The level is counted from top to bottom.

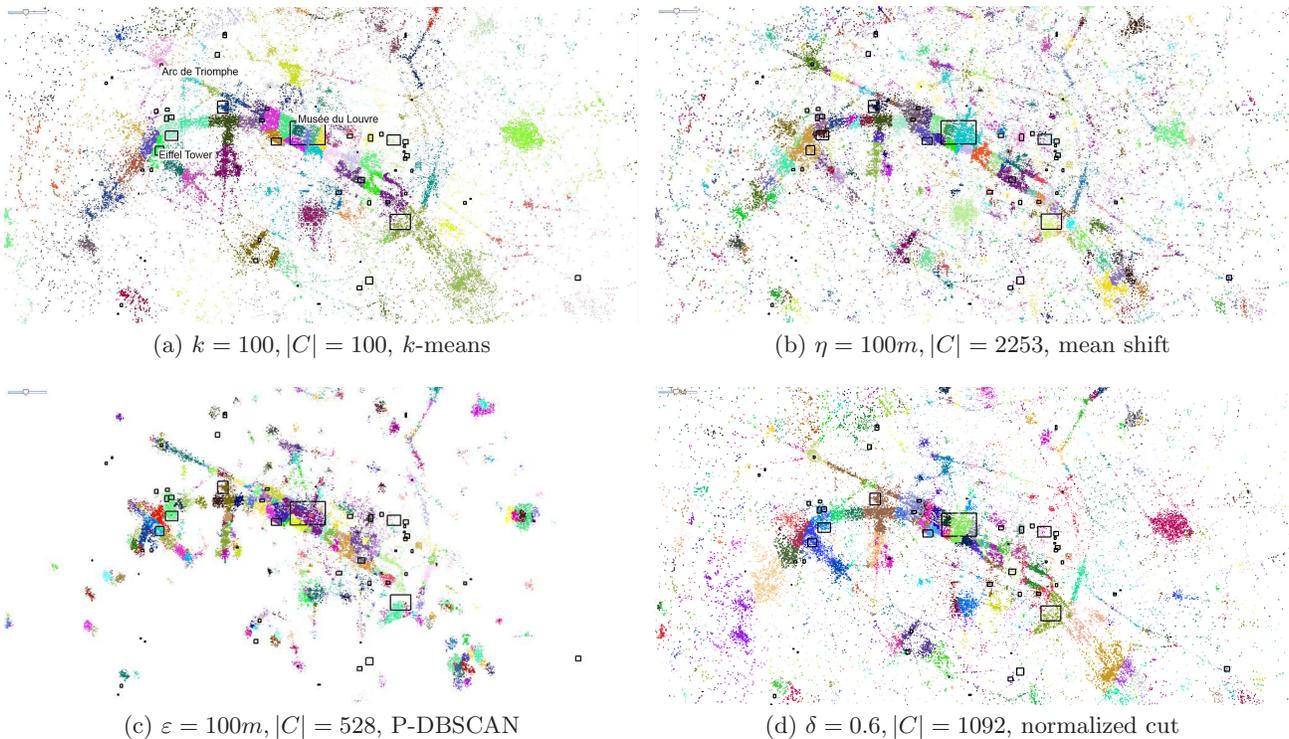


Figure 5: Effect of default parameters in  $k$ -means, mean shift, P-DBSCAN, and normalized cut clustering

as a cluster result  $C$  if  $\text{sim}(C)$  reaches to a good local peak. In this work, we simply set the local peak to the maximum value of the largest decreasing path. In other words, we group highly resembling sub-clusters into a larger cluster based on the similarity of their cut costs. The effectiveness of the self-tuning approach is shown in Section 6.

#### 4. REINFORCEMENT

So far, the clustering approaches only take distance information into account. Apparently, there is room for improvement as web photos constitute rich information in spatial, temporal, textual, and visual aspects. However, there is no explicit guideline to connect these aspects. For POIs identification, the most common method is to perform clustering based on distance and subsequently refine the result by the information from other aspects [7, 13]. In other words, there are quite a lot affluent information not involved in the clustering process. This may affect the clustering quality as some trip related information (i.e., density and visit sequence) are inherently recorded in web photos and are not involved in the clustering process. To be motivated by the discussion, we intend to study *reinforcement* which is a well accepted method for analyzing richly structured data [9, 25].

Consider the example in Figure 8, where we have 16 photos in the system. In Figure 8(a), we might form 2 clusters where the left cluster (in gray color) contains 12 photos and the right cluster (in white color) contains 4 photos if we only take pairwise distances information into account. If the temporal information are taken into account, we might have better result as demonstrated in Figure 8(b). The labels of the photos are represented the movement sequence of

the photographers. For instance, the movement sequence of the photograph  $a$  is represented by four objects  $a_1, a_2, a_3,$  and  $a_4$ . By considering both pairwise distances and movement sequences, we might group left 8 photos into one cluster and group the remaining photos into another cluster as illustrated in Figure 8(b).

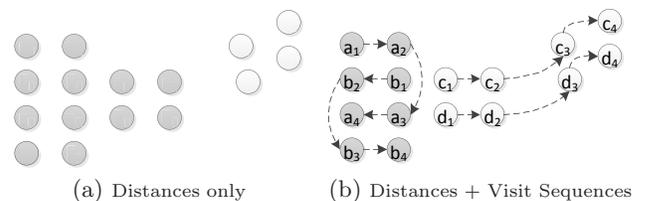


Figure 8: Effect of different trip information

In this work, we only attempt to connect spatial and temporal information by reinforcement. This can be viewed as a study to demonstrate the effectiveness of reinforcement for POIs identification. We add a note that other information (e.g., textual and visual) can be integrated well by reinforcement after specific study. In the following, we study the pairwise relationship construction for density (spatial) and visit sequence (temporal). The reinforcement algorithm will be discussed in next section.

**Density Relationship Construction.** As a crucial part in the density based clustering approaches (e.g., DBSCAN), density relationship should be thoughtfully considered in the

reinforcement. Similar to DBSCAN, the density of a photo is counted by the number of surrounding photos within a range, which is indicated by  $e_i$ . We denote the density relationship as  $E = [E_{ij}]_{n \times n}$ . The value of  $E_{ij}$  represents the density similarity of two photos  $i$  and  $j$ . The formal definition is shown in Equation 7.

$$E_{ij} = \frac{\min\{e_i, e_j\}}{\max\{e_i, e_j\}} \cdot (e_i + e_j) \quad (7)$$

where  $\frac{\min\{e_i, e_j\}}{\max\{e_i, e_j\}}$  is the density similarity of two photos and we multiply it by the sum of their densities due to Observation 2 which is well accepted in density based clustering.

**OBSERVATION 2.** *Highly dense regions are more substantial to POIs identification.*

**Visit Sequence Relationship Construction.** Besides spatial aspect, the movement sequences of photographers are also substantial to POIs identification as discussed in the previous example in Figure 8. We denote the visit sequence relationship as  $S = [S_{ij}]_{n \times n}$ . In general, two consecutive photos are probably taken at the same POI. According to this, we define the visit sequence relationship between two photos in Equation 8.

$$S_{ij} = \begin{cases} 1/seq(i, j) & \text{if } seq(i, j) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where  $seq(i, j)$  is the number of hops between photos  $i$  and  $j$  in the movement sequence of a photographer.  $seq(i, j)$  is set to zero if the photos  $i$  and  $j$  are not taken by the same photographer. For instance, consider the example in Figure 8,  $seq(c_1, c_3)$  and  $seq(a_1, b_1)$  are set to 2 and zero, respectively.

## 4.1 Reinforcement Model

Heretofore, we obtain three feature matrices,  $D$  (distance based),  $E$  (density based), and  $S$  (visit sequence based), from both spatial and temporal aspects. In this section, we describe our iterative reinforcement model that constructs a robust structure between these information. Intuitively, these features mutually affect each others in the POIs identification. Updating the values of one feature will surely influence the values of other features. Consequently, we apply an iterative strategy for obtaining the robust features relationship. More importantly, clustering photos using the robust features potentially provides us better results.

Our reinforcement model is shown in Equation 9. It is a simple iterative model and can easily bias to a feature by tuning  $\alpha$ ,  $\beta$ , and  $\gamma$  accordingly. Initially, we set  $D^{(0)}$ ,  $E^{(0)}$ , and  $S^{(0)}$  to the original feature matrices respectively.  $D^{(n)}$ ,  $E^{(n)}$ , and  $S^{(n)}$  indicate the  $D$ ,  $E$ , and  $S$  matrices at  $n^{th}$  iteration.

$$\begin{cases} D^{(n)} = \alpha D^{(0)} + (1 - \alpha)E^{(n-1)} \\ E^{(n)} = \beta E^{(0)} + (1 - \beta)S^{(n-1)} \\ S^{(n)} = \gamma S^{(0)} + (1 - \gamma)D^{(n-1)} \end{cases} \quad (9)$$

By iteratively executing Equation 9, the mutual influence of the features which include distance, density, and visit sequence is explored by the reinforcement model. Also, we normalize  $D^{(n)}$ ,  $E^{(n)}$ , and  $S^{(n)}$  at the end of each iteration to ensure reasonable reinforcements between them. After  $D^{(n)}$  is convergent to  $D^{(n-1)}$ , we terminate the iterative process and normalize the values of  $D^{(n)}$  by a Gaussian function as

suggested in [25]. The effect of  $\alpha$ ,  $\beta$ , and  $\gamma$  will be evaluated in the experiments section.

## 5. IMPLEMENTATION DETAIL

In this section, we discuss the implementation detail for our approach. Obviously, constructing a complete feature matrix for web scale datasets is too expensive. For instance, we have 216,092 geo-tagged photos in our Paris collection. For the distance matrix  $D$ , it will consume approximate 173.96 GB space if all photo pairs are stored into  $D$ . It is definitely too large for present computer systems.

To address the space issues, we first group the photos by Hilbert curve grouping [16] based on their distances, where the Hilbert curve grouping is a popular multi-dimensional indexing techniques and well accepted by spatial databases. The grouping is shown to preserve very good quality to the original datasets, especially in low dimensionality. In this work, we carefully tune Hilbert curve parameters (e.g., maximum number of objects and grouping size) such that the photos collections are reduced to  $\sim 10\%$  of their original size.

After grouping, the group feature relationship  $X_{IJ}^G$  is set to the average feature relationship of the photos across the groups. This is shown in Equation 10.

$$X_{IJ}^G = \frac{\sum X_{ij}}{|X_{ij}|}, i \in I, j \in J \quad (10)$$

where  $I$  and  $J$  are indicated two Hilbert curve groups of photos, respectively.

Furthermore, we should keep only substantial relations in the feature matrices. There are two reasons: 1) photos are generally grouped together into the cluster with strongest features similarity and 2) the weak features do not involve into the reinforcement process too much after normalization. One of the methods is to remove a distance relationship  $D_{IJ}^G$  from  $D^G$  if  $D_{IJ}^G$  is larger than 200m. The computational cost can be further reduced without much information loss.

## 6. EXPERIMENTS

In this section, we present several experiments to demonstrate the superiority of our findings. We first introduce our data preparation and evaluation measure in Section 6.1 and Section 6.2, respectively. We compare our self-tuning spectral clustering to other clustering approaches in Section 6.3. At last, we investigate the effect of parameters  $\alpha$ ,  $\beta$ , and  $\gamma$  for our reinforcement model.

### 6.1 Data Preparation

We collect three photos collections using Flickr API<sup>5</sup>. For each photo collection, we fetch all geo-tagged photos using the city name as the search key by the Flickr API. We filter out the photos that are not located in the city (by latitude and longitude boundary as shown in Table 2) or have identical geo-location. The statistics of our photos collections are summarized in Table 2.

### 6.2 Evaluation Measure

We extract a set of tourist attractions in the metadata of OpenStreetMap. The total number of attractions for each city is shown in Table 2. The tourist attractions are represented by MBRs as shown in Figure 5. In the evaluation,

<sup>5</sup><http://www.flickr.com/services/api/>

Table 2: List of photos collections

City	No. Photos	No. Clean Photos	Latitude and Longitude	No. MBRs
Paris	788,015	216,092	[48.8151 48.9030 2.2232 2.4742]	92
Hong Kong, HK	191,565	43,294	[22.17596 22.56773 113.79913 114.44252]	28
New York, NY	1,059,209	274,428	[40.49866 40.93115 -74.25797 -73.71277]	62

the MBRs are enlarged to 150% of their original size since photos might be taken in the surrounding area as well. An POI is perfectly identified by a cluster if and only if all photos in the MBR are grouped into the cluster and their sizes are identical. In the evaluation, we filter out the clusters not intersect to any MBRs.

In IR community, a common way to interpretation of clustering is to view it as a series of decisions, one for each of the  $N(N - 1)/2$  pairs of photos in the collection. We classify these decisions by a simple binary classification. For instance, a true positive decision assigns two photos in the same MBR to the same cluster  $C$  and  $TP$  is indicated the total number of true positive decisions. We summarize other types of decisions in Table 3.

Table 3: Binary Classification

	$P$	$N$
$T$	$C, MBR$	$\neg C, \neg MBR$
$F$	$C, \neg MBR$	$\neg C, MBR$

In [17], the precision  $P$ , recall  $R$ , and  $F_\beta$  measure are defined by Equation 11. In this paper, we use  $F_1$  measure as our evaluation measure. A large value of  $F_1$  measure indicates a better clustering.

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad (11)$$

### 6.3 Performance Evaluation

**Self-tuning Spectral Clustering.** In this sub-section, we demonstrate the superiority of our self-tuning technique by Flickr photos collections. Figure 9 shows the effect of the parameter values to the corresponding clustering approach. For each experiment, we vary a single parameter for a clustering approach, while setting the others to their default values as listed in Table 4.

We first compare our self-tuning technique to  $k$ -means, mean-shift, P-DBSCAN, and spectral clustering using Paris photo collection. According to results in Figure 9, it is obvious that our self-tuning technique is superior. Without any parameters tuning, its  $F_1$  measure is only worse than one case in mean shift ( $\eta = 100m$ ) and one case in P-DBSCAN ( $\varepsilon = 50m$ ). Surprisingly, our self-tuning method is better than spectral clustering for all tested parameters. It is because our approach can self-identify the dense region by the cut costs similarity instead of using a global threshold  $\delta$ .

Note that our self-tuning approach might be worse than  $k$ -means, mean shift, P-DBSCAN, or spectral clustering in other parameter settings. However, as demonstrated by Figure 9, the self-tuning approach provides an acceptable quality without manual tuning. As a reference, we illustrate the snapshot of the self-tuning approach in Figure 10. This is

not only valuable to the clustering problem (i.e., POIs identification) but also to other trip related mining problems.

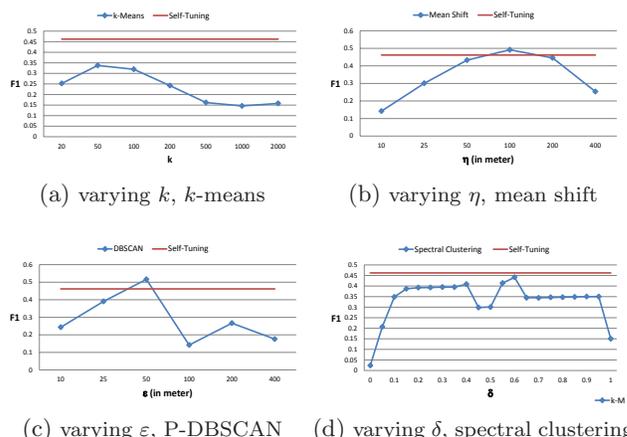


Figure 9:  $F_1$  measures for different clustering

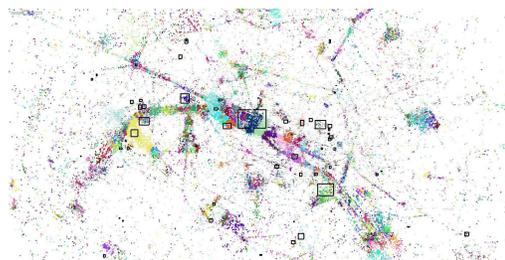


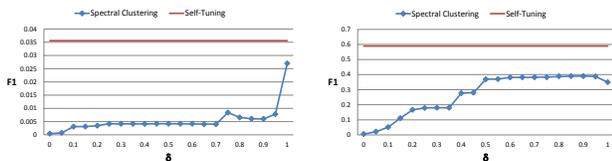
Figure 10: Self-tuning spectral clustering

Figure 11 compares the  $F_1$  measures between spectral clustering and self-tuning approach for Hong Kong and New York photos collections. Again, our self-tuning technique outperforms spectral clustering for all tested  $\delta$  values. The  $F_1$  measures of Hong Kong collection are very small since the tourist attractions provided by OpenStreetMap are very limited and the sizes of attractions are also very small.

**Reinforcement Model.** In Equation 9, we can see that  $\alpha, \beta, \gamma$  are three important parameters in the reinforcement model which control the degree of propagation. In this sub-section, we evaluate their influences to the self-tuning performance. Due to space limitations, we fix  $\alpha$  to 0.9 as the distance information involve the most in the clustering process. The effect of other parameters ( $\beta$  and  $\gamma$ ) is shown in Figure 12. While setting  $\alpha = 0.9, \beta = 0.9,$  and  $\gamma = 0.7,$  the reinforcement model is 27% better than the distance based

Table 4: Range of parameter values

Clustering	Default Settings	Parameter	Values
<i>k</i> -means	-	<i>k</i>	20, 50, 100, 200, 500, 1000, 2000
mean shift	Gaussian kernel, $\sigma^2 = 0.2$	$\eta$ (in meter)	10, 25, 50, 100, 200, 400
P-DBSCAN	$\theta = 50, \omega = 0.1$	$\varepsilon$ (in meter)	10, 25, 50, 100, 200, 400
spectral clustering	-	$\delta$	0, 0.05, 0.1, ..., 0.9, 0.95, 1



(a) spectral clustering, HK (b) spectral clustering, NY

Figure 11: Comparison between spectral clustering and self-tuning for Hong Kong and New York photos collections

model. Besides, our self-tuning technique is only 4.3% (in average) worse than the corresponding best spectral clustering result (by tuning  $\delta$ ) among all testings<sup>6</sup>. It shows the robustness of our self-tuning technique.

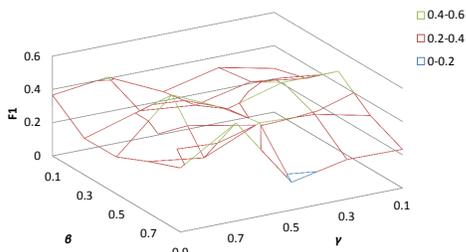


Figure 12:  $F_1$  vs.  $\beta$  and  $\gamma$  (setting  $\alpha = 0.9$ ) on Paris photo collection

## 7. RELATED WORK

Rattenbury et al. [21] is a prior work to discover the names of events and landmarks using geo-tagged and textual data of photos from Flickr. Kennedy and Naaman [13] and Crandall et al. [7] study a problem which can identify landmarks by applying clustering algorithms on geo-tagged photos. The result in [7] is used in many subsequent works being discussed shortly. The authors also propose a classification method to identify the taken location of a photo based on both visual and textual features. However, these works do not consider temporal information to improve the clustering quality.

Besides identifying POIs, there are many recent works that deduce trip information from web photos. Clements et al. [5] study a work that predicts travel interests for a user who had rich travel data in the past. Popescu and Grefenstette [19] attempt to deduce the typical visit duration for tourist attractions. The authors first study a heuristic filtering procedures to clean the photo collection. Then, an

<sup>6</sup>The details are omitted due to the space limitations.

external coverage geographical gazetteer is used to map between the photos and POIs based on their textual and geo-tagged information. Other trip related information are extracted based on their result, such as maximum, minimum, and average duration time of the POIs. A similar work [20] is proposed by the same authors that extracts some day-tour information, such as people visit interests, visit time, and duration time. Their solution first identifies POIs by an external knowledge base, Wikipedia and then extracts the day-tour information based on the identified POIs.

Automatic tour planning from geo-tagged photos has been recently studied in [3, 12]. Choudhury et al. [3] constructs intra-city travel itineraries using spatio-temporal data from Flickr. Their solution first identifies a set of time paths and each time path is a sequence of POIs traversed by a user. The duration time and the transit time of POIs are subsequently extracted by the time paths. Antourage [12] automatically constructs tourist trip from the geo-tagged photos, by specifying a start location and maximum distance covered by the trip. A hexagonal grid overlay is used to map landmarks of a city. Each hexagonal cell is weighted by the number of photos taken inside. Therefore, the tour planning problem in their work becomes an optimization problem that maximizes the total weight of the selected cells where the total distance cannot exceed the given constraint.

Regarding self-tuning clustering, there are some results reported in machine learning [18, 26]. These methods aim at finding the best  $k$  in  $k$ -means by statistical test or eigenvectors analysis. These algorithms typically determine all clusters at once which is different from our hierarchical clustering technique.

## 8. CONCLUSIONS

In this paper, we study spectral clustering which is the first attempt for POIs identification. In addition, we deeply compare the state of the art approaches and figure out the parametric issues. Therefore, we analyze spectral clustering procedures and propose a self-tuning approach based on the cut cost similarity so that the effect of the parameters is eliminated from spectral clustering. Furthermore, we study reinforcement that connects information from diverse aspects by iterative learning.

In the future works, we are intent to study the POIs ranking problems. In general, the POIs are ranked by simple observation (i.e., number of photos, number of users, or size). There is so much room for improvement. In addition, identifying POIs can be viewed as a module of other trip related mining tasks. We will investigate how these related tasks can get improved by our clustering framework. Moreover, we are highly interested in comparing different self-tuning clustering techniques for web scale photos collections.

## Acknowledgments

This work was partially sponsored by Grant RG066/09-10S/11R/GZG/FST from University of Macau Research Committee.

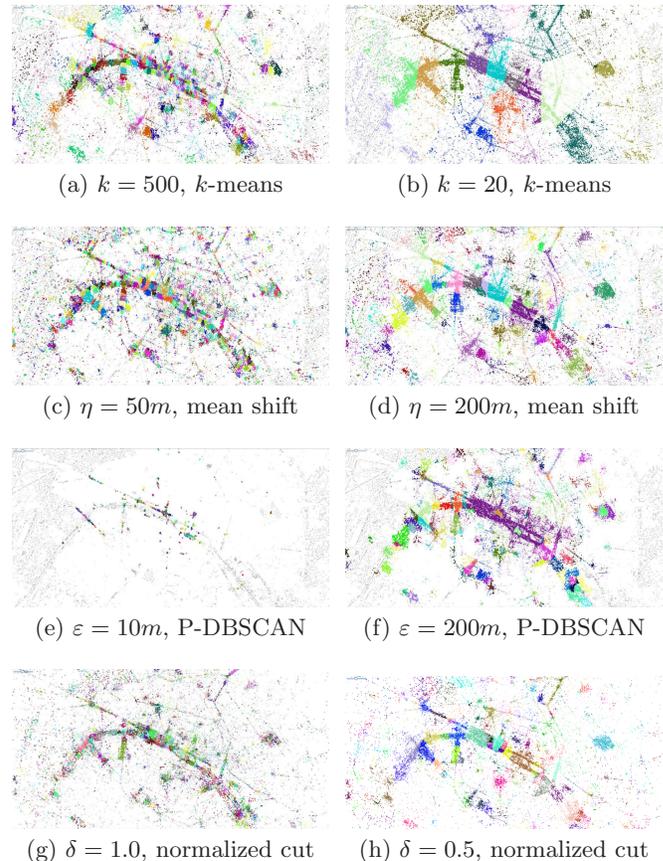
## 9. REFERENCES

- [1] T. Bühler and M. Hein. Spectral clustering based on the graph  $\rho$ -laplacian. In *ICML*, page 11, 2009.
- [2] J. Cheeger. A lower bound for the smallest eigenvalue of the laplacian. *Problems in Analysis*, page 195, 1970.
- [3] M. D. Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. Constructing travel itineraries from tagged geo-temporal breadcrumbs. In *WWW*, pages 1083–1084, 2010.
- [4] F. R. K. Chung. Spectral graph theory. *Regional Conference Series in Mathematics, American Mathematical Society*, 92:1–212, 1997.
- [5] M. Clements, P. Serdyukov, A. P. de Vries, and M. J. T. Reinders. Using flickr geotags to predict user travel behaviour. In *SIGIR*, pages 851–852, 2010.
- [6] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.
- [7] D. J. Crandall, L. Backstrom, D. P. Huttenlocher, and J. M. Kleinberg. Mapping the world’s photos. In *WWW*, pages 761–770, 2009.
- [8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.
- [9] L. Getoor. Link mining: a new data mining challenge. *SIGKDD Explorations*, 5(1):84–89, 2003.
- [10] G. H. Golub and C. F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.
- [11] D. S. Hochbaum. Polynomial time algorithms for ratio regions and a variant of normalized cut. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(5):889–898, 2010.
- [12] S. Jain, S. Seufert, and S. J. Bedathur. Antourage: mining distance-constrained trips from flickr. In *WWW*, pages 1121–1122, 2010.
- [13] L. S. Kennedy and M. Naaman. Generating diverse and representative image search results for landmarks. In *WWW*, pages 297–306, 2008.
- [14] S. Kisilevich, F. Mansmann, and D. A. Keim. P-dbscan: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. In *COM.Geo*, 2010.
- [15] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.
- [16] J. K. Lawder and P. J. H. King. Querying multi-dimensional data indexed using the hilbert space-filling curve. *SIGMOD Record*, 30(1):19–24, 2001.
- [17] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
- [18] D. Pelleg and A. W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, pages 727–734, 2000.
- [19] A. Popescu and G. Grefenstette. Deducing trip related

information from flickr. In *WWW*, pages 1183–1184, 2009.

- [20] A. Popescu, G. Grefenstette, and P.-A. Moëllic. Mining tourist information from user-supplied collections. In *CIKM*, pages 1713–1716, 2009.
- [21] T. Rattenbury, N. Good, and M. Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *SIGIR*, pages 103–110, 2007.
- [22] P. Serdyukov, V. Murdock, and R. van Zwol. Placing flickr photos on a map. In *SIGIR*, pages 484–491, 2009.
- [23] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.
- [24] S. Wang and J. M. Siskind. Image segmentation with ratio cut. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(6):675–690, 2003.
- [25] X.-J. Wang, W.-Y. Ma, L. Z. 0001, and X. Li. Iteratively clustering web images based on link and attribute reinforcements. In *ACM Multimedia*, pages 122–131, 2005.
- [26] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *NIPS*, 2004.

## APPENDIX



**Figure 13: Effect of other parameters in  $k$ -means, mean shift, P-DBSCAN, and normalized cut clustering**