Presenting Diverse Location Views with Real-time Near-duplicate Photo Elimination

Jiajun Liu[1] Zi Huang[1] Hong Cheng[2] Yueguo Chen[3] Heng Tao Shen[1] Yanchun Zhang[4]

[1]*School of Information Technology and Electrical Engineering, The University of Queensland, Australia*
[2]*Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong*
[3]*Key Laboratory of Data Engineering and Knowledge Engineering, Ministry of Education, Renmin University of China*
[4]*School of Engineering and Science, Victoria University, Australia*
{jiajun,huang,shenht}@itee.uq.edu.au,hcheng@se.cuhk.edu.hk,chenyueguo@ruc.edu.cn

***Abstract*—Supported by the technical advances and the commercial success of GPS-enabled mobile devices, geo-tagged photos have drawn plenteous attention in research community. The explosive growth of geo-tagged photos enables many large-scale applications, such as location-based photo browsing, landmark recognition, etc. Meanwhile, as the number of geo-tagged photos continues to climb, new challenges are brought to various applications. The existence of massive near-duplicate geo-tagged photos jeopardizes the effective presentation for the above applications. A new dimension in the search and presentation of geo-tagged photos is urgently demanded. In this paper, we devise a location visualization framework to efficiently retrieve and present diverse views captured within a local proximity. Novel photos, in terms of capture locations and visual content, are identified and returned in response to a query location for diverse visualization. For real-time response and good scalability, a new Hybrid Index structure which integrates R-tree and Geographic Grid is proposed to quickly identify the Maximal Near-duplicate Photo Groups (MNPG) in the query proximity. The most novel photos from different groups are then returned to generate diverse views on the location. Extensive experiments on synthetic and real-life photo datasets prove the novelty and efficiency of our methods.**

I. Introduction

The emergence and evolution of GPS-enhanced devices have created a whole new dimension of existing multimedia applications in the past decade [21]. The geo-information encoded in the multimedia data, images and videos in most cases [7], [5], [1], [21], facilitates various geo-tag driven applications. The geo-locations recorded in the photos offer a unique cue for photo browsing [7], [11], photo annotating and tagging [4], [15], landmark recognition and visualization [5], [18], [17], [19], [27], etc. For instance, Panoramio[1] offers a sophisticated interface of managing and browsing geo-tagged photos in an overlaid pattern with satellite images. A user can browse photos in a multi-level way on the whole map, or just search photos for a location. Its service has been integrated to Google Earth[2] seamlessly. Meanwhile, Flickr has also empowered its conventional browsing features with geo-tags, where similar browsing and searching interfaces are provided. Additionally, the trajectory of a traveler can be extracted from a group of his geo-tagged photos, preserving valuable information about how people travel such as locations, visiting time, and the views of places. These information grants ample possibility of mining useful travel patterns to assist the others [3], [8], [10].

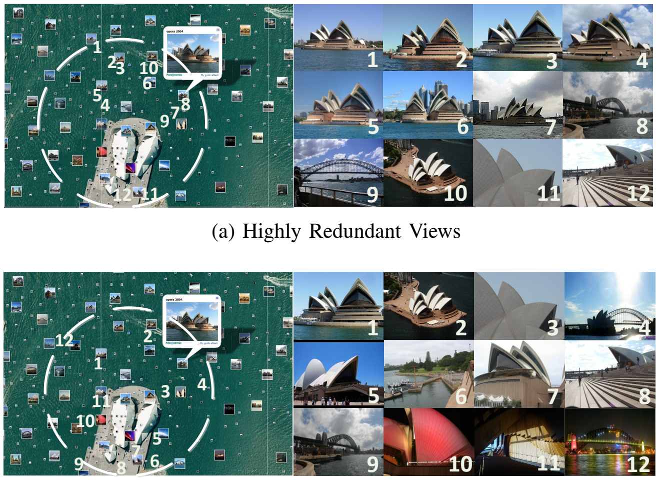
Among those geo-tagged photo applications, one that has drawn less attention is location visualization. The goal of it is to provide as much and as novel visual information as possible to visually describe the environment at and surrounding a certain location. Not only the objects at the exact location are considered relevant, but also are the views of interests surrounding it. That means, the photo can show either nearby objects around the location, or faraway scenes taken at the location due to the zoom lens. The StreetView built on top of Google Earth is one of the popular location visualization services in the real world. However, it depends heavily on the operation of particularly designed cars with GPS and 360° full-view cameras to collect the views. Naturally the locations that can be visualized are quite limited due to the road boundaries and privacy concerns in some countries and particular regions. On the other hand, current services present a simple scheme of visualizing a location by listing the user-uploaded photos captured within the proximity. As the number of uploaded photos continues to grow dramatically, this simple scheme begins to be increasingly inadequate. The overwhelming existence of near-duplicate photos around the same location makes the visual content for a location highly redundant. Diverse views around a location which can also be of great interests are often discarded.

Figure 1 shows a real example of the location visualization interface around the Sydney Opera House. For each sub figure, the left half is the browsing interface and the right half is the visualization interface. The red circle defines the range for preliminary image search in the proximity. Unlike landmark visualization, location visualization is supposed to display various objects and views that can be seen around the query location. For example, as shown in Figure 1a, the displayed photos show a convergence on their visual contents, i.e., most of the photos, even taken at slightly different geo-coordinates, are of the same object despite the fact that there also exist many photos for other interesting scenes, e.g., the sidewalk around the Opera House, the ocean view from near it, the skyscrapers of the city, etc. Excessive near-duplicates

[1] www.panoramio.com

[2] earth.google.com

978-1-4673-4910-9/13/$31.00 © 2013 IEEE



(a) Highly Redundant Views

(b) Diverse Views

Fig. 1: Effect of Diverse Location Visualization

greatly degrade the novelty of displayed photos and bring new challenges on how to effectively present diverse views to a user for location visualization. Figure 1b demonstrates much better visual novelty of the views around the query location. Given large-scale geo-tagged photo datasets with numerous near-duplicates, how to achieve this diversity efficiently for arbitrary queries becomes a great challenge.

Crucial changes and enhancements need to be added to the existing location visualization frameworks to achieve such diverse views, and at the mean time maintain satisfactory efficiency for real-life applications. Consequently, in this paper, as we propose a novel diverse location visualization framework, we make the following contributions. Firstly, to get a clear formulation of the problem, in Section III we introduce the definitions of Geographic-Visual Distance (GVD), Maximal Near-duplicate Photo Group (MNPG), Seed Photo and Novelty. We use these definitions to formulate the diverse location visualization problem. Then we devise a basic framework to tackle this problem in Section IV, where its main steps are discussed. In Section V we propose a Hybrid Indexing structure and explore how to greatly accelerate our framework with various lower-bound based pruning rules. Extensive experiments are conducted and discussed in Section VI to verify the performance of our framework. Finally we conclude the paper in Section VII.

II. Related Work

Generally, diverse location visualization has not been thoroughly studied by the research community, comparing to other geo-tag driven applications like photo browsing [7], [11], landmark visualization [5], [16], [17], tag inferring [15], etc. Landmark visualization is similar to location visualization in some aspects, but in fact they are very different applications. The deviation is in several perspectives. The foremost one is that landmark visualization aims to depict a single object, e.g. an architect or a mountain, from different view angles. However, location visualization targets at providing comprehensive visual information at the query proximity. The visual information can describe objects within the proximity, or scenes that can be viewed from the proximity. Nevertheless, the literature regarding location visualization remains lacking.

In [5], an interesting geo-tagged photo mining system is designed to generate a tourist map with the icons for landmarks. It uses geo-tags and user tags of photos to cluster photos. Content similarity is not considered. [16] proposes a k-mean based method to classify the images for the landmark search results. It uses k-means to cluster the photo set by their visual content and locations, then ranks the clusters and their representative photos. In video re-ranking, a near-duplicate graph-based method is invented to identify cohesive clusters and then pick representative videos from clusters [12]. However, the efficiency issue becomes the major concern for real-life applications which deal with large-scale datasets and have high requirement of interactive speed of visualization [20].

It is natural and necessary to include indexing techniques in our work when efficiency is of high priority. There exist many efficient and sophisticated structures to index spatial data and high-dimensional data. R-Tree [9] has been the most popular spatial index in the spatial database community. Other well-known methods include Quadtree, kd-tree, M-tree [6], etc. For high-dimensional indexing, in recent years, one-dimensional transformation [24], hashing-based methods (e.g., LSB-Tree [25]), and data approximation (e.g., data co-reduction [13]) have drawn much attention. An interesting index structure, which indexes text with geographic reference, is also defined in [28]. However, none of these indexing methods are able to provide the ability of quickly and dynamically identifying Near-duplicate Photo Groups within a query proximity from a large-scale geo-tagged photo database.

III. Definitions

In this section we present the necessary definitions to support the presentation of our work. Table I lists some notations used throughout the paper.

Notation	Description
v	the visual feature of a photo
r	the geographic search range for the query
d_g, d_v	the geographic and visual distances
λ	the weight of geographic distance
ϵ	the GVD threshold
$l = \{l_x, l_y\}$	the location of a photo
$\mathcal{P} = \{p_1, ...\}$	the local photo set
$\mathcal{G} = \{g_1, ...\}$	the MNPG set
$\mathcal{S} = \{s_1, ...\}$	the seed photo set
θ	the importance of the seed photo
ϕ	the uniqueness of the seed photo

TABLE I: Notations

Definition 1 (Geographic Distance): Given two photos p_i and p_j, the geographic distance between them is defined as:

$$d_g(p_i, p_j) = \sqrt{(l_{xi} - l_{xj})^2 + (l_{yi} - l_{yj})^2} \quad (1)$$

where l_x is the longitude and l_y is the latitude of the location of a photo.

Definition 2 (Visual Distance): Given two photos p_i and p_j, the visual distance between them is defined as:

$$d_v(p_i, p_j) = f(v_i, v_j), \tag{2}$$

where v_i and v_j are the visual features for p_i and p_j respectively. $f(*, *)$ is a metric to assess the visual distance between two photos, which varies according to the type of feature used. For instance, L_2 distance.

Definition 3 (Local Photo Set): Given a database of geotagged photos, a query location q, and a geographic range r that describes the location proximity around q, a local photo set corresponding to q, denoted as $\mathcal{P} = \{p_1, p_2, ...\}$, is composed of photos that satisfy

$$\forall p_i \in \mathcal{P},\ d_g(p_i, q) \leq r. \tag{3}$$

Here d_g computes the geographic distance between p_i and the query location q.

Note that geographic distance and visual distance are not directly comparable. In order to integrate them into a single value for all the photos in the local photo set, they have to be normalized properly into the range of [0,1]. For geographic distance, we normalize it by the maximal geographic distance $d_g.max$ among indexed photos, i.e., $d_g = \frac{d_g}{d_g.max}$. For visual distance, we normalize it by the maximal visual distance among all the photos, i.e., $d_v = \frac{d_v}{d_v.max}$.

Given the normalized geographic distance and visual distance, we define the Geographic-Visual Distance (GVD) as:

Definition 4 (GVD): Given two photos p_i and p_j, the GVD between them is defined as:

$$gvd(p_i, p_j) = \lambda d_g + (1 - \lambda) d_v \tag{4}$$

where λ specifies the weight of d_g in the fusion.

Definition 5 (Near-duplicate Photo Group): Given a local photo set $\mathcal{P}$ and a GVD threshold ϵ, a Near-duplicate Photo Group (NPG) $g_k = \{p_{k1}, p_{k2} ...\} \subseteq \mathcal{P}$ satisfies

$$\forall p_{ki} \in g_k,\ gvd(p_{ki}, c_k) \leq \epsilon \tag{5}$$

where c_k is the centroid of g_k, which is defined as the mean of all the photos in the group, along both geographic and visual dimensions.

Definition 6 (Maximal Near-duplicate Photo Group): A Near-duplicate Photo Group (NPG) $g_k = \{p_{k1}, p_{k2} ...\} \subseteq \mathcal{P}$ is a Maximal Near-duplicate Photo Group (MNPG) if there does not exist a superset of g_k which is also an NPG.

Definition 7 (Seed Photo): Given an MNPG $g_k = \{p_{k1}, p_{k2}, ...\}$, its seed photo is the photo that has the smallest GVD to its centroid c_k.

Definition 8 (Seed Photo Set): Given a MNPG set $\mathcal{G} = \{g_1, g_2, ...\}$, all the seed photos from each MNPG form the the seed photo set, denoted as

$$\mathcal{S} = \{s_1, s_2, ...\} \tag{6}$$

where s_i is the seed photo of g_i.

A seed photo is regarded as the most representative photo for an MNPG. Among all the seed photos, each can be further measured by its *novelty* score, which is used to rank seed photos so as to select the desired number of most novel photos for location visualization.

Definition 9 (Novelty): Given a local photo set $\mathcal{P} = \{p_1, p_2, ..., \}$ corresponding to a location query q, assume that the photos have been organized into the MNPG set $\mathcal{G} = \{g_1, g_2, ...\}$ and the seed photo set is identified as $\mathcal{S} = \{s_1, s_2, ...\}$. For each seed photo s_i, its *novelty* is determined by its *importance* θ and *uniqueness* ϕ, as:

$$\eta(s_i) = \theta(s_i)\phi(s_i); \tag{7}$$

where importance θ and uniqueness ϕ are defined as:

$$\begin{aligned} \theta(s_i) &= \frac{d_g(s_i, q)|g_i| e^{-\frac{\sum_{p_{ij} \in g_i} gvd(p_{ij}, c_i)}{|g_i|}}}{r|g_{max}|} \\ \phi(s_i) &= 1 - e^{-\frac{\sum_{s_j \in \mathcal{S}}^{j \neq i} gvd(s_i, s_j)}{|\mathcal{S}| - 1}} \end{aligned} \tag{8}$$

where $|*|$ is the size of the set and g_{max} is the largest MNPG.

The importance θ reflects how significant the photo is by considering three aspects: the group's homogeneity $e^{-\frac{\sum_{p_{ij} \in g_i} stvd(p_{ij}, c_k)}{|g_i|}}$, the relative group size $\frac{|g_i|}{|g_{max}|}$, and the relative distance to the query location $\frac{d_g(s_i, q)}{r}$. The uniqueness ϕ describes how distinctive a seed photo is in the seed photo set by calculating the average distance to other seed photos in the seed photo set. Clearly the values of θ and ϕ fall into the ranges of $(0, 1]$ and $[0, 1)$ respectively, so the range of η is $[0, 1)$, which describes the novelty of a photo and can be compared directly with each other.

IV. Diverse Location Visualization

A. Framework

Diverse location visualization requires the identification of most novel photos from a set of photos. In the order of processing, our framework to tackle this problem involves the following major steps, given a query location:

1) Retrieving the subset of photos, taken in the proximity that have smaller geographic distances than the geographic search range to the query location, i.e., the local photo set.
2) Discovering all the MNPGs and their corresponding seed photos in the local photo set.
3) Ranking all the seed photos based on their novelty values and return the most novel photos according to the user-defined number of returned photos.

With the definitions in the previous section, the problem can now be formalized as follows. Given a database of geotagged photos and a query location q, retrieve the local photo set $\mathcal{P} = \{p_1, p_2, ...\}$, discover the MNPG set $\mathcal{G} = \{g_1, g_2, ...\}$ in $\mathcal{P}$ and its seed photo set $\mathcal{S} = \{s_1, s_2, ...\}$, and subsequently return the top-*k* most novel seed photos.

As the objective is to present diverse photos in terms of geographic and visual variations, two types of features are needed. For geographic feature, we simply use the geo-tag in a photo to describe its location. As for visual feature, there are certain requirements that make it different from geographic feature. Firstly, the visual feature should be distinctive so that irrelevant photos can be filtered quickly. Secondly, the feature should tolerate considerable changes in scale and rotation, since people do not always take photos on the same object at the exact same location with the exact same scale and direction. Ideally the feature should be able to identify those photos for the same object from different view angles and in different zooming scales, while at the same time keep out irrelevant photos from being further processed. Thirdly, unlike geographic feature, visual feature requires much more complex data to preserve the content. In the computation of GVD between photos, the most time-consuming part is caused by the visual distance computation. The dimensionality of the visual feature will greatly affect the performance of the visualization. Lately in multimedia applications, the Bag-of-Word (BoW) model from detected scale-rotation invariant local keypoints has become very popular for its tolerance to scale-rotation changes, moderately high dimensionality, and descriptive power for local objects [14]. In our work, it is very natural to use this model to describe the visual content.

In our framework, step 1 and 2 are most time-consuming and critical. In the following subsections, we detail the solution for each step.

B. Retrieving $\mathcal{P}$

To visualize a certain location, naturally only the photos taken within the local proximity need to be considered. When a place somewhere in Paris is requested to be visualized, it makes no sense to include photos taken in London. Geographic closeness describes an essential discipline for choosing photos for location visualization. In such a case, the geo-tag of a photo, i.e., the latitude and longitude of the location, can be used to compute the geographic distance among photos. Given the high efficiency of R-tree in managing 2-dimensional data, R-Tree is used to index the geo-tags of photos in this work. Given a query location q and a geographic search range r, all the photos taken within the local proximity specified by q and r can be quickly retrieved by performing an efficient range search in R-tree. The results form the local photo set $\mathcal{P}$ with respect to q.

C. Discovering $\mathcal{G}$ and $\mathcal{S}$

Given a local photo set $\mathcal{P}$, it is known that finding the MNPG set $\mathcal{G}$ according to Definition 6 is NP-hard [22]. However, the properties that an NPG has provide us with other feasibilities to tackle the MNPG identification problem. Here by adopting the definition used in [22], we utilize a useful property called *Maximally Expanded-by-one* to define Maximally-expanded NPG.

Definition 10 (Maximally-expanded NPG): For an NPG $g_k = \{p_{k1}, p_{k2}, ...p_{kn}\}$, if there exists a pair of photos p_{k1} and p_{k2} that can establish one permutation of the NPGs which induces the group sequence $(\{p_{k1}, p_{k2}\}, \{p_{k1}, p_{k2}, p_{k3}\}, ..., \{p_{k1}, p_{k2}, p_{k3}, ..., p_{kn-2}\}, \{p_{k1}, p_{k2}, p_{k3}, ..., p_{kn-1}\}, \{p_{k1}, p_{k2}, p_{k3}, ..., p_{kn}\})$, such that all groups in the sequence are NPGs, then the NPG sequence is called expanded-by-one. If g_k can not be further expanded by other photos in the local photo set, then it is maximally expanded-by-one and g_k is called Maximally-expanded NPG.

The intuition behind this definition is that having an NPG that is started by a pair of near-duplicate photos, when it is iteratively expanded with the constraint in Definition 5, all the resulting groups satisfy Definition 5 and are hence NPGs. When there exists no more photos in the local photo set that can be added to the current NPG without breaking the NPG constraint, we can consider that an MNPG is found by relaxing the definition of MNPG to be Maximally-expanded NPG. Based on this, we give the MNPG discovery algorithm in Algorithm 1.

```
   Input  : $\mathcal{P}, \epsilon$
   Output: $\mathcal{G}$
1  flag ← false;
2  while $|\mathcal{P}| > 0$ do
3      if $flag =$ false then
4          find nearest photo pair in $\mathcal{P}$;
5          if their GVD $> \epsilon$ then
6              return $\mathcal{G}$
7          create a new NPG $g$ with the pair;
8          remove the pair from $\mathcal{P}$;
9          flag ← true;
10     while $flag =$ true do
11         find $p_i \in$ remaining candidates, so that for
           $j \neq i$, $d_i^{max}$, the maximal distance between
           group members to the centroid of $g \bigcup \{p_i\}$ is
           smaller than any other $d_j^{max}$ of $g \bigcup \{p_j\}$ ;
12         if $d_i^{max} < \epsilon$ then
13             add $p_i$ to $g$;
14             remove $p_i$ from $\mathcal{P}$;
15         else
16             add $g$ to $\mathcal{G}$;
17             flag ← false;
18             break;
19 return $\mathcal{G}$;
```

Algorithm 1: Basic Algorithm for Discovering $\mathcal{G}$

The algorithm takes the local photo set and the GVD threshold as inputs, and it returns all the MNPGs it has found. Its main body is a loop (Line 2). In this loop, the algorithm performs two tasks, i.e., creating new NPGs (Line 3-9) and expanding existing NPGs (Line 10-18). In the first task, it needs to start a new NPG, so it attempts to find the initial photo pair for the new NPG by discovering the nearest pair of photos in the remaining local photo set (Line 4). If the GVD between the nearest pair is already larger than ϵ, the accumulated NPG collection $\mathcal{G}$ is returned and the algorithm is terminated (Lines 5-6); otherwise the pair is removed from the local photo set and the newly created NPG is forwarded to the following NPG expansion task by setting the flag as true (Lines 7-9). In the expansion task (Lines 10-18), the algorithm tries to evaluate all the remaining photos in $\mathcal{P}$, by finding the next group member p_i for the current NPG which results in the smallest radius of the updated group (Line 11). If the radius, i.e., d_i^{max} is less than ϵ, p_i is a qualified group member. p_i is added into the current NPG and removed from $\mathcal{P}$ (Lines 12-14); otherwise, the current NPG is a Maximally-expanded NPG and added into the MNPG set (Lines 15-18). The algorithm iterates until all the MNPGs are found. Note that the algorithm does not generate groups with a single photo. The remaining photos that do not belong to any group are considered as outliers.

It is been widely accepted that the *Antimonotonicity* property greatly affects the effectiveness of all frequent pattern mining techniques [22], [12]. Antimonotonicity can be understood as if a pattern P satisfies a certain constraint C, then any sub-pattern $P' \subset P$ also satisfies C. However, the constraint defined in Definition 5 is not antimonotone. The reason is that the centroid of any NPG varies when updates occur to the group members. It is possible that the original group is an NPG, but one of its subgroup is not an NPG.

A relaxed definition of the *Antimonotonicity* is the *loose Antimonotonicity*: if a pattern P satisfies a certain constraint C, then there exists at least one sub-pattern $P' \subset P$ that $|P'| = |P| - 1$ also satisfies C. Consequently we have the following lemma:

Lemma 1: The MNPGs discovered by Algorithm 1 are loose antimonotone.

Proof: The identification of MNPGs mainly follows the ϵ constraint which applies when creating a new NPG and adding new photos to an existing NPG. When creating a new NPG, the ϵ constraint (Line 5 in Algorithm 1) guarantees that the newly created group is an NPG. And then the expansion with new photos must follow the constraint as well (Line 12 in Algorithm 1). Now for any MNPG g_k, there at least exists one g'_k that is generated by removing the most recently added photo from g_k, has the size of $|g_k| - 1$ and satisfies the ϵ constraint. Hence g_k is loose antimonotone.

The *Loose Antimonotonicity* property of Algorithm 1 presents a solid theoretically support to the quality of the MNPGs identified [12], [22]. After this step, a seed photo for each MNPG can be easily found based on Definition 8. The next task is to discover top-*k* most novel photos from all the seed photos, whose novelty scores can be computed based on Definition 9. The definition of *novelty* reasonably reflects the photo's significance in terms of the cohesion of its group, the size of its group, the geographic distance to the query location, and the dissimilarity regarding other seed photos. If there do not exist k MNPGs, we can expand the set with other novel photos which do not belong to any MNPG.

Algorithm 1 describes the details of the main task in the diverse location visualization application. However, the major issue of this algorithm is that its high complexity makes it hardly scalable. Line 4 and Line 11 consist of expensive distance computations. In the worst case, the time complexity of the algorithm is $O(|\mathcal{P}|^3)$. Thus in the following sections, we will focus on how to improve the efficiency of Algorithm 1 to enable quick identification of MNPGs.

V. Hybrid Indexing

In this section, we introduce a new index structure called Hybrid Index to quickly discover MNPGs for a query location, powered by query optimization techniques to reduce the full distance computations. The time complexity of the refined algorithm for discovering MNPGs is also analyzed.

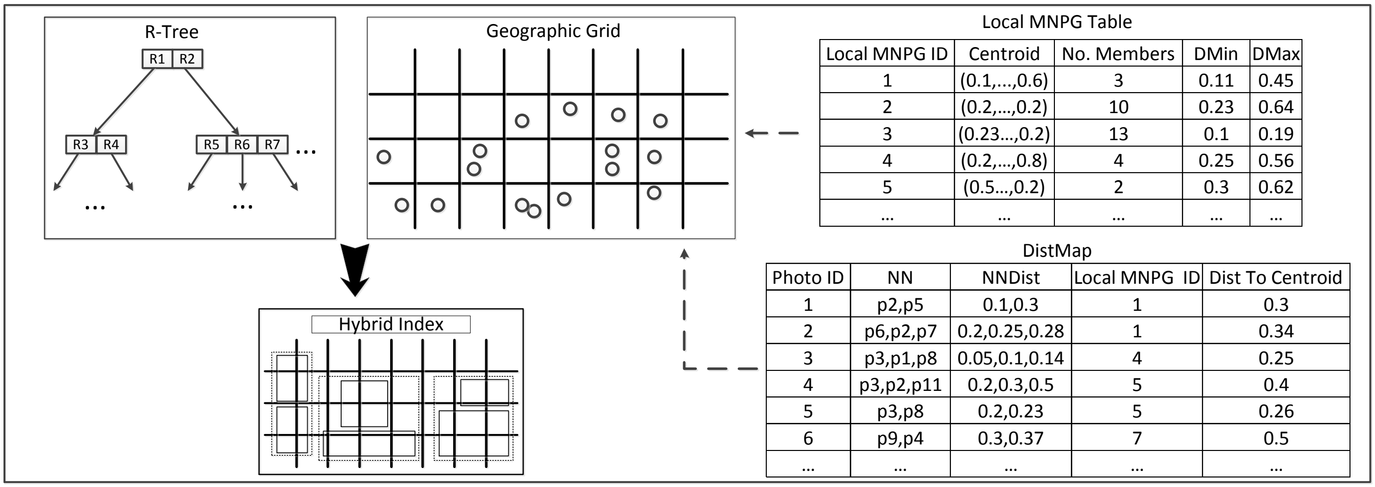
A. Index Structure

The basic idea of the Hybrid Index is to preprocess as much as information that is needed to identify the MNPGs for any arbitrary query. It is obvious that pre-computing the MNPGs for all the geo-tags or locations in the database is impractical, not only for the extremely long time needed but also for that the pre-computed MNPGs may not satisfy a specific geographic search range. So we devise a hybrid indexing structure that integrates the followings.

1) An R-Tree is used to index the geo-tags of photos so as to retrieve the local photo set efficiently.
2) A Geographic Grid is used to partition the geographic space into grids. For photos within each individual grid, their local MNGPs are pre-computed and indexed in a local MNPG table. One grid may contain multiple local MNPGs. The details of each local MNPG containing the centroid, ids of members, maximal and minimal distances from members to centroid are recorded in the local MNPG table. A distance map that stores auxiliary information including every photo's id, its near-duplicates (NN), its distances to near-duplicates, its local MNPG id and the distance to its local MNPG centroid, is also maintained. According to Definition 5, the distance threshold for being an NN is 2ϵ.

Figure 2 illustrates the index structure of the Hybrid Index. The information in the index structure is used to efficiently identify MNPGs for a query location with its geographic search range. To construct the index, standard R-tree operations can be applied to build an R-tree on photos' locations. As for the Geographic Grid, the whole geographic space is firstly divided into grids. For each grid which represents a locality, all the photos in the grid then applies Algorithm 1 to find all the local MNPGs in the grid which are kept in the local MNGP table. Meanwhile, information for each individual photo on its near-duplicates and local MNPG in the grid is also recorded in the distance map. Both the local MNGP table and distance map are large in size and stored externally.

With the Hybrid Index, the basic algorithm for discovering MNPGs, i.e., Algorithm 1, can be significantly improved. The key is to utilize the pre-computed information in the local MNPG table and distance map. Given a query location and its geographic search range, the local photo set can be efficiently



Local MNPG Table

Local MNPG ID	Centroid	No. Members	DMin	DMax
1	(0.1,...,0.6)	3	0.11	0.45
2	(0.2,...,0.2)	10	0.23	0.64
3	(0.23...,0.2)	13	0.1	0.19
4	(0.2,...,0.8)	4	0.25	0.56
5	(0.5...,0.2)	2	0.3	0.62
...	...	...	...	...

DistMap

Photo ID	NN	NNDist	Local MNPG ID	Dist To Centroid
1	p2,p5	0.1,0.3	1	0.3
2	p6,p2,p7	0.2,0.25,0.28	1	0.34
3	p3,p1,p8	0.05,0.1,0.14	4	0.25
4	p3,p2,p11	0.2,0.3,0.5	5	0.4
5	p3,p8	0.2,0.23	5	0.26
6	p9,p4	0.3,0.37	7	0.5
...	...	...	...	...

Fig. 2: Hybrid Index

retrieved by searching the R-tree, and all the grids that intersect with the query search area can be quickly found by searching the Geographic Grid. The corresponding entries in the distance map and local MNPG table are then fetched into memory to facilitate the discovery of MNPGs. The reason we use R-tree on top of the Geographic Grid is that the local photo set returned from R-tree is a subset of the photo set contained in the intersecting grids. Therefore, the partial distance map fetched into the memory can be smaller. By arranging the data continuously on the disk, few random accesses occur.

The major challenge here is to reduce the time complexity of Algorithm 1. Both the local MNPG table and distance map are designed for this purpose. Especially, the local MNPG table provides an opportunity to prune data in a group basis. Given a query location and its geographic search range, either one or multiple grids are involved. Since the space is pre-split and the local MNPGs are pre-computed according to the grids, the pre-computed local MNPGs may not be the correct MNPGs for the query.

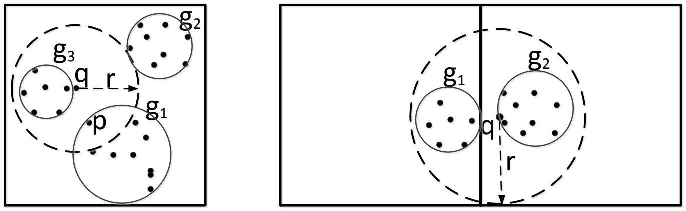


Fig. 3: Local Photo Set on Single and Multiple Grids

Figure 3 demonstrates two cases, where the rectangles are geographic grids, the points are the photos in the grids, the dashed circles indicate geographic search ranges for the queries and the solid circles indicate the pre-computed local MNPGs in the grids. In both cases, there exists inconsistency in the results for MNPG identification. For instance, in the left sub figure, part of g_1 is included in the local photo set of the query and may form a single MNPG for the query. In the right sub figure, two grids are involved for the query and both local MNPGs from two grids may form a single MNPG for the query since they are very close. Clearly, if we directly manipulate these identified local MNPGs to construct the MNPGs for any arbitrary query, excessive re-computations are inevitable. Fortunately the way the Hybrid Index stores data enables great potential for various query optimizations. In the next subsection, we focus on how to exploit the local MNPG table and distance map to dramatically reduce the complexity of discovering MNPGs.

B. Query Optimization

Here we fully utilize the local MNPG table and distance map to refine the basic algorithm for discovering MNPGs, by reducing expensive distance computations. We first introduce some lemmas and theorems to support our refinement.

Lemma 2: The GVD function is metric.
Proof is omitted due to its simplicity.

Lemma 3: Given an MNPG g_k and a photo $p \notin g_k$, the GVD between the new centroid c'_k of $g_k \bigcup \{p\}$ and the original centroid c_k of g_k satisfies $gvd(c_k, c'_k) = \frac{gvd(p,c_k)}{|g_k|+1}$.
Proof: Given $gvd(c_k, c'_k) = \lambda d_g(c_k, c'_k) + (1-\lambda) d_v(c_k, c'_k)$, let us see the visual distance component in $gvd(c_k, c'_k)$. We have $d_v(c_k, c'_k) = \sqrt{\sum_{i=1}^{D_v} (d^i)^2}$, where D_v is the dimensionality of the visual feature and d^i is the visual distance on the i^{th} dimension, i.e. $d^i = c_k^i - c_k^{i\,\prime}$. Obviously, the values on the i^{th} dimension of c_k and c'_k are computed with $c_k^i = \frac{\sum_{p_{kj} \in g_k} p_{kj}^i}{|g_k|}$ and $c_k^{i\,\prime} = \frac{\sum_{p_{kj} \in g_k \bigcup \{p\}} p_{kj}^i}{|g_k \bigcup \{p\}|}$. So by denoting $n = |g_k|$ and $t^i = \sum_{p_{kj} \in g_k} p_{kj}^i$, we get:

$$
\begin{aligned}
d^i &= c_k^i - c_k^{i\,\prime} = \frac{t^i}{n} - \frac{t^i + p^i}{n+1} && (9)\\
&= \frac{nt^i + t^i - nt^i - np^i}{n(n+1)} = \frac{t^i - np^i}{n(n+1)} = \frac{c_k^i - p^i}{n+1} && (10)
\end{aligned}
$$

and

$$d_v(c_k, c'_k) = \sqrt{\sum_{i=1}^{D_v}(\frac{c_k^i - p^i}{n+1})^2} \quad (11)$$

$$= \frac{\sqrt{\sum_{i=1}^{D_v}(c_k^i - p^i)}}{n+1} = \frac{d_v(c_k, p)}{n+1} \quad (12)$$

Similarly, we get $d_g(c_k, c'_k) = \frac{d_g(c_k, p)}{n+1}$ and hence

$$gvd(c_k, c'_k) = \frac{gvd(p, c_k)}{|g_k| + 1} \quad (13)$$

Lemma 3 is very important as it leads to the computation of the lower bounds for individual photo candidates.

In Algorithm 1, Line 4 and Line 11 are the major causes of high complexity. In Line 4, too many unnecessary pairwise distance computations are performed, and in Line 11 excessive numbers of candidates are assessed for distance computations. Our objective is to effectively remove those false photo candidates, thus to reduce distance computations as much as possible while keeping the correctness of the found MNPGs for the query.

1) Lower Bounds for Individual Photos: In NPG expansion of Algorithm 1, when an individual photo in a grid is considered, there exist several cases in which the photo can be quickly neglected without expensive computations. The first studies the distances between a photo to any photo in the current NPG.

Theorem 1: Given an NPG g_k, if photo p satisfies

$$\exists p_{ki} \in g_k \; so \; that \; gvd(p, p_{ki}) > 2\epsilon \quad (14)$$

then $p \notin g'_k \supseteq g_k$.
Proof: For any NPG, the maximally possible GVD between a pair of group members is 2ϵ. Given that the GVD is metric, if $p \in g_k$, the maximal distance between p and any other photo in g_k could be 2ϵ. This contradicts with the condition of $gvd(p, p_{ki}) > 2\epsilon$.

In the distance map, we need to keep the near-duplicate photos that have smaller than or equal to 2ϵ GVD values. When we expand an NPG, Theorem 1 guarantees that only the near-duplicates need to be considered.

Theorem 2: Given an NPG g_k, a photo $p \notin g_k$, denote c'_k as the centroid of the group $g_k \bigcup \{p\}$, the GVD between $p_{ki} \in g_k$ and the centroid c_k of g_k as d_1, and the GVD between p_{ki} and p is d_2, if

$$|d_1 - d_2| > \frac{(|g_k| + 1)\epsilon}{|g_k|} \quad (15)$$

then $p \notin g'_k \supseteq g_k$
Proof: Figure 4 demonstrates this theorem. We can easily prove that $gvd(p, c'_k) = \frac{|g_k| gvd(p, c_k)}{|g_k|+1}$. Now with triangular inequity, we have $gvd(p, c_k) \geq |d_1 - d_2|$, hence $gvd(p, c'_k) \geq \frac{|g_k||d_1 - d_2|}{|g_k|+1}$. When $\frac{|g_k||d_1 - d_2|}{|g_k|+1} > \epsilon$, $gvd(p, c'_k) > \epsilon$, the theorem is established.

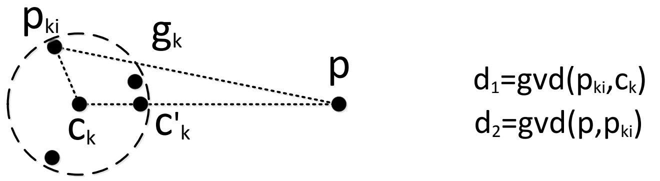


Fig. 4: Lower Bound for Individual Photos

Theorem 3: Given an NPG g_k and a photo p, denote the centroid of g_k as c_k, if p satisfies

$$gvd(p, c_k) > \frac{(|g_k| + 1)\epsilon}{|g_k|} \quad (16)$$

then $p \notin g'_k \supseteq g_k$.
Proof: According to Lemma 3, after the addition of p, the new centroid will move towards p for a distance of $\frac{gvd(p, c_k)}{|g_k|+1}$. So the GVD between p and the new centroid is $gvd(p, c'_k) = gvd(p, c_k) - \frac{gvd(p, c_k)}{|g_k|+1} = \frac{|g_k| gvd(p, c_k)}{|g_k|+1}$. Given Inequity 16, $gvd(p, c'_k) > \epsilon$, hence p can not be added to g_k.

Theorems 1, 2 and 3 provide effective multi-criteria pruning of the individual candidates and determine the way we build the distance map. They are seamlessly supported by the Hybrid Index. With Theorem 1, by joining the NN lists in the distance map, only a very small set of candidates remain. Then, in each iteration of NPG expansion, further reduction to the candidate set is performed with Theorem 2 using the NN distance (i.e., d_2) and distance to centroid (i.e., d_1) in the distance map. Theorem 3 requires one GVD computation to avoid iterative distance computations of false candidates in NPG expansion. As we can see, the distance map is fully utilized to compute the lower bounds for individual photos in the local photo set.

2) Lower Bounds for Groups: Another major advantage of the Hybrid Index is that it supports group-based pruning, given the local MNPG table.

Theorem 4: Given two NPGs g_{k1} and g_{k2}, denote the distance between the centroids as $d_1 = gvd(c_{k1}, c_{k2})$, the maximal distance from the members in g_{k2} to the centroid c_{k2} as d_2, $\forall p_{k2i} \in g_{k2}, p_{k2i} \notin g'_{k1} \supseteq g_{k1}$, if:

$$d_1 - d_2 > \frac{(|g_{k1}| + 1)\epsilon}{|g_{k1}|} \quad (17)$$

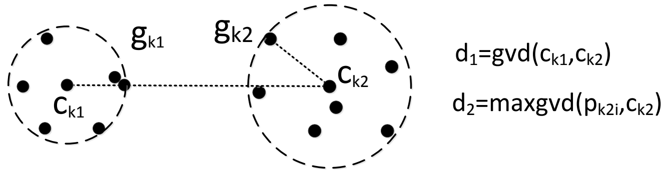


Fig. 5: Lower Bound for Groups

Proof: As Figure 5 shows, the GVD between any photo p_{k2i} and the centroid c_{k1} satisfies $gvd(p_{k2i}, c_{k1}) \geq d_1 - d_2$, when $d_1 - d_2 > \frac{(|g_{k1}|+1)\epsilon}{|g_{k1}|}$ and $gvd(p_{k2i}, c_{k1}) > \frac{(|g_{k1}|+1)\epsilon}{|g_{k1}|}$. According to Theorem 3, p_{k2i} can not be added to g_{k1}.

This theorem is used to help to exclude local MNPGs in the grids from the candidate set. Since the information of the local MNPGs, such as centroid and maximal distance, is stored in the local MNPG table, it is very efficient to prune the whole group of photo candidates from being compared. The only extra computation involved is to calculate the GVD between two centroids, according to Theorem 4.

```
   Input  : $\mathcal{P}, \epsilon$, distance map $DMAP$ and local
            MNPG table $MTAB$
   Output: $\mathcal{G}$
1  flag ← false;
2  while $|\mathcal{P}| > 0$ do
3      if $flag =$ false then
4          find nearest pair in $\mathcal{P}$ with $DMAP$;
5          if their GVD $> \epsilon$ then
6              return $\mathcal{G}$
7          create a new NPG $g$ with the pair;
8          remove the pair from $\mathcal{P}$;
9          flag ← true;
10     reduce candidate photos with Theorem 1;
11     while $flag =$ true do
12         reduce local MNPGs with Theorem 4;
13         reduce candidate photos with Theorem 2 & 3;
14         find $p_i \in$ remaining candidates, so that for
           $j \neq i$, $d_i^{max}$, the maximal distance between
           group members to the centroid in $g \bigcup \{p_i\}$ is
           smaller than any other $d_j^{max}$ of $g \bigcup \{p_j\}$ ;
15         if $d_i^{max} < \epsilon$ then
16             add $p_i$ to $g$;
17             remove $p_i$ from $\mathcal{P}$;
18         else
19             add $g$ to $\mathcal{G}$;
20             flag ← false;
21             break;
22 return $\mathcal{G}$;
```

Algorithm 2: Refined Algorithm 1 with Hybrid Index

3) Algorithm Refinement: The lower bounds we discovered provide us with rich options to improve the performance of MNPG identification with the Hybrid Index. The refined algorithm, outlined in Algorithm 2, employs all these lower bounds to improve the efficiency significantly. Note that only partial distance map and local MNPG table are fetched into memory based on the search results from R-tree and Geographic Grid. Compared with the basic Algorithm 1, there are several major differences. In Line 4 of Algorithm 2, the nearest pair in the local photo set can be quickly found by referring to the distance map. In Line 10 of Algorithm 2, the candidate photo set is filtered with Theorem 1 before the algorithm enters the expansion loop. For each iteration of the expansion loop, the number of local MNPGs is firstly reduced with Theorem 4 (Line 12), followed by pruning the remaining candidate set with Theorem 2 & 3 (Line 13). After the above filtering steps are performed, most candidates are removed from being further considered. Since distance information for near-duplicate photos is maintained in the distance map, the number of distance computation can be minimized. Actually, only the distances to centroids need to be computed. This is valid with the assumption that the memory is large enough to hold partial and small distance map and local MNPG table. Next we estimate the time complexity for Algorithm 2.

4) Complexity Analysis: For analysis purpose, we assume that all the photos are uniformly distributed in both the geographic space and the visual space. We note that there needs no distance computation in Line 4 of Algorithm 2 at all. So we only need to evaluate the upper bound size of the candidate photos that are processed in Line 14. In Line 10, it is guaranteed that for each MNPG, only the photos that have smaller GVD values than 2ϵ to all the current group members are considered. So it is a safe upper bound for the number of candidate photos to be assessed in Line 14.

We employ the concept of correlation fractal dimension of the point set [23], [2] for analysis. For a general search algorithm, the following equation can be used to estimate the number of neighbors given a search range ρ:

$$nb(\rho,' shape') = \left(\frac{v(\rho,' shape')}{v(\rho,' rectangle')} \right)^{\frac{D_2}{D}} (N-1)(2\rho)^{D_2} \quad (18)$$

Here $v(\rho,' shape')$ is the volume of a high-dimensional object with the radius ρ, N is the number of points, D is the dimensionality of the data space, and D_2 is the correlation fractal dimension of the dataset. $nb(\rho,' shape')$ estimates the number of neighbors within the query volume $v(\rho,' shape')$.

Obviously in our case, the search range defined by 2ϵ is a hypersphere. Since the computation of a hypersphere depends on its space dimensionality, here we assume the dimensionality is even and $D = 2k$ (same method can be applied for an odd D). Then we get the volumes for the query and the whole data space respectively as:

$$v(2\epsilon,' hypersphere') = \frac{\pi^k}{k!}(2\epsilon)^{2k} \quad (19)$$

$$v(2\epsilon,' hypercube') = (4\epsilon)^{2k}. \quad (20)$$

Subsequently Equation 18 for our case can be rewritten as:

$$nb(2\epsilon,' hypersphere') = \left(\frac{\pi^k}{2k!} \right)^{\frac{D_2}{2k}} (N-1)(4\epsilon)^{D_2} \quad (21)$$

This is the upper bound of the number of photos to be assessed in the expansion loop. In Line 12 and 13, for each loop the candidate set is further reduced to be much smaller than the upper bound. However, it is difficult to theoretically estimate their reduction degrees. Their effect will be fully tested in the experiments. Nonetheless, denote the candidate set after Line

10 as $\mathcal{C}$ and the further reduced candidate set after Line 13 in each loop as $\mathcal{C}_{\nabla}$, the time complexity of the basic Algorithm 1 is reduced from $O(|\mathcal{P}|^3)$ to be $O(|\mathcal{P}| \cdot |\mathcal{C}| \cdot |\mathcal{C}_{\nabla}|)$. It is expected that $|\mathcal{C}_{\nabla}| \ll |\mathcal{C}| \ll |\mathcal{P}|$.

VI. Experiments

In this section, we investigate the performance of our framework with extensive experiments on a synthetic dataset and a real geo-photo dataset. In the experiments, we make ten random queries and evaluate the average performance. After the framework has identified all the MNPGs within the search proximity, top-k most novel seed photos are returned. All the experiments are conducted on a desktop computer with Intel Core i7 2.93GHz CPU and 8GB memory.

A. Datasets

Two datasets are prepared to test the quality and efficiency of the proposed framework for diverse location visualization:

1) Point1M: This is a synthetic dataset containing 1,000,000 points. It is generated in such a way that all the points are well distributed in the geographic space, while each point has a random number of near-duplicates. Such properties provide good reflection of the performance on a general dataset in our framework. Specifically, we start with randomly creating a point with a randomly generated geo-tag, then we generate a (ranged) random number of near-duplicates for it. Based on Definition 5, the generated near-duplicates have the distances smaller than the GVD threshold ϵ. The procedure continues until we have 1,000,000 points. The average number of near-duplicates for a point is about 30.
2) Geo600K: This dataset consists of 600,000 real geo-tagged photos collected from Panoramio and Flickr, which is a superset of Paris500K [26]. Paris500K is a well organized dataset of geo-tagged photos crawled from Panoramio, and all the photos in this dataset are taken at Paris. To further increase the data size, we complement Paris500K with 100,000 more geo-tagged photos of Paris from Panoramio and Flickr.

The pre-processing of data involves feature extraction (for Geo600K) and index construction (for both databsets). The index construction requires some parameters specified, so unless specified, we used the default settings as in Table II:

Parameter	Default Value(s)
GVD threshold ϵ	0.15
Number of returned novel photos k	20
Fusion weight of d_g in GVD λ	0.5
Geographic Grid size scaler μ	1.0
Geographic search range r	0.003^o

TABLE II: Default Parameter Settings

We have carefully generated the data values in Point1M to make it align with the scales in Geo600K, so that we can use the same parameter setting for both datasets throughout the experiments. 500-dimensional BoW feature generated from SIFT local keypoints is used to represent each photo in Geo600K. Accordingly, the dimensionality for Point1M is also 500 with the same value range. For the Geographic Grid size, we first compute the ranges of latitude and longitude from the geo-tags, and then we split the regions they cover so that each grid contains around 1000 photos, where the grid size is 0.0033^o latitude and 0.0047° longitude. To assess the effect of grid size, we use a scaler μ on it, so that in our index the actual grid size is $0.0033\mu^o$ latitude and $0.0047\mu^o$ longitude.

B. Reference Methods

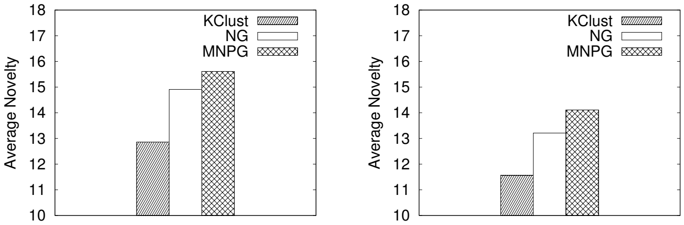
Two reference methods are implemented to compare the performance with our framework. The first one is KClustering-based novel photo identification (KClust) [16], and the second is Near-duplicate Graph (NG) [12]. After retrieving local photo set, we perform novel photo identification with both reference methods, and the outcomes are measured and compared. In KClust, we use k to specify the number of clusters. And then we find the nearest point to the center of each cluster. For NG, we use a carefully tuned value for its parameter γ, i.e., $\gamma = 0.85$, and it defines a similar criteria to identify the representative photos.

C. Quality of Novel Contents

The effectiveness of our framework is evaluated by the *average novelty* of returned seed photos. According to Definition 9, given a set of the k returned photos $\mathcal{S}^k$, the average novelty is computed by

$$\frac{\sum_{s_i \in \mathcal{S}^k} \eta(s_i)}{|\mathcal{S}^k|}. \qquad (22)$$

All the returned photos by each method are evaluated with the average novelty, which are are displayed and compared.



(a) Novelty of Point1M

(b) Novelty of Geo600K

Fig. 6: Comparison on Novelty

1) Comparison: Figure 6 illustrate the difference in the performance of three methods w.r.t the average novelty values. In Figure 6a, clearly our MNPG discovering algorithm has the best quality by giving an average novelty of 15.6, and NG follows MNPG as the second by an average novelty of 14.9. KCluster yields the worst novelty, i.e., 12.8, which can be expected since KCluster does not have any constraints and thus leads to the probability of creating huge clusters. NG uses a similar strategy with cohesion and connectivity constraints to find the near-duplicate graphs. Its performance is hence similar to MNPG. Figure 6b shows the same order of the three methods by their average novelty values. We note that the novelty values for Geo600K are smaller than Point1M. That is mainly because in Point1M the near-duplicate groups have slightly greater sizes and the distances between group members are smaller.

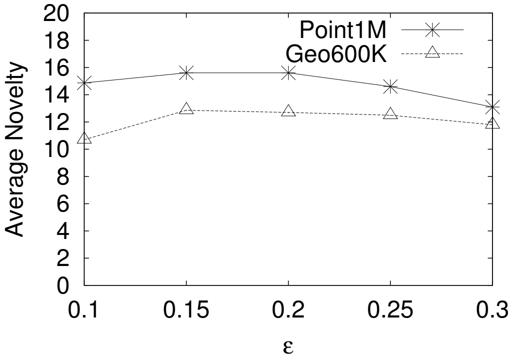


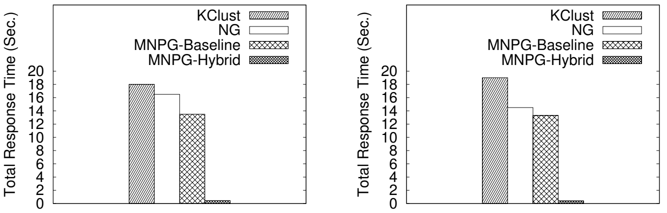
Fig. 7: Effect of ϵ on Novelty

2) Effect of ϵ: The effect of ϵ on the average novelty values is reflected in Figure 7. The ϵ is the only constraint that determines the quality of the MNPGs. With a greater ϵ, the average size of MNPGs is greater. However non-near-duplicates may be added into the MNPG and thus the average distance between other group members to the group centroid also becomes greater. On the other hand, given a small ϵ, the MNPGs tend to be tighter, which means the average distance between other group members to the group centroid is smaller and the group size is also smaller. However there exists possibility that true near-duplicates may not be added to the correct MNPG. Instead it may start another MNPG, which reduces the *uniqueness* component in the novelty measure. Figure 7 supports the observations above. Both figures prove that only within a proper range the average novelty would peak. Very small or large ϵ values degrade the average novelty values greatly. ϵ also affects several other aspects of the framework including total response time, number of expensive computations, index size and memory usage, etc. Its effect on those aspects will be further investigated in the rest of the experiments.

D. Efficiency of Search

The total response time is mainly used to examine the effect of parameters on the efficiency aspect of our framework.

1) Comparison: The average total response time values are reported for the two reference methods, our basic algorithm, and the refined algorithm with the Hybird Index in Figure 8. KCluster, NG and MNPG-Baseline have similar speed (18, 15 and 14 seconds), while the refined MNPG-hybrid outstands with 0.4 second only. The high complexity of KCluster, NG and MNPG-Baseline leads to their unsatisfactory speed. MNPG-Hybrid is greatly accelerated by the Hybrid Index and the attached optimizations, where only a small number full distance computations occur.



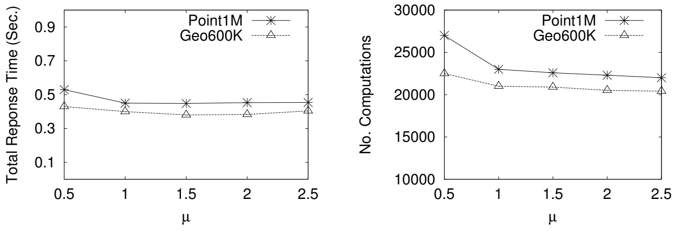
(a) Point1M

(b) Geo600K

Fig. 8: Comparison on Search Efficiency

2) Effect of ϵ: Figure 9 illustrates the effect of ϵ on the search efficiency, with both total response time and number of full distance computations reported. The trend of changes in Figure 9 reflects that in MNPG-Baseline, when ϵ is increased, the photos in the local photo set converge into MNPGs more quickly, making the candidate set shrink faster as well. Hence the total response time and the number of computations drop with a larger ϵ value. MNPG-Hybrid shows a similar pattern of changes in its total response time but it has a slightly different change in the number of distance computations. This is because with the increase of ϵ, less points can be filtered by the query optimizations. But the faster construction of MNPGs compensates this loss and thus the total response time remains the down trend.

3) Effect of μ: The Geographic Grid size, represented by μ, determines the average number of photos indexed in a grid, and consequently affects the quality of local MNPGs in the grids. A greater μ will make the local MNPG more accurate, i.e., local MNPGs are more complete with less truncations by the partition of grids. Its effect on the search efficiency is demonstrated in Figure 10. As MNPG-baseline is not affected by this factor, only the performance of MNPG-Hybrid is shown. Clearly, with local MNPGs being more complete, more groups are directly removed from the candidate set, thus both the total response time and the number of computations declines as μ increases. However it is infeasible to set a very large μ, as a greater μ will also increase the index size, and will greatly slow down the index construction. μ's effects on index construction and memory usage are reported shortly.

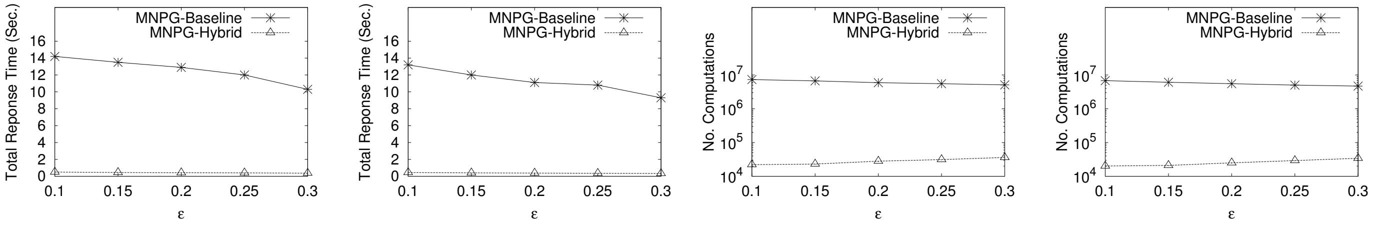


(a) Total Response Time

(b) No. Computations

Fig. 10: Effect of μ on Search Efficiency

4) Effect of r: The geographic range of a visualization query determines the number photos in the local photo set.



(a) Point1M (b) Geo600K (c) Point1M (d) Geo600K

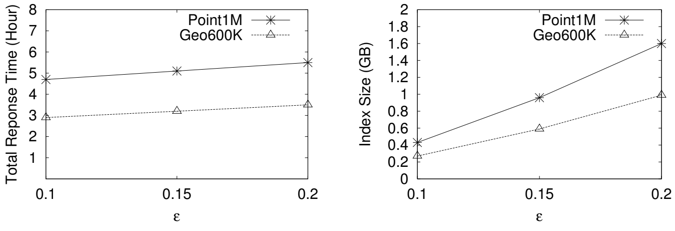
Fig. 9: Effect of ϵ on Search Efficiency

Naturally it has great impact on the search efficiency. In Figure 11 we test how the total response time and the number of distance computations change with r. Both the results for MNPG-Baseline and MNPG-Hybrid are included. r is set to change through $1e^{-3}$ to $5e^{-3}$ degrees. Recall that the default geographic grid size is 0.0033^o latitude and 0.0047^o longitude, the range of r covers the possibility of using single grid to multiple grids in the query processing. Generally, a greater r will lead to a greater candidate set in the local photo set, which slows the query processing down. This is particularly evident from the total response time and number of distance computations of MNPG-Baseline. However, for MNPG-Hybrid, as irrelevant photos can be quickly filtered, its efficiency is less affected.

E. Index Construction

Next let us see how parameters ϵ and μ affect the index construction in terms of the time cost and the index size.

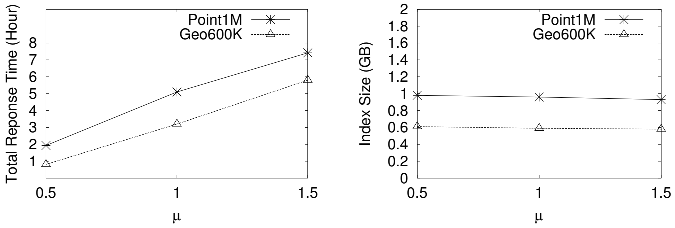
a) Effect of ϵ: ϵ has impacts on both the index construction time and the index size. In Figure 12, the total time cost does not vary too much with ϵ, as we previously explained, a greater ϵ leads to faster local MNPG identification with the MNPG-Baseline algorithm. However, the index size grows rapidly in Figure 12, indicating the number of near-duplicates a point or photo has increases rapidly when ϵ gets larger.



(a) Time Cost (b) Index Size

Fig. 12: Effect of ϵ on Index Construction

b) Effect of μ: When the Geographic Grid size is enlarged, more photos are involved in the local MNPG identification. As local MNPG identification is computationally expensive, the time cost of index construction is greatly increased. As for index size, an increased μ reduces the total number of local MNPGs and hence shrinks the local MNPG table for each Geographic Grid. As the number of total photos does not change, this slightly reduces the index size. Figure 13 clearly supports the above observations.



(a) Time Cost (b) Index Size

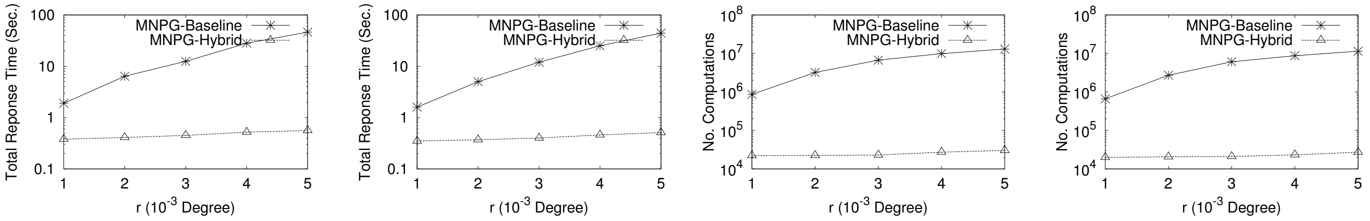
Fig. 13: Effect of μ on Index Construction

F. Memory Usage

As memory usage is an important aspect of efficiency evaluation, we also test it in our experiments. The results are shown in Figure 14. The memory usage for a query with default settings usually takes several MBs only. Two factors are considered when evaluating the memory usage, i.e., the query range r and the GVD threshold ϵ. r affects the memory usage as the number of grids it fetches varies with r. With a very small r, at least one grid needs to be fetched, so when $r = 1e^{-3}$ and $r = 2e^{-3}$ the memory usage is almost the same. As r continues to grow, more frequently the query proximity covers several grids and subsequently more grids need to be fetched into memory. For ϵ, it affects the memory usage in the sense that the group size is determined by it. It is a direct result of many larger groups which can be observed in Figure 14.

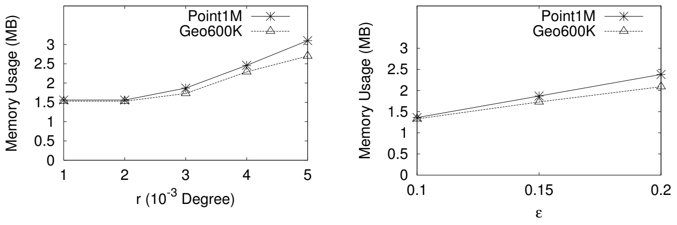
VII. Conclusion

Visualizing a location with geo-tagged photos is a novel application that requires urgent research attention. However, fueled by the popularity of GPD-enabled mobile devices, near-duplicates in the geo-tagged photos are manifesting themselves ceaselessly. The quality of location visualization is greatly harmed by highly redundant contents it returns to the user. To tackle this problem, in this work, a diverse location visualization framework is proposed to provide the user with diversified scenes around a location. We define a *novelty*



(a) Point1M (b) Geo600K (c) Point1M (d) Geo600K

Fig. 11: Effect of r on Search Efficiency



(a) Effect of r (b) Effect of ϵ

Fig. 14: Memory Usage

model and a Maximal Near-duplicate Photo Group (MNPG) identification algorithm to identify the near-duplicate groups. Then, to improve the efficiency of the framework, we design a Hybrid Index that simultaneously indexes geographic information, near-duplicate information and local MNPG information. The Hybrid Index creates unique opportunity to refine the original MNPG identification by lower bound pruning techniques. Finally, we conduct comprehensive experiments on two datasets, and verify the advantages of our framework on both quality and efficiency. In future, we plan to investigate the hierarchical version of the Geographic Grid to visualize locations in a multi-resolution fasion.

References

[1] S. A. Ay, R. Zimmermann, and S. H. Kim. Relevance ranking in georeferenced video search. *Multimedia Syst.*, 16(2):105–125, 2010.

[2] A. Belussi and C. Faloutsos. Estimating the selectivity of spatial queries using the ‘correlation’ fractal dimension. In *VLDB*, pages 299–310, 1995.

[3] L. Cao, J. Luo, A. C. Gallagher, X. Jin, J. Han, and T. S. Huang. A worldwide tourism recommendation system based on geotaggedweb photos. In *ICASSP*, pages 2274–2277, 2010.

[4] L. Cao, J. Yu, J. Luo, and T. S. Huang. Enhancing semantic and geographic annotation of web images via logistic canonical correlation regression. In *ACM Multimedia*, pages 125–134, 2009.

[5] W.-C. Chen, A. Battestini, N. Gelfand, and V. Setlur. Visual summaries of popular landmarks from community photo collections. In *ACM Multimedia*, pages 789–792, 2009.

[6] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *VLDB*, pages 426–435. Morgan Kaufmann, 1997.

[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] G. C. de Silva and K. Aizawa. Retrieving multimedia travel stories using location data and spatial queries. In *ACM Multimedia*, pages 785–788, 2009.

[9] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *ACM SIGMOD*, pages 47–57, 1984.

[10] Q. Hao, R. Cai, J.-M. Yang, R. Xiao, L. Liu, S. Wang, and L. Zhang. Travelscope: standing on the shoulders of dedicated travelers. In *ACM Multimedia*, pages 1021–1022, 2009.

[11] C.-C. Hsieh, W.-H. Cheng, C.-H. Chang, Y.-Y. Chuang, and J.-L. Wu. Photo navigator. In *ACM Multimedia*, pages 419–428, 2008.

[12] Z. Huang, B. Hu, H. Cheng, H. T. Shen, H. Liu, and X. Zhou. Mining near-duplicate graph for cluster-based reranking of web video search results. *ACM Trans. Inf. Syst.*, 28(4):22, 2010.

[13] Z. Huang, H. T. Shen, J. Liu, and X. Zhou. Effective data co-reduction for multimedia similarity search. In *ACM SIGMOD*, pages 1021–1032, 2011.

[14] Y.-G. Jiang and C.-W. Ngo. Bag-of-visual-words expansion using visual relatedness for video indexing. In *SIGIR*, pages 769–770, 2008.

[15] D. Joshi, A. C. Gallagher, J. Yu, and J. Luo. Exploring user image tags for geo-location inference. In *ICASSP*, pages 5598–5601, 2010.

[16] L. S. Kennedy and M. Naaman. Generating diverse and representative image search results for landmarks. In *WWW*, pages 297–306, 2008.

[17] L. S. Kennedy, M. Naaman, S. Ahern, R. Nair, and T. Rattenbury. How flickr helps us make sense of the world: context and content in community-contributed media collections. In *ACM Multimedia*, pages 631–640, 2007.

[18] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV*, pages 427–440, 2008.

[19] Y. Li, D. J. Crandall, and D. P. Huttenlocher. Landmark classification in large-scale image collections. In *ICCV*, pages 1957–1964, 2009.

[20] J. Liu, Z. Huang, H. Cai, H. T. Shen, C.-W. Ngo, and W. Wang. Near-duplicate video retrieval: Current research and future trends. *ACM Computing Surveys*, 2012.

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

[22] F. Moser, R. Colak, A. Rafiey, and M. Ester. Mining cohesive patterns from graphs with feature vectors. In *SDM*, pages 593–604, 2009.

[23] A. Papadopoulos and Y. Manolopoulos. Performance of nearest neighbor queries in r-trees. In *ICDT*, pages 394–408, 1997.

[24] H. T. Shen, B. C. Ooi, and X. Zhou. Towards effective indexing for very large video sequence database. In *SIGMOD*, pages 730–741, 2005.

[25] Y. Tao, K. Yi, C. Sheng, and P. Kalnis. Efficient and accurate nearest neighbor and closest pair search in high-dimensional space. *ACM Trans. Database Syst.*, 35(3), 2010.

[26] T. Weyand, J. Hosang, and B. Leibe. An evaluation of two automatic landmark building discovery algorithms for city reconstruction. In *RMLE*, 2010.

[27] Y. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven. Tour the world: Building a web-scale landmark recognition engine. In *CVPR*, pages 1085–1092, 2009.

[28] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma. Hybrid index structures for location-based web search. In *CIKM*, pages 155–162, 2005.