Summary

**

1. VISION

Smartphones have the ability to capture high-resolution images and video. Given the penetration of smartphones, it is now feasible to consider crowd-sourcing searches for visual evidence in forensic applications. For example, a police investigative unit might, after a crime, request bystanders for crime scene images, possibly in exchange for some financial incentives. [Add other examples?]

We envision a system in which a *requester* poses a *query* for images or video. In our system, a cloud service handles the query and extracts relevant images or video from smart phones of participating users.

Queries may be of different types (aggregate, enumeration), and may have timeliness requirements. More important, two aspects of this problem space make the problem challenging:

- First, images from different users may be related with each other (e.g., may have been taken at the same time and place), and related images may convey less information than disparate images.
- Second, each smartphone may have a limited "pipe" for sending images, given data plan costs as well as the fact that cellular network bandwidth has not kept pace with improvements in camera technology and smartphone storage.

To address these challenges, we consider a general framework for crowd-sourced queries for media. In this framework, relationships between images or videos, as determined by their metadata, is encapsulated in an *information network*. Images satisfying a query can be determined by traversing this information network. However, given wireless bandwidth constraints, not all of these images can be returned as responses for the crowd search, so an *objective function* is used to determine the most relevant queries. this objective function is constrained by a *network model*, which codifies network bandwidth constraints.

In the next section, we give several examples of this framework.

2. EXAMPLE OF DIFFERENT PROBLEMS

Above mentioned general framework for crowd-sourced queries can have different *information model*, *network model* and *objective function*. In this section, we provide several examples to these three components.

2.1 Network Model

Network model basically defines how media files are uploaded to the application server. Due to different scenarios or applications, different network models might be used. Here we briefly describe two possible models, naive model and peer-assistant model.

2.1.1 Naive Model

In naive model, each mobile phone only make one connection, to the application server, and will only upload its own media files. In this model, we assume there is no interaction or interference among phones. For simplicity, we assume each phone will upload multiple media files sequentially.

2.1.2 Peer-Assistant Model

For some queries with short deadline, by solely uploading its own files for each phone, the response result might not be satisfactory. If neighboring phones can help each other for uploading, it is more flexible and we can expect better response result. That is the intuition of peer-assistant model. Concretely, besides the connection to the application server, each phone can also connect to its neighboring phones through ad-hoc like links. In this case, each phone can upload its own media file as well as its neighbor's files. For example, consider an extreme case, where all the related media files are located in one phone, with strict deadline maybe only 1 photo can be uploaded; however with peer-assistant model, this phone can ask its neighbors for simultaneous uploading which might upload more files.

2.2 Information Model

Network model determines which uploading schedules (selection of subset of media files to upload) are feasible, while information model determines which uploading schedule is the best one. There are multiple possible information models, here we briefly discuss two models, coverage model and similarity model.

2.2.1 Coverage Model

The information of a media file can be represented by covered range on the physical spatial-temporal space. In other words, each media file covers some physical space. Then given a set of media files, the information such set provides is the intersection of the spaces covered by all media files.

2.2.2 Similarity Model

Similarity model would not answer the provided information question directly; instead, similarity model describes the similarity between a pair of media files. In this model, given a set of media files, the information such set provides is represented by how similar each pair of media files of the subset is. Intuitively, a high averaged similarity subset of media files indicates low information, and vice versa.

2.3 Objective Function

2.3.1

3. OUR PROBLEM

3.1 Preliminaries

We focus on one specific problem of the framework. In our problem we consider a cloud service, the central server and multiple participated smartphones. The query to our problem contains a feature vector (metadata) including *location*, *time*, *number of buildings*, *number of peoples*, *number of cars* and *photo histogram*. In addition, there is a *deadline* associated with each query.

To respond each query there are generally 4 steps:

- Given the query contents, server filters out irrelevant media files and keeps the qualified ones;
- Based on the information model and the network model, server determines uploading schedules for each participated smartphone;
- Smartphones receives its schedule from the server, upload the media files correspondingly;
- Server gets all the scheduled media files, responds the query.

For the first step, filtering step, we can build index based on the metadata. The index uses partial metadata as keys and sorts the all the metadatas such that for each query, qualified files can be located efficiently. Given a set of qualified files, different *network model*, *information model* and *objective function* might determine different schedules and thus different query response. Following we define such components for our problem concretely.

3.2 Network Model

We consider a fairly simple network model in our problem, namely, *naive model*. Assume there are M participated smartphones, each of whose uploading speed is b_i. Each smartphone will upload files sequentially if there are multiple files to upload. Each smartphone can upload files without any interference to others.

3.3 Information Model

We define $similarity$ to describe the relationship between any pair of files. To make problem clear, we take two files: file i and file j as an example here, the similarity vector $\overrightarrow{v}_{i,j}$ is composed by the following elements: time difference $t_{i,j}$, location difference $d_{i,j}$, object difference $o_{i,j}$, color distribution difference $c_{i,j}$, trace direction $tr_{i,j}$, etc. Since files are owned by different users, then file owner is also an important part for similarity, and we add this information as a binary value in similarity. So now we normalize these elements and define similarity as the weighted sum of these elements. With normalization, the similarity vector can be represented as follows:

$$\overrightarrow{v}_{ij} = \left(\frac{t_{i,j}}{\max\limits_{i,j}(t_{i,j})}, \frac{d_{i,j}}{\max\limits_{i,j}(d_{i,j})}, \frac{o_{i,j}}{\max\limits_{i,j}(o_{i,j})}, \frac{c_{i,j}}{\max\limits_{i,j}(c_{i,j})}, \frac{tr_{i,j}}{\max\limits_{i,j}(tr_{i,j})} \right) \tag{1}$$

To quantify the similarity value, we define a weighted sum function for similarity vector, $f(\overrightarrow{v}_{i,j})$, which is a numerical value.

$$f\left(\overrightarrow{v}_{ij}\right) = \alpha_1 \frac{t_{i,j}}{\max\limits_{i,j}(t_{i,j})} + \alpha_2 \frac{d_{i,j}}{\max\limits_{i,j}(d_{i,j})} + \alpha_3 \frac{o_{i,j}}{\max\limits_{i,j}(o_{i,j})} + \alpha_4 \frac{c_{i,j}}{\max\limits_{i,j}(c_{i,j})} + \alpha_5 \frac{tr_{i,j}}{\max\limits_{i,j}(tr_{i,j})} \tag{2}$$

in which, $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 = 1$, and each α's value is assigned by the query's preference, e.g. if commander prefers location difference, then α_1 gets more percentage.

3.4 Objective Function and Problem Formulation

Consider the simplest case, each file is of the same size and one query scenario. After filtering, we get N photos from M users, each user i owns a_{i-1} photos, that is, $a_0 + \ldots + a_{M-1} = N$. With the bandwidth and deadline constraint, users can only upload limited qualified files, more precisely, for user i, he can upload s_i files under the constraint, in which, $s_i \leq a_i$ and $s_0 + \ldots + s_{M-1} = K$. The problem now is to select a subset K photos from these N photos, such that:

$$\sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \sum_{k_i=0}^{s_i-1} \sum_{k_j=0}^{s_j-1} f\left(\vec{v}_{k_i k_j}\right) \tag{3}$$

get maximized, in which, $f(\overrightarrow{v}_{k_i k_j}) = 0$ if $k_i = k_j$ and $i = j$.

4. RELATED WORKS

4.1 Sensor Selection Problem

Sensor selection problem focus on how to select a subset of sensors for best aggregated sensing information.

4.1.1 Query

The query is to get an aggregated and overall sensing result of the entire sensor network.

4.1.2 Objective

Select a subset of all the sensors to get the most representative sensing results.

4.1.3 Information Model

Can be any information model. Actually a sensor in the sensor selection problem is exactly a media file in our problem. So any possible information model we discussed before can be applied to sensor selection problem directly.

4.1.4 Network Model

There is no network model for sensor selection problem, which is the main difference compared to our problem. For sensor selection problem, each sensor is a media file in our problem, which means that there is only one phone. In addition, there is no deadline constraint. So the best subset of sensors solely depends on the information model.

4.2 PhotoNet

4.2.1 Query

No specific query in this paper.

4.2.2 Objective

Because storage space is limited, each phone should decide the optimized photo set to store locally; namely, the stored photo set should present lowest average pairwise unsimilarity. (Based on the same idea but another minor objective is, to exchange the photo with highest unsimilarity first while encountering.)

4.2.3 Information Model

Use distance concept between a pair of photos, distance defined by binary geo-distance or time and color histogram. Users exchange only limited photos because of the limited meeting time and bandwidth. Peer-connection is manually.

4.2.4 Network Model

A typical DTN model, encounter-transmission strategy.

4.3 PhotoNet with Outlier Elimination

4.3.1 Query

Same as PhotoNet.

4.3.2 Objective

Same as PhotoNet, each phone is trying to maintain the most representative photo set. Instead of maximizing the distance directly, outliers are considered as the least representative photos (and will be dropped at very beginning); in the case of without any outliers, the cluster with the most number of photos will be dropped photo first, since the cluster provide redundant information.

4.3.3 Information Model

The distance concept is the same as above. Based on the distance, photos are grouped into clusters. The information provided by a cluster is related to how many photos are within this cluster. After ranked the photo within one cluster, as the number of photos contained in a cluster increasing, the additional information provided by the last photo (marginal utility) decreases; namely, first photo provides $p(c)$, second provides $l \cdot p(c)$, third provides $l^2 \cdot p(c)$, where l is a parameter which is less than 1. A cluster with relatively small number of photos is defined as a outlier cluster.

4.3.4 Network Model

Same as PhotoNet.

4.4 Diversity Caching

4.4.1 Query

Same as PhotoNet.

4.4.2 Objective

Maximize the utility function based on the cache. The solution is simple, always drop the lowest rank item of the cluster with the maximum number of items.

4.4.3 Information Model

Same as "PhotoNet with Outlier Elimination", photos are grouped into clusters, the utility of the entire set is defined as the sum of the utility of all the clusters. Each cluster's utility is using the decreasing model, namely, $p(c)$, $l \cdot p(c)$, $l^2 \cdot p(c)$.

4.4.4 Network Model

Not any.

4.5 CrowdSearch

4.5.1 Query

Query is initiated at the mobile phone, which is different from our architecture. Query is a image (query image); the server will return the images with the same contents of the query image.

4.5.2 Objective

Two phases photo selection for good accuracy of image response (first step is the server automatically select photos based on tags; second step is human validation).

4.5.3 Information Model

On the server side, all the photos are tagged automatically (SIFT). Given query image, the server will provide candidate images; after that human validation will happen on the candidates based on AMT platform.

4.5.4 Network Model

Delay on network communication is not important, they mainly focus on the delay of human validation of candidate images.

4.6 TagSense

4.6.1 Query

Not any.

4.6.2 *Objective*

Tag people occurred in a photo captured by smart phone automatically.

4.6.3 *Information Model*

Not any.

4.6.4 *Network Model*

Use WiFi connection to estimate distance between phones and thus tag people occurred in the photo.