Active Query Sensing for Mobile Location Search

Felix X. Yu
Dept. of Electrical Engineering
Columbia University
New York, NY 10027
yuxinnan@ee.columbia.edu

Rongrong Ji
Dept. of Electrical Engineering
Columbia University
New York, NY 10027
rrji@ee.columbia.edu

Shih-Fu Chang
Dept. of Electrical Engineering
Columbia University
New York, NY 10027
sfchang@ee.columbia.edu

ABSTRACT

While much exciting progress is being made in mobile visual search, one important question has been left unexplored in all current systems. When the first query fails to find the right target (up to 50% likelihood), how should the user form his/her search strategy in the subsequent interaction? In this paper, we propose a novel *Active Query Sensing* system to suggest the best way for sensing the surrounding scenes while forming the second query for location search. We accomplish the goal by developing several unique components – an offline process for analyzing the saliency of the views associated with each geographical location based on score distribution modeling, predicting the visual search precision of individual views and locations, estimating the view of an unseen query, and suggesting the best subsequent view change. Using a scalable visual search system implemented over a NYC street view data set (0.3 million images), we show a performance gain as high as two folds, reducing the failure rate of mobile location search to only 12% after the second query. This work may open up an exciting new direction for developing interactive mobile media applications through innovative exploitation of active sensing and query formulation.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.4 [**Image Processing and Computer Vision**]: [Scene Analysis, Object recognition]

General Terms

Algorithms, System, Measurement

Keywords

Mobile Visual Search, Mobile Location Recognition, Active Query Sensing, Content-based Image Retrieval

MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA.

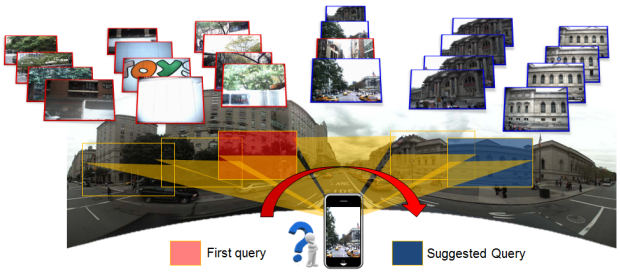

Figure 1: In mobile location search, the initial query often fails, as indicated by the incorrect matching results (in red). We propose Active Query Sensing to suggest the best view for subsequent queries (shown in blue) without leaving users to trials and errors.

1. INTRODUCTION

As mobile handheld devices become pervasive, new applications like mobile media search emerge. One promising example is searching information about products, locations, or landmarks, by taking a photograph of the target of interest using the mobile device. The captured image is used as a query, sent over the mobile network to the server in which reference images of the candidate objects or locations are matched in order to recognize the true target. Such mobile visual search functionalities have been shown recently in commercial systems, such as snaptell [1], Nokia point and find [2], kooaba [3], and a few research prototypes [4][5][6]. One of the most interesting topics with wide applications is the mobile location search [6][7]. Mobile location search offers a service complementary to GPS or network based localization, because the recognized location may be more precise and no satellite or cellular network infrastructures are needed.

Mobile location search systems mentioned above are built based on image matching, whose success depends on the separability of image content associated with different targets (inter-class distance), divergence of content among reference images of the same target (within-class variation), and distortion added to the query image during the mobile imaging process. Ideally, every view of a target location can be used as a query to successfully recognize the true target, correctly rejecting others.

However, not every view[1] of a location is distinctive enough to be used as a successful query, as illustrated in Figure 1.

[1]We use a generic term "view" to refer to the orientation, scale, and location for taking the image as the mobile visual query input.

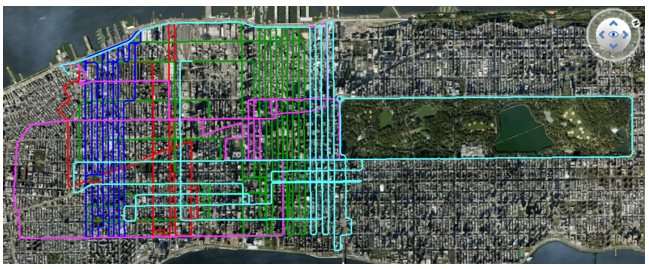

Figure 2: The geographical distribution of the five routes in the NAVTEQ New York City data set, which covers 0.3 million street view images taken from 50,000 locations.

Figure 3: (a) The visual coverage of six cameras in the NAVTEQ NYC data set. (b) Typical example of six photos in one location (the Metropolitan Museum of Art) of the NYC street view data set.

In an experiment to be described later in the paper, only 47% of locations are successfully recognized when the online mobile query is taken from an arbitrary view. When it fails, incorrect locations with visually similar appearances are returned as the top match. Such performance appears to be consistent with the modest accuracy (0.4-0.7 average precision) reported in the state-of-the-art systems [8][4][6][9] for mobile location search. Several exciting research directions are being pursued to advance the state of the art and reduce the recognition errors. Discussion of the related works will be presented in Section 5.

Different from prior works, in this paper we focus on a novel aspect of improving the mobile visual search experience. We hypothesize that there exist unique preferred views for successful recognition of each target location. For example, some views of a location consist of unique "signature" attributes that are distinctively different from others. Other views may contain common objects (trees, walls, *etc.*) that are much less distinctive. When searching for specific targets, queries using such unique preferred views will lead to much better recognition results. To this end, we propose an automated **A**ctive **Q**uery **S**ensing (**AQS**) system to automatically determine the best view for visual sensing to take the visual query.

Fully automatic AQS is difficult to achieve for the initial query because the user location is unknown to the system a priori. Thus location-specific information is missing for determining the best view for query. Although some prior information (like GPS or previously seen locations plus trajectories) may be available for predicting the likely current locations, it is not always reliable. As a result, we adopt a more pragmatic goal, aiming at significantly improving the success rate of the second query if and after the first query fails. This will bring major improvements over today's systems, in which users are often left helpless and have to appeal to repetitive trials and errors after the first query fails.

Specifically, we present an innovative AQS system that includes two major components of both offline salient view learning and online active query sensing:

- First, we have developed automatic methods for assessing the "saliency" of views associated with a location. Such saliency measures are derived from offline analysis of the matching scores between a given view and other images (including those of the same location and different locations), unique image features contained in the view, or combinations of these two. We found the proposed saliency measure can provide much more reliable predictions about the best query views, compared with alternatives using random selection or the dominant view.
- Second, we use the first query as a "probe" to narrow down the search space and form a small set of candidate locations, from each of which the first query is aligned, then the optimal view change (*e.g.*, turn to the right of the first query view) is predicted[2] in order to sense the discriminative view for the next query.

The proposed AQS system is general, without requiring special hardware on mobile devices. It is applicable to any visual location search system in which each location is associated with multiple reference views. Using a large database of about 50,000 locations (300,000 street view images), we will demonstrate the power of the proposed AQS system in significantly reducing the location search error.

The rest of the paper is organized as follows. In Section 2, we present a case study over the NAVTEQ street view data set as further justification of the proposed Active Query Sensing idea. Section 3 includes an overview of the AQS system and constituent components, including the basic visual matching component suitable for large scale data sets, the offline salient view analysis, the online query alignment, and the best query view suggestion. Section 4 shows the quantitative evaluation and in-depth analysis of the results. We discuss related works and summarize novel contributions of this work in Section 5. Finally, in Section 6, we present conclusions and open research problems.

2. CASE STUDY AND PROBLEM JUSTIFICATION

To further motivate the problem, we present a brief summary of a case study of mobile location search using a NAVTEQ data set captured in Manhattan of New York City. Details about implementations of the visual search system will be described in Section 3 of the paper.

2.1 NAVTEQ 0.3M NYC Data Set

The data set consists of close to 300,000 images of about 50,000 locations in Manhattan collected by the NAVTEQ street view imaging system during September and October of 2009 over 5 routes, as shown in Figure 2. Each geographical location contains six surrounding views separated by 45 degrees. Note the right rear view and the rear view are

[2]For each location in the database, we have multiple image samples with parametric viewing angles.

Figure 4: Location dependent visual distinctiveness. (a) Percentage of locations in the test query set (cropped from Google Street View) with different numbers of searchable views. About 87% of locations have a partial set of views (1 to 5) that can be used to successfully recognize the right locations, confirming the need of prudent query view selection. (b) Distribution of the test query locations with different degrees of searchability over the NYC Manhattan map.

Figure 5: Two exemplar image panoramas in the NAVTEQ NYC data set.

not covered in this setup. The locations are imaged at a four meter interval on average. The view orientations are shown in Figure 3 (a) and a typical example of six photos of a location is shown in Figure 3 (b). In addition, for each location there is also a panorama image captured by the panoramic camera (Ladybug 3), as shown in Figure 5. Both reference images and panoramas are geo-tagged based on the GPS system plus the Inertial Measurement Unit (IMU) and Distance Measurement Instrument (DMI) on the mobile imaging vehicle.

In this preliminary case study, we simulate the mobile location search scenarios by creating a test set using the Google Street View interface. We manually cropped queries from Google Street View (Figure 7) in 226 randomly chosen locations covered by the above mentioned routes in NYC. Although such test images are less ideal compared to real photos captured by mobile phones, we use them for initial testing since they are quite different from the reference images in the NAVTEQ data set and many challenging conditions (like occlusion and time change) are presented.

For each location, six query images are cropped from viewing angles similar to the view orientations used in the database (as shown in Figure 3 (a)). This results in 1,356 images with angles and ground truth locations tags. For each query image, we consider a returned reference image as relevant if it has visual overlap with the query image, as shown in Figure 3. Due to the fixed geometry and interval used in image acquisition, the ground truths of each query result set can be computed without the laborious manual annotation. Depending on the viewing angle of a reference image, the number of relevant images in the database varies, *e.g.* the front view has more relevant images as its field of view overlaps with more images. Note this is different from the ground truth definition for locations that based on a certain distance threshold.

2.2 Location Dependence and Need of Active Query Sensing

For each simulated query image from each of the random locations, we find the most likely location (among the 50,000 locations in the database) that has the highest aggregated matching scores between the query image and the multiple views associated with the location. Details of the matching process will be described in Section 3.1. A returned location is considered correct if it is within a distance threshold from the query location. Setting the appropriate threshold needs to consider several factors, such as the application requirements and the visual overlap of reference images. We set it to 200 meters in this initial study since two locations may still share overlapped views at this distance in our data set. Figure 4 (a) shows the proportions of the test locations that can be correctly recognized, broken to groups that can be recognized by a different number of searchable views (0 to 6). 11.1% of the locations cannot be recognized using any of the views cropped from the same location on Google Street View. This appears to validate our assumption that there is sufficient content difference between the reference images in the NAVTEQ database and the simulated test images cropped from Google Street View. Only 3.5% of the locations can be correctly recognized by all of their six constituent views. The rest of locations (85.4%) are recognizable only with a subset of the views, with most locations being searchable by 3 of the 6 views. This supports our hypothesis presented in Section 1 - each location has a unique subset of preferred views for recognition and the problem of automatically determining such preferred views are interesting and practical. In addition, only 42.1% of the test query images from Google Street View were successful. This indicates that if a mobile user randomly selects views for location search, more than half of the times he/she will not get correct results (even though the distance threshold was set to be quite generous, say 200m). This again validates the motivation of AQS.

Another important finding is *Location Dependence*. Figure 4 (b) shows the distribution of the locations with different numbers of searchable views. It's clear that different locations have different degrees of "difficulty" - some have more searchable angles than others. Additionally, locations of the same search difficulty do not significantly cluster together. This implies that any active query suggestion solution needs to be location adaptive and take into account the unique characteristics of the query in an online fashion. Finally, we also find there is no single dominant view that can successfully recognize all locations, though some views (*e.g.*, the front view) are more effective than others.

In addition, this Location Dependence assumption forms another basis of our Active Query Sensing: Only in the case

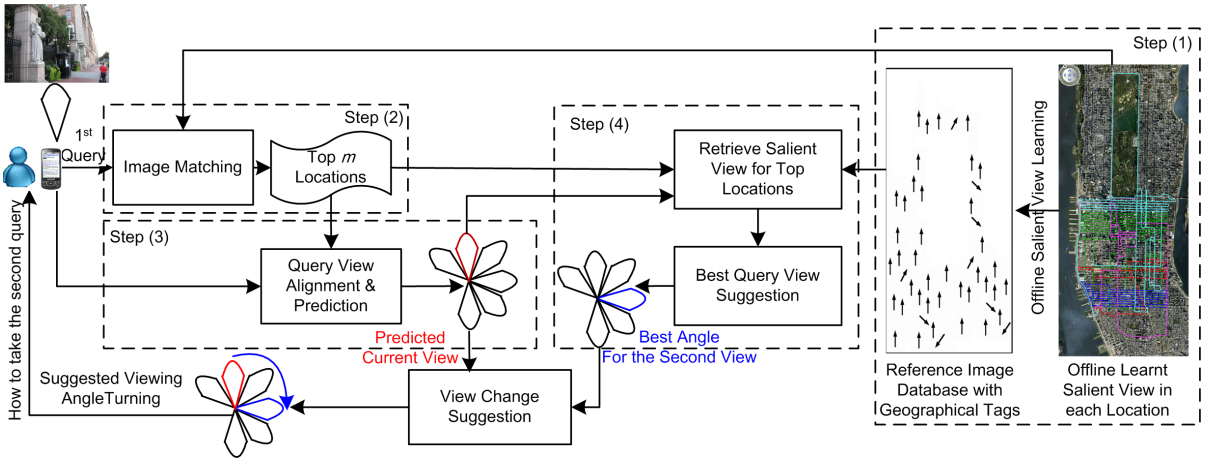

Figure 6: The architecture and process flow of the proposed Active Query Sensing system.

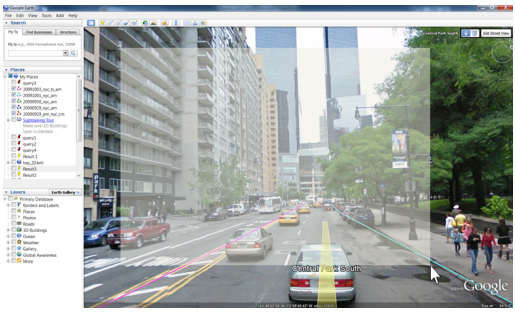

Figure 7: The Google Street View interface is used to crop images to simulate online mobile queries.

that the mobile visual search results are "Location Dependent", we need Active Query Sensing. Otherwise we can always suggest the mobile user a "dominant" view, say pointing at the middle of the street (the front view) *etc.* We also find that only 3.5% of locations can be correctly searched when using a random query viewing angle. It means random selection of query view has a big gap from this "search bound", which is the ultimate goal of our active query sensing.

3. ACTIVE QUERY SENSING

Two fundamental components are needed in developing a mobile AQS system to actively suggest the best query view for searching each location that user might be interested in. First, an offline analysis is needed to discover the best query view(s) for each location indexed by the system. Second, an online mapping process is needed to estimate the most likely view of the first query (which has failed). Afterwards, an online step based on the majority voting principle is used to suggest the most beneficial view change to the user in order to form the next query.

Figure 6 shows the overall architecture and the process flow of the proposed AQS system. Given the first query submitted by the user, the image matching component (Section 3.1) computes the scores for the reference images and returns the top candidate locations. Assuming the top returned location is incorrect[3], the active view suggestion process kicks in. First, the most likely view of the current query image is estimated through coarse classification of the query image to some of the predefined views (*e.g.*, side, front-back, *etc.*) and then refined by aligning the query image to the panorama or the multi-view image set associated with each of the top candidate locations (Section 3.3). Such alignment process can also be used to filter out outlier locations that are inconsistent with the majority of the candidate locations in terms of predicting the current query view. The filtered candidate location set is then used to retrieve the salient view associated with each possible location, which has been pre-computed offline (Section 3.2). A majority voting process is used to determine the best query view for the next query (Section 3.3). The difference between the best query view and the predicted current view is then used to come up with the view change suggestion to the user, who may then turn the camera phone by following the suggested change.

We go through each of the components described above in the following subsections.

3.1 Million Scale Image Matching

We achieve approximate image matching in a million scale image database using Bag of visual Words (BoW) with inverted indexing technique. Our implementation basically follows the algorithm described in [11], which uses a hierarchical tree based structure for efficient codebook construction and visual local feature quantization. In addition, we also test the incorporation of multi-path search [7] and spatial verification [10] to improve the accuracy. The experimental result over a validation set is shown in Figure 8. Our current system uses the six high-resolution views captured by individual cameras, though image crops from panoramas can be incorporated as well. We summarize below some of the specific findings about implementations over the NAVTEQ NYC data set.

Local Feature Extraction: Both interest point detection and dense sampling are tested in building our search

[3]We assume only the top location is returned to the mobile user and the user is able to realize the response is actually wrong by inspecting the scenes of the predicted location or verifying the associated information (*e.g.*, recommended stores missing in the neighborhood)

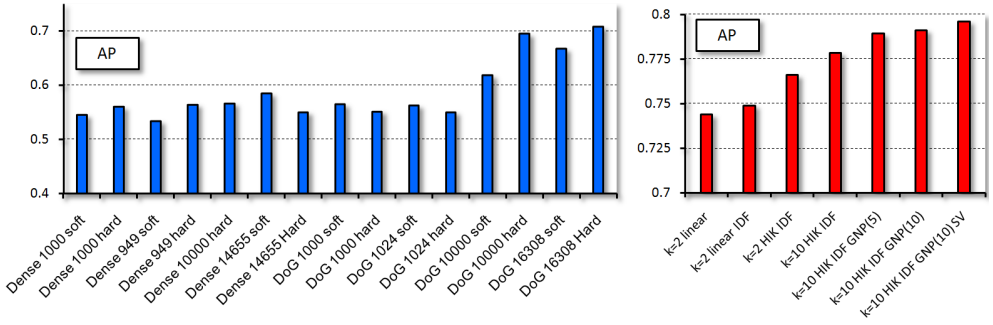

Figure 8: Comparison of image matching performance with different configurations. (a) Small codebook configurations with varying local detector, codebook size, hierarchical clustering, and quantization; (b) Million sized codebook configuration with varying branching factor, histogram intersection kernel, GNP search [7], IDF, and spatial verification [10].

system. The former is based on Difference of Gaussian [12], while the latter is based on multi-scale sliding window with 3 scales and fixed steps, producing approximately equal numbers of local features. As shown in Figure 8 (a), DoG outperforms dense sampling under different configurations of codebook sizes and quantization methods.

Local Feature Clustering: We use the Hierarchical K Means Clustering to build a million scale Vocabulary Tree [11]. There are two basic settings in building the Vocabulary Tree: (1) Branching Factor B controls how many clusters are built to partition a given set of local features into its lower hierarchy; (2) Hierarchical Layer H controls the number of hierarchical layers in the tree. There is a tradeoff between speed and quantization accuracy in choosing different B and H values. With empirical validation, we set $B = 10$, and $H = 6$ to construct our final codebook of approximately one million codewords.

Quantization: Soft quantization has been found in [13] to improve the retrieval precision. In our case, soft quantization performs better when the codebook size is small. But as the codebook size increases, the performance of soft quantization degrades, and is outperformed by hard quantization in our million-scale codebook configuration. Greedy N-Best Path (GNP) [7] is used to rectify the quantization errors by searching multiple paths over the quantization tree. We found that by using GNP of 10 paths we achieved a slight gain in average precision (2%).

Spatial Verification: We further incorporate the spatial matching proposed in [10], which considers a point in one image matches a point in another image only if a sufficient number of nearby local features are also matched. This process is easy to be implemented, but the performance improvement is minor (only 0.5%), perhaps due to the precise matching capability of the large codebook (one million).

Inverted Indexing: Finally, we implement Histogram Intersection Kernel (HIK) [14] matching with inverted indexing in our search system to ensure the scalability to the million scale database. Once a certain amount of local descriptors from one query are assigned into a given visual word, we assign all images indexed by this visual word a certain score.

The final million-scale codebook configuration is shown in Table 1. The performance of the final baseline system is satisfactory and comparable with the state of the art. It achieves an average precision at 0.79 over a validation subset from the NAVTEQ data set. Over the widely used UKbench benchmark [11], it even slightly outperforms the results in [11] by a large-scale vocabulary tree.

Table 1: The final one million codebook configuration

Component	Choice
Local Feature	DoG + SIFT [12]
Clustering	Hierarchical K Means ($B = 10, H = 6$)
Quantization	Hard Quantization with GNP [7] $N = 10$
Inverted Indexing	HIK kernel based
Spatial Verification	Neighborhood Voting [10]
Word Frequency	TF-IDF

It is important to note that image matching is not the focus of the paper. Our proposed AQS approach is general, independent of the image matching subsystem used as long as the assumption about the location-dependent preferred query view holds. In addition, it can be applied to any visual location search system without requiring any additional hardware on mobile devices.

3.2 Offline Salient View Learning

Given a location of interest, which view is the best candidate for matching the reference images in the database and recognizing the true location? As discussed earlier, each location has certain views that are more salient and can be used for successful retrieval. Figure 9 shows another example to illustrate the concept. Intuitively, two approaches can be considered - *content-based* and *training performance based*. The former explores the unique attributes contained in each view such as distinct objects, local features, *etc.*, while the latter predicts the test performance by assessing the query effectiveness over a training data set. We have developed methods based on both ideas. Note in the discussion below, we assume the continuous space of view can be appropriately discretized, for example, to a finite set of choices (*e.g.*, six angles used in the NAVTEQ data set).

Content based View Saliency Prediction: With the BoW representation, distinctive visual words have a better discriminating power than words that appear in most locations in the databases. This concept has been well explored in the information retrieval and image classification literatures, following the concept of TF-IDF (Term Frequency - Inverse Document Frequency). A word is considered more distinctive if its frequency of occurrence in the database images (documents) is low. Extending this concept, we define

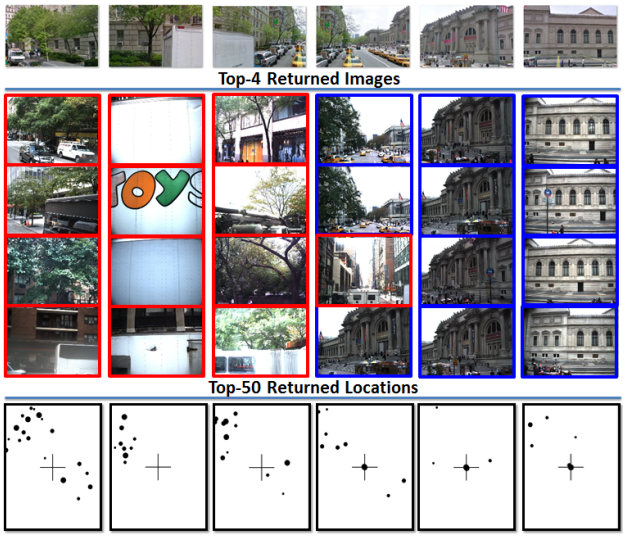

Figure 9: Top: the six views of the Metropolitan Museum of Art in New York City; Middle: The top 4 returned images for each view (blue: correct, red: incorrect); Bottom: the geographical distribution of returned locations of different views. Cross indicates the actual query location while locations of the matched images are shown in different sizes according to the rank orders.

a TF-IDF related content-based feature as follows:

$$F(k) = count(word_i | IDF(word_i) > k/K \times IDF_{max}), \quad (1)$$

where $k = 1, 2, ..., K-1$. If we set K to be 10, then the above feature accounts for the number of visual words whose IDF exceed certain thresholds (up to 90% with 10% increment). Intuitively, images of salient views will have more words with high IDF than images of non-salient views.

We train a Support Vector Machine (SVM) based classifier and use its classification score to predict the saliency of an image. A subset of geo-tagged locations sampled from Google Street View (described in Section 2.1) was used as a labeled training set to train the SVM classifier. Since the feature dimension is kept low (10 if $K = 10$), a training set of such a size is adequate.

Training Performance based View Saliency Prediction: As described in Section 2, each location is associated with a finite set of reference images captured in different views. Each of the reference images can be used to query the database and evaluate its capability in retrieving related images of the same location, or other locations sharing overlapped scenes. Although there is always a gap between such training performance and the real test performance when querying by new images that have not been seen before, the score distributions of relevant (positive) images and irrelevant (negative) images can serve as an approximate measure.

An ideal score distribution is the one that has maximal separation between the scores of the positive results and those of the negative ones, like the first score distribution shown in Figure 10. Scores that have very small separation (second distribution in Figure 10) or mixed results (third and fourth distributions in Figure 10) do not generalize well. To approximate the robustness of such query results, we develop two simple methods. The first one is based on the

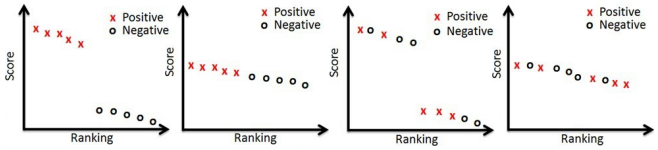

Figure 10: A good score distribution is the one that can maximally distinguish the positive results (red) from the negative (black). Several examples of score-rank distributions are shown, with the first one being the most reliable one and the last being the least (second one with the identical AP to the first).

commonly used metrics, Average Precision (AP), as below:

$$AP = \frac{\sum_{r=1}^{N_{relevant}} P(r)}{N_{relevant}}. \quad (2)$$

$N_{relevant}$ is the number of relevant documents to the current query; r is the the rth relevant document; $P(r)$ is the precision at the cut-off rank of document r. In the literature, there are some subtle variations in definition of AP. The one used above sometimes is also called full-length AP.

The other measurement, called *Saliency* as defined below, is similar to AP with several important modifications. First, we compute the ratio of the positive score statistics to that of the negative scores. Second, we incorporate the actual score values in the measure. Both modifications are important, as we are interested in the score separation between the positive and negative classes:

$$Saliency = \frac{\sum_{i=1}^{N} \sum_{j=1}^{i} score(j) rel(j)/i}{\sum_{i=1}^{N} \sum_{j=1}^{i} score(j) \overline{rel}(j)/i}, \quad (3)$$

where N is the number of returned locations, which can be a fixed size or adjusted based on the number of positive samples[4]. $score(j)$ is the location matching score, which is the maximal score of its six views. $rel(j)$ is the relevance judgement of the jth returned location, which gives 1 for correct locations and 0 for incorrect. Conceptually, other statistical measures, such as KL Divergence, can also be used.

Note the numerator in Equation 3 is very similar to that of AP, except the score values are used instead of binary values (1 for positive and 0 for negative) and the inner average is repeated for every sample, not just for the positive points. Despite the simplicity of the above saliency measure, surprisingly it has a significantly superior prediction accuracy to all other measures (Section 4).

The offline measures of saliency for each view can also be used to "grade" the searchability of each location, as discussed in Section 2. Based on the search results of the associated views, a location can be categorized into one of the following groups.

- *View-Independent Confident Location:* Users can take photos in any arbitrary view to find this location.

[4]In our experiments to be reported later, we use 3 times of the number of positive points so that we can compute the statistics from the top result set covering both sufficient positive and negative results.

Algorithm 1: Online view estimation and Active Query Sensing procedure

Given query q, get the top N most likely locations (Section 3.1).
if *the first location is incorrect* **then**
  Obtaining candidate location set $\hat{L}$ {
    Remove locations in the top-N set that are geographically close to the first location;
    Predict the viewing angle of the first query using GIST + SVM with the voting refinement (Equation 4);
    Discard the outlier locations with predicted viewing angles inconsistent with the one obtained before;
  }
  Majority voting within $\hat{L}$ {
  for $l \in \hat{L}$ **do**
    Retrieve the saliency of the remaining views Θ;
    Estimate the camera movement to the most salient view in l;
  end
    Majority voting to determine the best view for the second query.
  }
end

- *View-Dependent Searchable Location:* Users can take photos in certain views to find this location.
- *Difficult Location:* Users cannot find this location no matter which view he/she use to form the visual query.

There is room for improvement with the above analysis. First, the offline analysis is only limited to the discrete views that have been indexed in the database. In practice, users can sample the view space in a much more flexible manner. Second, there is likely to be generalization gap between offline analysis based on training performance and the real-world online testing. Nonetheless, the offline analysis offers an approximate yet systematic process of discovering the preferred query views for all of the searchable locations.

3.3 Online View Estimation and Active Query Sensing

We describe the modules for online view estimation and active query view suggestion in this section. The main process is summarized in Algorithm 1. Given the first query that fails to recognize the correct location, the objective is to develop automatic methods that can estimate the likely view captured by the first query, and from the candidate location set, discover the best view for the next query.

Using the image matching subsystem, we first find a small set of top-N most likely locations. Locations close to the first location are removed as the first location has been judged as incorrect by the user. Next, we employ a SVM classifier to assign the first query image to one of few rough orientations, followed by refinement based on image matching. Algorithm 1 shows the working pipeline of our online active query sensing. Some key components of Algorithm 1 are explained in detail below.

Viewing Angle Estimation: Although the visual content in different views of the location database could be very diverse, there exist general patterns differentiating each other. For example, the side views tend to contain more features related to buildings, trees, and sides of parked vehicles, while other views (*e.g.*, front) has more attributes like sky lines, streets, and front/back views of vehicles. Such differences tend to be holistic reflecting the overall characteristics of the scenes, thus motivating the choice of GIST descriptor [15] for view classification.

We train the SVM classifiers offline based on GIST features extracted from 3000 images (500 for each view) randomly chosen from our database. Given an online query, we use the classifier to predict the current viewing angle in one-vs-all manner. GIST feature is efficient in describing the global configuration of images. However, in our case, as shown in Figure 3, View 1 and View 5, as well as View 2 and View 6 have very similar visual appearance. To address this problem, we further use a refinement step based on the image matching results:

$$\arg\max_{\theta_i} \sum_{l} P(\theta_i|l,q)P(l|q), \tag{4}$$

where θ_i is a candidate view under consideration, $P(\theta_i|l,q)$ is the matching score between query q and view i of location l. $P(l|q)$ can be modeled by location score distribution based on query q, combined with additional metadata such as GPS or the history data about the user locations. The default $P(l|q)$ is a positive constant for top matching locations (candidates), and zero for others.

This refinement works because similar views, even from different locations, typically have similar visual contents *e.g.*, skylines, side of a truck *etc.*, which are more likely to be included in the top image match results. This phenomenon is also observed in the recent work of [5] for street view image search. So the final angle estimation method is based on the combination of both local feature (SIFT for image matching) and global feature (GIST for SVM classification). This approach is fairly robust in our application scenarios, which will be shown experimentally later. It should be noted that when the solution space for view prediction (and alignment) is large, a more sophisticated correspondence matching method, such as RANSAC, may be needed to reliably align the query image to the panorama associated with each location. However, the impact on the speed will need to be carefully investigated as a fast response is needed for such online applications.

Once the current query view is estimated, we further use it to filter out the outlier locations that produce inconsistent view estimation. Empirically, we found that this step is very useful.

Majority Voting for View Suggestion: Given the filtered candidate set of the locations, we use a majority voting scheme to predict the most beneficial view to be used as the next query. It can be expressed as:

$$\arg\max_{\theta_i \in \Theta} \sum_{l \in \hat{L}} H_{saliency}(\theta_i|l)P(l|q), \tag{5}$$

where $H_{saliency}(\theta_i|l)$ outputs 1 if the saliency of view θ_i and location l is greater than a threshold. $P(l|q)$ is modeled by the location score distribution after removing outlier locations with inconsistent view estimations. Setting $P(l|q)$ as a positive constant for all top-N locations and zero for others leads the above equation to a majority voting.

The scheme takes into account the saliency of each view with respect to each remaining candidate location. With the predicted best query view and the estimated viewing angle of the current query, we can then make suggestions to the

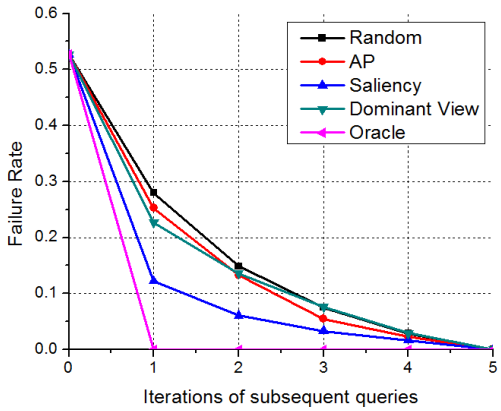

Figure 11: Failure rates over successive query iterations based on different active query sensing strategies: (1) Random view suggestion, (2) AP-based view suggestion, (3) Saliency based view suggestion, (4) Dominant view suggestion, and (5) Oracle view suggestion.

Table 2: Percentages of successful location search using query images from different views

$View_1$	$View_2$	$View_3$	$View_4$	$View_5$	$View_6$
0.45	0.26	0.58	0.65	0.53	0.35

user to inform him/her of the best way of turning or moving the camera phone for the subsequent visual search.

4. EVALUATIONS

We evaluate the performance of the components and overall system of the proposed active query sensing approaches for mobile location search, using the NAVTEQ NYC data set (about 300,000 images, 50,000 locations). The test queries are the 1,356 images over 226 locations randomly cropped from the Google Street View interface as described in Section 2.1. Out of the 226 locations, 11.1% were found to be unsearchable by any of the views and thus were discarded. The remaining 201 locations are searchable by at least one viewing angle. The proportions of searchable locations by various numbers of views are shown in Figure 4.

We first analyze the "dominance" of each view of the query set. Table 2 shows the percentages of successful searches over the 201 test locations by each of the six views. It is interesting to see each view has a reasonable chance of success (between 35% and 65%) while View 2 has the lowest rate. This is due to the relatively low quality of the camera used for View 2 in the database. View 4, the one pointing to the front of the imaging vehicle, has the highest success rate as it appears to cover the most visual objects (*e.g.*, buildings on both sides), as well as the most distinctive features such as sky lines.

Salient View Prediction: Next, we evaluate the performance of predicting search robustness using offline saliency analysis, described in Section 3.2. Table 3 shows the percentages of successful search over 201 test locations by using different methods to predict the best view for each test location. We compared two types of proposed methods - training performance based: AP and Saliency, and content based: IDF SVM[5] classifier, as described in Section 3.3, against

[5]Since the content-based SVM classifier requires supervised training, we adopted a five-fold cross validation process to partition the test query set for separate training and testing.

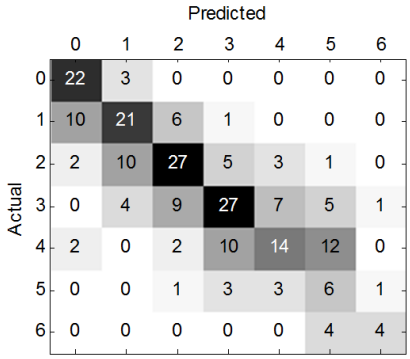

Figure 13: Confusion matrix for location search difficulty prediction (the predicted number of successful views vs. the ground truths)

the random view selection and the one that always chooses the most dominant view (View 4). Surprisingly, the content-based classifier did not perform as well as expected, possibly due to the large content confusion among different locations in the NYC data set. Among all the competing approaches, the saliency measure (as defined in Equation 3) incorporating the score statistics ratio between the positive and the negative groups turned out to achieve the highest performance (84%) with a large margin over other approaches (the next best one is 68% by AP).

Table 3 shows the robustness comparison of different approaches. The experiment is performed on all the 201 searchable locations. For each location, we pick up the external mobile query with the most salient angle predicted offline. Then, we test whether or not we can correctly find the true location using this query. This table shows the robustness validation of different view discrimination measurements. For content based approach (SVM based on statistics of distinctive visual words), we set $K = 10$ in Equation 1. For each testing location, we use an SVM classifier to get the probability based classification results, and the viewing angle with largest probability to be discriminative is predicted to be the most salient viewing angle. The result is based on a five-fold cross validation on the test set. As shown in Table 3, our saliency measurement obtains the highest score.

Query View Estimation: For the module of view estimation of test queries, we found the GIST based SVM classifier was able to achieve 86.5% classification accuracy over the 1,356 test image queries if we are only concerned with View 1-4. When Views 5 and 6 are added, they cause confusion with views of highly similar content (View 1 with View 5, and View 2 with View 6). This is reasonable due to symmetrical nature (180 degrees opposite direction) and thus similar visual content between the two views in each of the symmetrical pairs. To resolve this, we applied the maximal voting scheme based on image matching scores (as described in Equation 4). It helps to keep the view estimation accuracy as high as 82.1% among all six viewing angles.

Active Query View Sensing: Finally, we evaluate the effectiveness of the proposed AQS system in helping users choose the best view for subsequent queries after the first query fails. We initialized the simulated system with a randomly chosen viewing angle in the first visual search. As

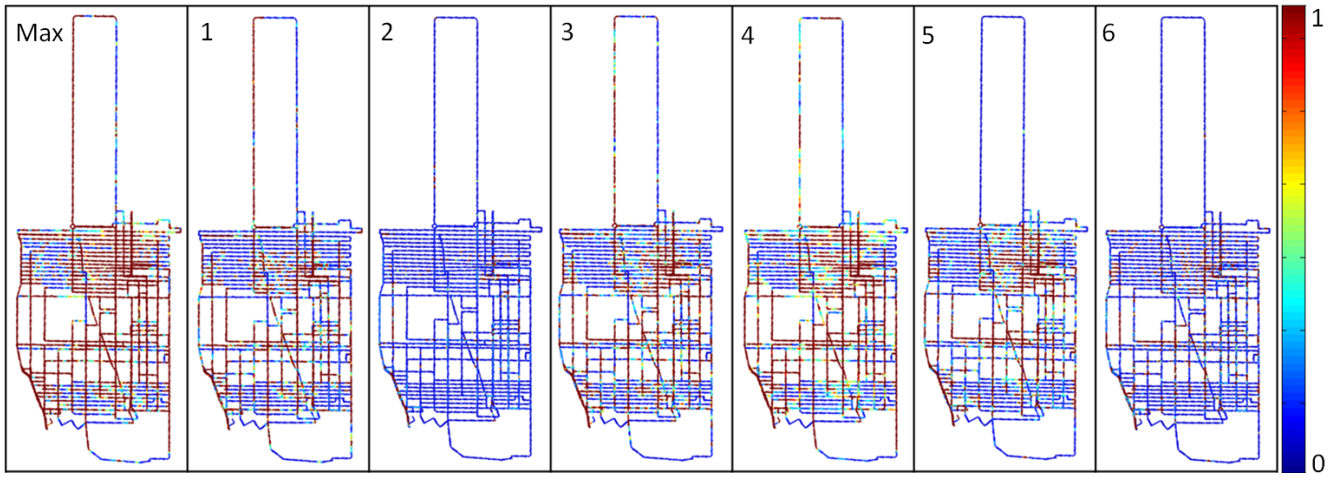

Figure 12: Geographical location confidence (measured by saliency) distribution in New York City with respect to different views, where "Max" denotes assigning the maximal confidence among six views to each location.

Table 3: Location search accuracy using different methods in predicting the best query view for each location

Method	Random	Dominant Angle	Content Based	AP	Saliency
Percentage correct prediction	0.4735	0.6517	0.5521	0.6816	0.8458

discussed earlier, only 47% of the random query images succeeded, resulting in a 53% failure rate after the first query (as shown in Figure 11). We then evaluated the performance of reducing the failure rates in subsequent queries by using different active query strategies, including the proposed AQS method based on view saliency measure, other methods based on AP, dominant view, random selection, and the oracle scheme which knows the correct answers and always chooses a successful view after the first query. The performance gain achieved by the saliency based AQS scheme is quite impressive – 12% error rate after only one additional query compared to the next best one 23% by the dominant view. As the number of query iterations increases, the proposed AQS solution consistently maintains significant edge over all other methods. Eventually, the error rate drops to zero for all methods since this test set consists of searchable locations only (*i.e.*, at least one view can be used to recognize the location).

Location Difficulty Level Prediction: We further evaluate how well the proposed saliency measure (described in Section 3.2) can be used to predict the difficulty level of each location in terms of location recognition. Ideally, we would like to be able to generate the confidence distribution map similar to the one shown in Figure 4. Accurate prediction of such a confidence distribution map can facilitate development of very interesting applications. For example, users may use such information to determine how much to trust the visual search based location information when he/she has additional information (like GPS) to roughly know the geographical region he/she is located in. Figure 12 shows the distributions of saliency for each viewing angle as well as their maximum among all views of each location computed from the NYC data set. To avoid outliers caused by poor image quality or missing data, a simple Gaussian smoothing was also used. The resulting distribution maps confirm the location dependence assumption we have made in the paper. By thresholding the saliency values in each view, we further predicted the number of views that can be used to successfully search each location. We compared the estimated number of successful views against the ground truths associated with the 1,356 test query images cropped from Google Street View. The confusion matrix, shown in Figure 13, confirms the effectiveness of the proposed saliency based estimation.

5. RELATED WORK

Many exciting research directions are being pursued to advance the field of mobile visual search. For mobile location recognition, [16] adopted sequential matching of more than one reference views to estimate the pose and motion direction. [7] presented a method for large-scale location recognition based on geo-tagged video streams with multi-path search over a vocabulary tree. [17] also adopted a vocabulary tree based approach for real-time loop closing. [18] described an approach to recognize locations by mobile image search, which utilized a hybrid color histogram to compensate its original ranking results. [19] presented a system to identify landmark buildings based on image data, metadata, and other photos taken within a consecutive 15-minute window. [20] adopted a structure-from-motion technique to build 3D scene models for vision-based city scene localization. Compact descriptors were proposed in [6] to greatly reduce the delay in computation and communication. Discovery of vanishing points and identification of facades were proposed in [9]. Confusing features such as trees and road markings were removed based on geo-tag supervision [5]. Fusion of multiple image representations (facade-aligned and viewpoint-aligned) was used to improve the search accuracy [8]. Despite the impressive progress shown, none of these works address the active sensing problem to help users take a better second query in the subsequent location search.

In image search, there have been long standing works in relevance feedback [21]. The basic idea is to use iterative labeling of the top returned results to learn a refined ranking

function [21][22]. By dealing with the user gap [23], there are related works in helping and suggesting users to better formulate their queries. For instance, [23] presented a work for visual query suggestion, which provides images to help users express their search intent despite the ambiguity in the textual queries. One closely related topic is active learning [24], in which the system tries to find the most informative sample to collect user feedbacks in order to learn an improved decision function. Our problem is different in actively determining the best view to sense in the real-world scene (rather than the sample space in the database) and in trying to recognize the correct target (instead of a decision).

Our work is also related to a few topics in information retrieval. For instance, [25] proposed a learning scheme to predict the query difficulty, which measures the contribution of each query term to the final result set. [26] learnt a support vector machine to predict the query success based on TF and IDF of query keywords. [27] studied several predictors of the query performance based on the IDF statistics of the query terms. The suggestion models mentioned above are typically learnt from the statistical distributions of the document corpus and query logs. On the contrary, our paper focuses on strategies for visually searching the physical surroundings of real world geographical locations.

6. CONCLUSIONS AND FUTURE WORKS

Active Query Sensing aims to help the mobile user take a successful second query once the first query fails (more than 50% of chance as discussed in the paper). It informs the mobile user how to sense his/her surrounding environment so that the captured image is most distinctive and can be used to recognize the location. To achieve this goal, we develop a novel Active Query Sensing system that actively discovers the best query strategy and suggests the best sensing view after the first visual query fails. To the best of our knowledge, this is the first effort in addressing the user needs by iterative refinements of mobile visual search. We develop several saliency measures based on score distributions to predict the robustness of each query view and the search difficulty of each location. In addition, we develop an online process for estimating the viewing angle of an unseen novel query and suggesting the best view change prior to forming the next query. Using a large street view image data set (0.3 million images over 50,000 locations) of the New York City, our system shows a performance gain as high as two-fold: Our sensing strategy can reduce the failure rate of mobile location search to only 12% after the second query. Our statistical saliency measure can also be used to robustly predict the difficulty of visually recognizing individual locations, allowing users to decide where the mobile search service can be best trusted.

The future work includes incorporation of other image representations *e.g.*, panorama and range data, to provide more flexibility in view suggestion (translation, zoom *etc*). The proposed system also has great potential for applications beyond location search, such as products and landmarks.

Another future work is to provide intuitive user interfaces to maximize the utility of the proposed AQS search system. For example, when the user is not sure about the correctness of the search result, we may provide extra information (besides just the images of the predicted location) such as street names, landmarks in the vicinity, *etc.*

7. ACKNOWLEDGEMENTS

We would like to thank NAVTEQ for providing the NYC image data set, Dr. Xin Chen and Dr. Jeff Bach for their generous help. This work has been supported in part by NSF awards #CNS-07-51078 and #CNS-07-16203.

8. REFERENCES

[1] `http://www.snaptell.com/`.
[2] `http://www.pointandfind.nokia.com/`.
[3] `http://www.kooaba.com/`.
[4] J. Knopp, J. Sivic, and T. Pajdla. Avoiding confusing features place recognition. *ECCV*, 2010.
[5] B. Kaneva, J. Sivic, A. Torralba, S Avidan, and W. Freeman. Matching and predicting street level images. *Workshop on Vision for Cognitive Tasks, ECCV*, 2010.
[6] B. Girod, V. Chandrasekhar, D. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. Tsai, and R. Vedantham. Mobile visual search. *IEEE Signal Processing Magazine*, July, 2011.
[7] G. Schindler and M. Brown. City-scale location recognition. *CVPR*, 2007.
[8] D. Chen, G. Baatz, K. Koser, S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. *CVPR*, 2011.
[9] G. Baatz, K. Koser, D. Chen, R. Grzeszczuk, and M. Pollefeys. Handling urban location recognition as a 2d homothetic problem. *ECCV*, 2010.
[10] J. Sivic and A. Zisserman. Video google a text retrieval approach to object matching videos. *ICCV*, 2003.
[11] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *CVPR*, 2006.
[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.
[13] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost quantization: Improving particular object retrieval large scale image databases. *CVPR*, 2008.
[14] J. Wu and J. Rehg. Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. *ICCV*, 2009.
[15] A. Oliva and A. Torralba. Modeling the shape of the scene a holistic representation of the spatial envelope. *IJCV*, 2001.
[16] W. Zhang and J. Kosecka. Image based localization urban environments. *3DPVT*, 2006.
[17] E. Eade and T.Drummond. Unified loop closing and recovery for real time monocular slam. *BMVC*, 2008.
[18] T. Yeh, K. Tollmar, and T. Darrell. Searching the web with mobile images for location recognition. *CVPR*, 2004.
[19] D. Crandall, L. Backstrom, and D. Huttenlocher. Mapping the world's photos. *WWW*, 2009.
[20] A. Irschara, C. Zach, J. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. *CVPR*, 2009.
[21] Y. Rui, T. Huang, and S. Chang. Image retrieval current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 1999.
[22] R. Datta, D. Joshi, J. Li, and J. Wang. Image retrieval ideas, influences, and trends of the new age. *ACM Computing Surveys*, 2008.
[23] Z.-J. Zha, L. Yang, T. Mei, M. Wang, and Z. Wang. Visual query suggestion. *ACM Multimedia*, 2009.
[24] S. Tong and E. Chang. Support vector machine active learning for image retrieval. *ACM Multimedia*, 2001.
[25] E. YomTov, S. Fine, D. Carmel, and A. Darlow. Learning to estimate query difficulty. *SIGIR*, 2005.
[26] K. Kwok, L. Grunfeld, H. Sun, P. Deng, and N. Dinstl. Robust track experiments using pircs. *TREC*, 2004.
[27] B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. *String Processing and Information Retrieval*, 2004.