LOCATION-BASED IMAGE RETRIEVAL FOR URBAN ENVIRONMENTS

Jerry Zhang, Aaron Hallquist, Eric Liang, and Avideh Zakhor
Department of Electrical Engineering and Computer Science, University of California, Berkeley
{zhangz, aaronh, ekliang, avz}@eecs.berkeley.edu

ABSTRACT

Image based localization is an important problem with many applications. The basic idea is to match a user generated query image against a database of geo-tagged images with known 6 degrees of freedom poses. Once this retrieval problem is solved, it is possible to recover the pose of the query image. A challenging problem in image retrieval is performance degradation as the size of the image database grows. In this paper we describe an approach to large scale image retrieval for user localization in urban environment by taking advantage of coarse position estimates available, e.g. via cell tower triangulation, on many mobile devices today. The basic idea is to partition the large image database for a large region into a number of overlapping cells each with its own prebuilt search and retrieval structure. We demonstrate retrieval results over a ~12,000 image database covering a 1 km^2 area of downtown Berkeley.

Index Terms – augmented reality, tagged images, image matching, image retrieval, visual landmark recognition

1. INTRODUCTION

Localization is an important problem in many civilian and military applications, such as enhancing the query image with additional metadata, or helping the user navigate from point A to point B in an unknown environment. The most commonly used form of localization for outdoor environments is GPS. However, in many situations, GPS signal is not readily available due to loss of line of sight to GPS satellites or due to adversarial jamming. Cell tower and Wi-Fi triangulation are yet another way of localizing cell phone users[1]; even though the accuracy of these methods is lower than GPS, they are less sensitive to occlusions in urban environments.

An alternate approach to localization is to match a user generated query image, for example via a mobile device, against an existing image database of a region, e.g. from Google's Street View, or Microsoft's StreetSide. Once the best match has been retrieved, the pose of the database image can be used to determine the pose of the query image supplied by the user [**1**]. There are a number of existing approaches to image based localization. [**2**] uses a vocabulary tree to perform large scale localization over 30,000 images covering a continuous 20km stretch of urban terrain. [**3**] also uses a vocabulary tree to perform large scale localization using 31,034 images from the Earthmine database, but incorporates 3D building information to preprocess building imagery into orthophotos.

In most existing large scale image retrieval systems, performance degrades as the number of candidate images in the database increases. [**4**] addresses this problem by using GPS information to localize users into a uniform region grid; a kd-tree is then constructed over a preprocessed set of features from nearby regions for device queries. In practice, GPS information is oftentimes not readily available, especially in urban environments, and is usually subject to noise. Thus to avoid retrieval performance degradation with image database size, we propose to take advantage of the approximate coarse localization available on most mobile devices, e.g. via cell tower triangulation, by decomposing large geographic areas into overlapping cells, much the same way as wireless operators divide a region into smaller cells to deal with the handoff problem; thus, each cell has its own mini image database with fewer pictures than the one corresponding to the entire region. We then find the best match to the query image against the smaller databases corresponding to these cells. The main advantage of such a "divide and conquer" approach is that it mitigates the performance degradation resulting from image retrieval against a very large database; as such, it scalable to arbitrarily large regions.

Using a coarse location approximation, the query image would only have to be matched against few cells, rather than all of them, making the problem more tractable. To determine the cells whose results are combined, we use the location and size of the "ambiguity circle" defined as the uncertainty in user location as specified by GPS or cell tower triangulation. The search cluster for each cell is built offline in order to ensure interactivity and real time operation; search results corresponding to selected cells are further combined and processed to retrieve the matched image to the user query. Once the best match for the query image is retrieved by combining the results from multiple cells, the pose of the "best" database image can be used to retrieve the pose and location of the user.

The outline of the paper is as follows: We describe our image retrieval approach, shown in Figure 1, in Section 2, go over our experimental setup and results in Section 3, and present our conclusions in Section 4.

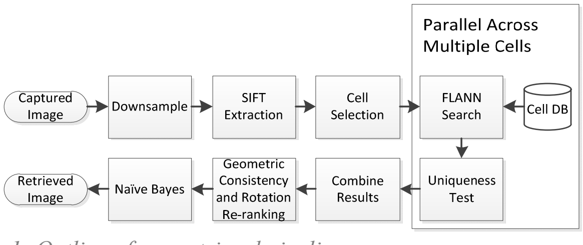

Fig 1: Outline of our retrieval pipeline.

[1] Specifically, in the United States the FCC e911 requirements specify for 67% (95%) of location requests to be localized within 50 (150) meters of the true user location.

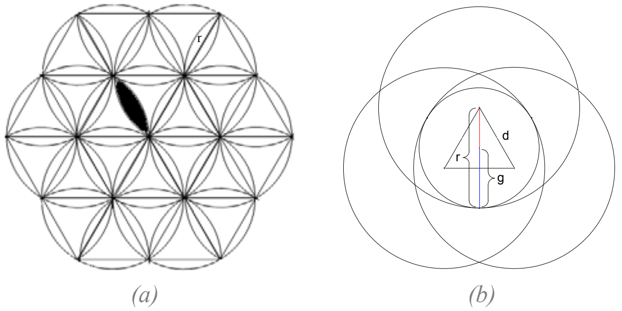

Fig 2: (a) The local search cells built over a hexagonal lattice. (b) Ambiguity circle of radius g fully contained in the intersection of 3 cells.

2. PROPOSED APPROACH

In urban environments, street-view datasets tend to have uniform spatial density. As such, we partition a city's geography into uniformly spaced overlapping cells of equal size so that the cells each contain approximately the same number of images. We do this by grouping local images into circular cells of radius r centered at the vertices of a hexagonal lattice, chosen for its symmetry and spatial packing properties, as shown in Figure 2(a).

Let α be an upper bound on the distance between two capture locations that capture the same view, and λ be the maximum discrepancy between a query's actual and reported location. To guarantee that a given query image has at least one cell containing all true matches, any circular region of radius $\lambda + \alpha$, when placed over the cell grid of Figure 2(a), must be fully contained within at least one cell. We refer to this condition as Single Cell Existence, or SCE, since satisfying it implies the existence of a true match, if one exists, within a single search cell. Intuitively, for SCE to be satisfied there must be sufficient overlap between cells. More specifically, geometric inspection of Figure 2(b) shows that this condition is satisfied if a circle of radius $g = \lambda + \alpha$, referred to as an "ambiguity circle", can be fully contained within the region of intersection of three adjacent cells. Let d denote the distance between the centers of the search cells. Even though sufficiently small values of d can satisfy SCE, in practice it is advantageous to use the largest possible value of d so as to minimize cell overlap and to reduce storage and computation overhead. In what follows we describe a way to find an upper bound on d that satisfies SCE.

As shown in Figure 2(b), the largest circular region, with radius g, which fits within the intersection of 3 overlapping cell must be internally tangent to each cell and its center must be equidistant to the centers of the 3 cells. Thus, the centers of the three cells form an equilateral triangle with sides of length d whose centroid is at the center of the circular region. Since the distance between the vertex and centroid of an equilateral triangle is $d/\sqrt{3}$, and the region is internally tangent to the cells, the radius of a cell must be $g + d/\sqrt{3}$. Thus for SCE to be satisfied, we must have:

$$d \leq \sqrt{3}(r - g) \tag{1}$$

The above relationship guarantees that a matching image to a query exists in the cell whose center is closest to its reported location; this is because the query's ambiguity circle is fully contained within that cell. If we further constrain the radius of every search cell to be equal to the distance between adjacent cells, i.e. $d = r$, then geometric inspection of Figure 2(a) reveals that every database image is always contained in either 3 or 4 search cells[2]. Specifically, any database image whose location falls within the "petal" region of the layout scheme, one of which is highlighted in Figure 2(a), is contained in exactly 4 cells. Similarly, database images whose location lie outside the "petal" regions are contained in exactly 3 cells. We exploit this observation in Section 2.2 to combine results from multiple cells.

2.1 Local Search Methods

We use a feature based approach similar to that of [**4**] [**5**] [**6**] for search in each local cell. Specifically, we pair SIFT features in the query image with those in the database images using a FLANN kd-tree of all features in the local cell [**7**] [**8**]. To determine whether a feature pair is a match or not, we use the *Uniqueness Test* outlined below. A score is then generated for each candidate database match as the number of feature matches between database image and a given query image. The database image that best matches the query image is the one with the largest score.

We now describe the *Uniqueness Test*. Similar to the multiple ratio test proposed in [**4**], we match the features in a query image I_q to features from a set of images $\boldsymbol{I_c} = \{I_1, I_2 \dots, I_m\}$ in a local cell c. While a kd-tree can provide us with the nearest neighbor database feature for any query feature, a nearest neighbor pairing alone is not always indicative of a "good" feature match. We propose a new method for evaluating the 'goodness' of a feature pair; provided there is a sufficiently large number of local cells in our database, for any query location we identify a dummy cell d with image set $I_d = \{I_1, I_2 \dots, I_n\}$ that the query location is known not to reside in. The features in I_c and I_d have been put into an approximate nearest neighbor kd-tree offline. As such, for each feature $f_q \in I_q$ we can, in parallel, compute its nearest neighbors f_c and f_d in I_c and I_d respectively. A feature pair (f_q, f_c) is considered a good match if (a) $\Delta(f_q, f_c) < \Delta(f_q, f_d)$ where Δ is the distance function, and (b) f_c has not already been matched with another feature in the query image.

2.2 Combining Results from Multiple Cells

We now describe the "Cell Selection" block in Figure 1. The most straightforward way to retrieve the matching image to a query is to search over the cell whose center comes closest to the reported location the query. However, given the cell geometry constraint introduced in this section ensuring that each reported location is either in 3 or 4 cells, it is conceivable to improve the single cell matching performance by combining match results from multiple cells. In practice, since we are given coarse reported locations rather than actual locations, it is impossible to determine which 3 to 4 cells to search over. As such, we search over all cells that intersect with a query's ambiguity circle, and combine the scores for all resulting candidate matches; we refer to this as "cell combination". Assuming the cell layout structure satisfies the condition in (1) with $d = r$, the ambiguity circle for a given query image can be shown to intersect with at most 9 cells, placing an upper bound on the maximum number of local searches per query. We combine results from multiple cells by a simple summation of the scores from the queried cells as shown in the "Combine Results" block in Figure 1.

[2] Strictly speaking, this does not hold if the image location is at the boundaries of the cell grid.

2.3 Geometric Consistency and Re-ranking of Results

After combining scores across multiple cells, we apply an additional geometric consistency check to eliminate all feature matches that do not satisfy the epi-polar constraints. Furthermore, to account for differences in angles, we filter out feature matches where the angle of the SIFT features differ by more than 0.2 radians. Applying these two additional constraints to the feature matches from step 2.2 yields a re-ranked list of candidate image as shown in the geometric consistency and rotation re-ranking block in Figure 1. Since we are concerned with retrieving only a single matching image, we can reduce computational cost by computing only a partial re-ranking. In particular, let S be the sorted list of ranked database candidates from step 2.2 and S' be a sorted list of re-ranked candidates that we wish to generate. We loop through, from best to worst, the candidates in S and insert the re-ranked candidates in S'. Since the geometric consistency and SIFT angle checks only remove bad feature matches and do not introduce new matches, the score of any particular candidate after re-ranking can only decrease. As such, the first j elements in S' are guaranteed to be stable once we come across a candidate, s, whose score before re-ranking is less than the re-ranked score of the j^{th} candidate in S'; i.e. $s'_j > s$.

2.4 Bayesian Post Processing on the Top Results

As a final step, we apply a "distance filter" to refine our results based on the distance from the cell phone's reported location to the location of the candidate image. Using this distance δ and the score s' from Section 2.3 normalized by the number of features in the query, we train a Naive Bayes classifier to generate a match likelihood $p(m|\delta, s')$ for each candidate image, where $m = 1$ represents a ground truth match and $m = 0$ represents no match. A Naive Bayes classifier is chosen because of our small training set, the ease in which it is updated with new data, and because it provides us with a confidence indicator. Since the parameters δ and s' are continuous, we use bins to train the distributions $p(\delta|m)$ and $p(s'|m)$ needed for classification.

The candidates are re-ranked one final time using the match likelihood generated from δ and s'. Not only does this give us a refined ranking of the candidates, but it also provides us with a confidence indicator. Since the match likelihood represents the probability that a candidate is a match given its parameters δ and s', we can use the match likelihood of the top candidate as a measure for how well the system performs on the query, referred to in this paper as a confidence indicator.

3. Experimental Results

Our database is from Earthmine Inc., i.e. the same source used in [**3**], and consists of street level images collected over a ~1 km^2 area of downtown Berkeley. Using multiple viewpoints of a scene has been shown to increase the detection rate in retrieval tasks [**9**]. As such, we extract 6 images per location with 3 images from each side of the capture vehicle, yielding roughly 12000 images with approximately one fronto-parallel and two perspective views per building. Each 768×512 pixel image has a 60 degree field of view and 50% overlap with neighboring images.

For our dataset, we have found 25 meters to be a reasonable value for α, with 50 meters being the maximum distance between two locations capturing the same view. Assuming that the maximum discrepancy in reported location λ is 75 meters, we space our cells based on an ambiguity radius g of 100 meters with $r = d =$ 236.6 meters. As such our database is divided over 25 cells with each cell containing roughly 1500 images. Table 1 shows the various query sets we use to characterize the performance of our system. As seen, 561 query images, downsampled to approximately the same size as our database images, are taken using a digital camera and cell phone in fair weather with automatic camera settings. These images are geo-tagged with GPS location information. The Naive Bayes classifier uses the 65 query images in set 2 with a total of 5499 candidate database images to train the distributions $p(m)$, $p(\delta|m)$, and $p(s'|m)$ necessary for generating match likelihoods.

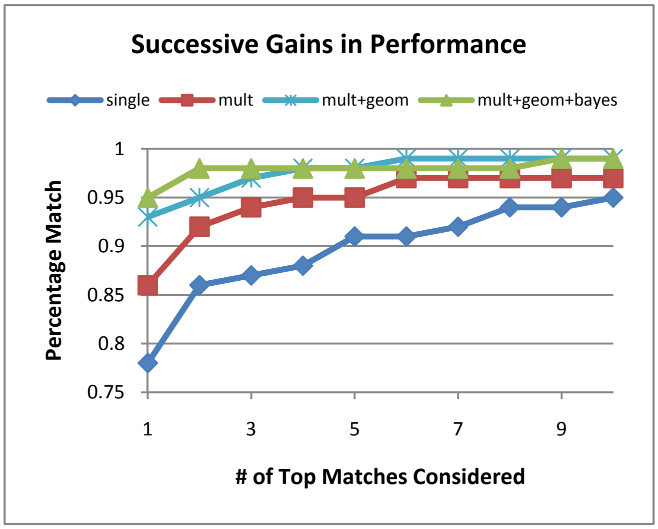

Fig 3: Successive performance gains due to each step in Section 2 compared against a single cell baseline.

Set #	Camera	Orientation	Zoom	Size	Comments
1	SLR	Landscape	Fixed	100	
2	SLR	Landscape	Fixed	65	Used for training
3	SLR	Landscape	Varying	84	
4	Smartphone	Portrait	Fixed	112	
5v	Smartphone	Landscape	Fixed	100	Same views as set1
5h	Smartphone	Portrait	Fixed	100	Same views as set1

Table 1: Query sets used to generate Figures 3 and 4.

Figure 3 shows the successive gains in performance due to each step in our retrieval pipeline for top 1-5 retrieved images using query set 1 shown in Table 1. As a baseline, we examine the results of querying against a single, rather than multiple, FLANN kd-tree cell without geometric verification. For top 1 retrieval the baseline single cell approach results in a 78% match rate, as compared to a 95% match rate from applying the steps described in Sections 2.2-2.4. In general, we find query images containing large amounts of street and sky features result in poor retrieval performance. This is most apparent in query sets 4 and 5v, which are taken in a portrait orientation and as such capture a great deal of street and sky detail. The performance of our retrieval pipeline across various datasets is shown in Figure 4. As seen, there is a visible decrease in performance on query sets with portrait orientations.

Even though the cell structure used to generate the results in Figures 3 and 4 has been designed to handle maximum error in reported location of up to $\lambda = 75$ meters, in practice, the reported GPS location obtained during query capture process was considerably more accurate. To simulate much noisier location readings, such as those obtained via cell tower triangulation, we uniformly sample with 1 meter resolution, all points up to $\lambda = 75$, meters from the acquired GPS location for each query image, yielding $\pi 75^2 \cong 17{,}000$ locations per query. We then feed the query images with these simulated locations into our retrieval pipeline. Our results, not shown here, indicate less than 1% change in performance under such simulated fuzziness in location ambiguity; this shows indicating that the reported location is not significantly important as long as it falls within $\lambda = 75$ meters of the actual location as supported by the system. To further characterize the robustness of our system, we have simulated its performance against location errors of up to 200 meters based on an exponential probability distribution approximating the 67% (95%) within 50 (150) meter location accuracy requirement mandated by the FCC for e911 purposes [**10**]. Results shown in Figure 4 for set 1, labeled as “set1 FCC”, indicate that the greater location ambiguity for “set1 FCC” leads to an 8% drop in top 1 retrieval performance as compared to “set1” which corresponds to the more accurate reported GPS location; top 2-10 retrieval performance roughly unchanged.

We have also found that the conditional probability $p(m|\delta, s')$ for the top result acts as a good confidence indicator for whether our system has found a correct match. For queries with a confidence indicator greater than 0.8, our image retrieval system generates a top result match 96% of the time, while queries with a confidence indicator less than 0.4 fail to generate a top result match 86% of the time. This correlation between our confidence indicator and image retrieval performance across all test sets is shown in Figure 5.

4. Conclusions and Future Work

In the paper, we have presented a method for large scale recognition and retrieval against large sets of geo-tagged images using coarse location information. Since our local search cells are relatively small, we have opted to use a feature-match-vote recognition scheme. However with more densely distributed image sets, or larger errors in reported versus actual location estimates, such a local search method might not scale, and more scalable retrieval structures might be needed. Future work involves exploring other feature descriptors and preprocessing methods as well as pose recovery once an image match is retrieved.

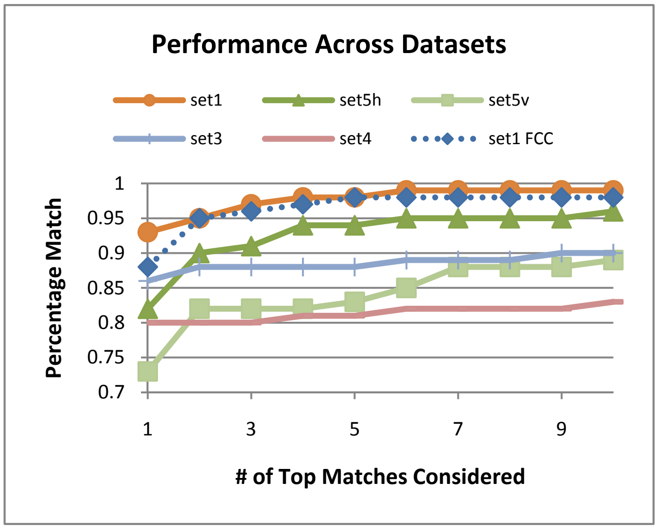

Fig 4: The performance of our image retrieval approach across various datasets.

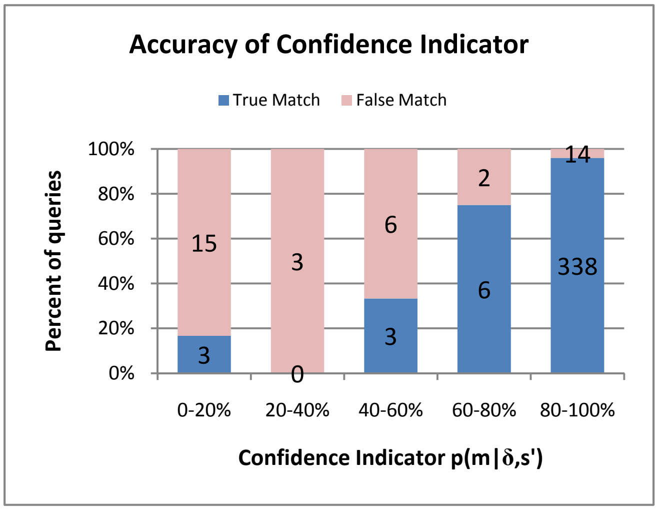

Fig 5: The performance of image retrieval based on the query's confidence indicator across all test sets.

5. References

[1] W. Zhang and J. Kosecka, "Image Based Localization in Urban Environments," in *3DPVT*, 2006.

[2] G. Schindler, M. Brown, and R. Szeliski, "City-Scale Location Recognition," in *CVPR*, 2007.

[3] G. Baatz, K. Koser, D. Chen, R. Grzeszczuk, and M. Pollefeys, "Handling Urban Location Recognition as a 2D Homothetic Problem," in *ECCV*, 2010.

[4] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, B. Girod, "Outdoors Augmented Reality on Mobile Phone using Loxel-Based Visual Feature Organization," in *MIR*, 2008.

[5] R. Paucher and M. Turk, "Location-based augmented reality on mobile phones," in *CVPR*, 2010.

[6] A. Mohamed, P. Welinder, M. Munich, and P. Perona, "Scaling Object Recognition: Benchmark of Current State of the Art Techniques.," in *ICCV*, 2009.

[7] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *IJCV*, vol. 2, no. 60, pp. 91-110, 2004.

[8] M. Muja and D. G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," in *VISAPP*, 2009.

[9] D. Chen, S. S. Tsai, V. Chandrasekhar, G. Takacs, J. Singh, B. Girod, "Robust Image Retrieval using Multiview Scalable Vocabulary Trees," in *VCIP*, 2009.

[10] Federal Communication Commission, "OET Bulletin No. 71 Guidelines for Testing and Verifying the Accuracy of Wireless E911 Location Systems," Federal Communication Commission, 2000.