LARGE SCALE IMAGE RETRIEVAL IN URBAN ENVIRONMENTS WITH PIXEL ACCURATE IMAGE TAGGING

by Jerry Zhang

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Avideh Zakhor
Research Advisor

(Date)

* * * * * * *

Professor Trevor Darrell
Second Reader

(Date)

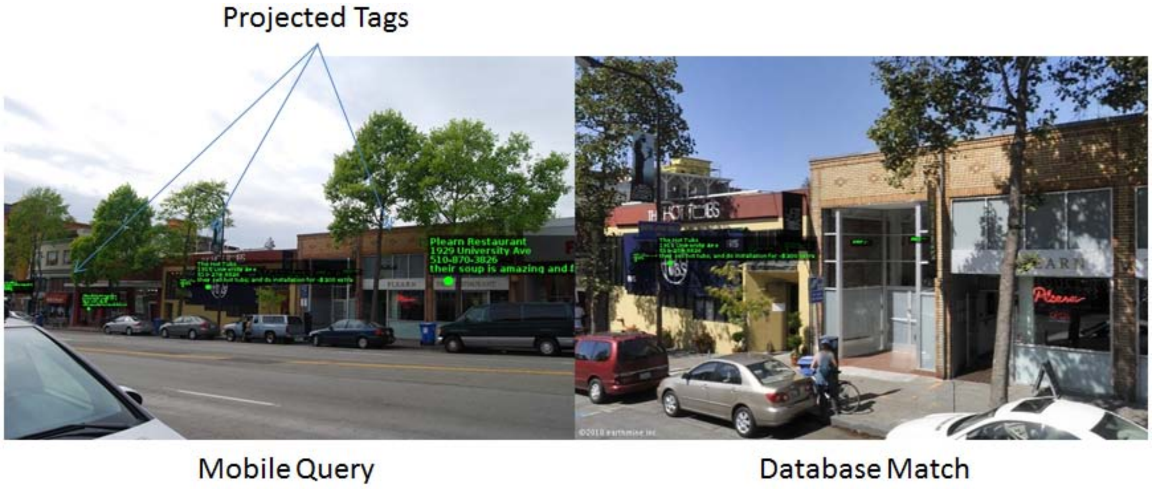


Figure 1: Address information and business reviews are projected from a tagged database image onto a user generated query using our system.

ABSTRACT

City-scale image retrieval and tagging is an important problem with many applications in localization and augmented reality. The basic idea is to match a user generated query image against a database of tagged images. Once a correct match is retrieved, pose information associated with the retrieved image can be used to augment the query image. In this report we describe an approach to large scale image retrieval in urban environment by taking advantage of coarse position estimates available on many mobile devices today, e.g. via GPS or cell tower triangulation. By partitioning the large image database for a given geographic region into a number of overlapping cells each with its own prebuilt search and retrieval structure, we avoid the performance degradation faced by many city-scale retrieval systems. Typically, both retrieval speed and retrieval accuracy decreases as the size of the database grows. Once a correct image match is found, a set of point to point correspondences between query and retrieved image is used to compute a homography transformation which can then be used to transfer tag information associated with points in the database image onto the query image with near pixel-level accuracy. An example of a tagged query outputted by our system and its corresponding database match is shown in Figure 1. We demonstrate retrieval results over a ~12,000 image database covering a 1 km^2 area of downtown Berkeley and illustrate tag transfer results over the same dataset.

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

CONTENTS

1. INTRODUCTION

In recent years, a number of large scale image databases for outdoor scenes such as Google StreetView and Bing StreetSide have been developed for general public use. In these applications, the user provides a textual description of a location in the form of a business name, address, or lat/lon coordinates in order to retrieve images associated with the location.

Given the widespread availability of such datasets, one can imagine performing the reverse image-to-text query operation. In particular, by matching a user captured image against a geo-tagged database of street level images, one can retrieve location specific meta-information, such as business ratings or address information for the buildings and structures in the vicinity. This can be used in mobile augmented reality applications, which for the most part do not currently utilize visual scene information[1].

Image retrieval systems where user generated query images are matched against an existing image database of a region, e.g. from Google's Street View, or Microsoft's StreetSide have traditionally been framed around solving the problem of localization and pose recovery. In these systems, once the best image match has been retrieved, the known pose of the database image is used to determine the pose of the query image supplied by the user [**1**]. A number of such systems have been proposed in recent years: [**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

[1] Systems that currently rely on visual information typically do so through the use of visual fiduciary markers; as such, they do not scale to large scale urban environments.

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 near real time over a preprocessed set of features from nearby regions for on-the-device queries. In practice however, GPS information is oftentimes not readily available, especially in urban environments where satellite reception is hindered by “urban canyons”, and typically subject to relatively large error. Therefore, to avoid retrieval performance degradation with image database size, we propose to take advantage of the approximate coarse localization available on most mobile devices[2], e.g. 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; as such, 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 and as such, it scales to arbitrarily large regions.

Using a coarse location approximation, we can direct the user generated query image to a small, bounded, subset of cells to be queried against. To determine the cells whose

[2] 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.

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 a matching image to the user query. Once the best database match for the query image is retrieved, the information associated with the database image can be projected onto the query image through the use of a homography matrix derived from the point-to-point correspondence between the two images.

An overview of our system is shown in Figure 2. In the remainder of this report, we will review various components of the system shown in Figure 2. The outline of the report is as follows: We describe our image retrieval and our tag transfer approach in Sections 2 and 3 respectively, go over our experimental setup and results in Section 4, and present our conclusions in Section 5.

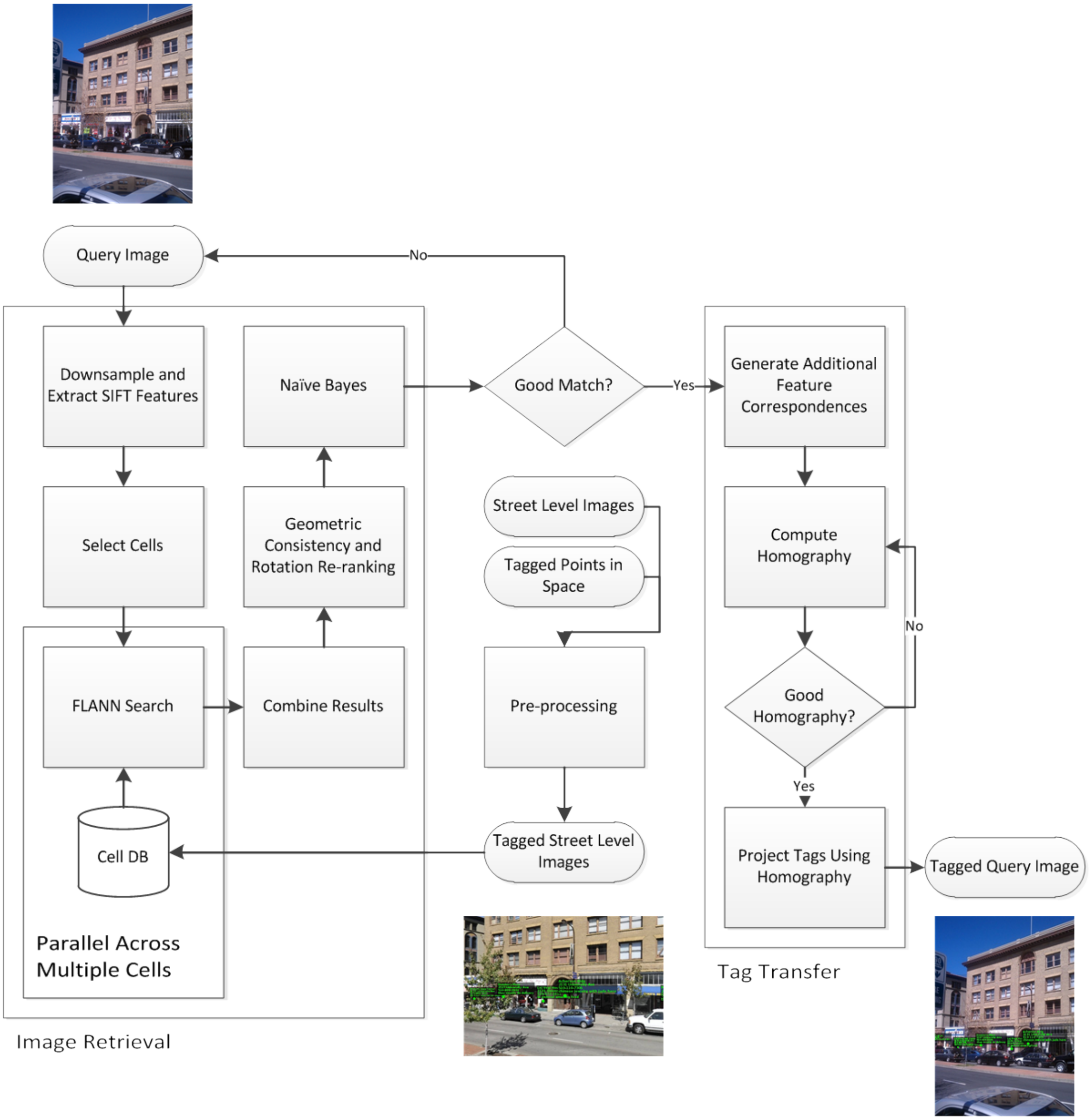
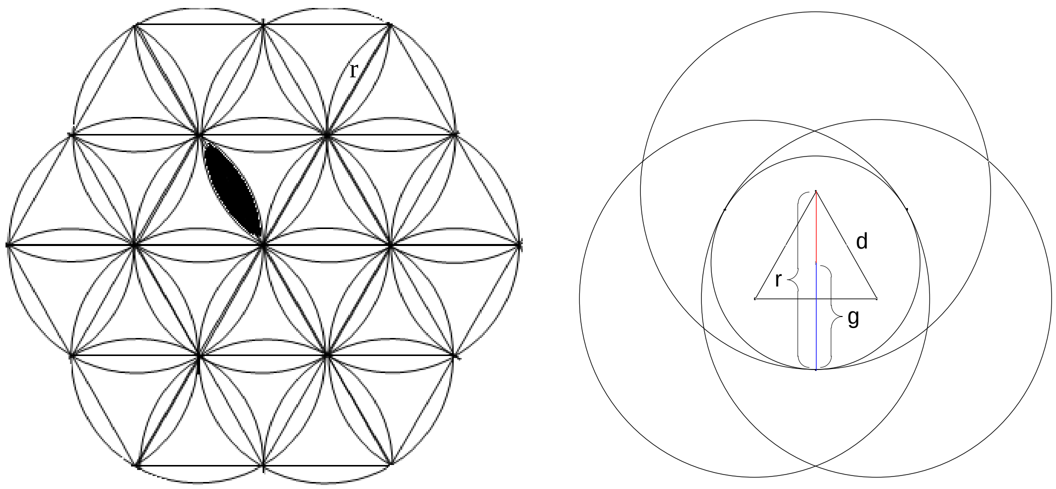


Figure 2: Outline of our retrieval/tagging pipeline.

2. Image Retrieval 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 each cell contains 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 3(a).



(a) (b)

Figure 3: (a) Local search cells built over a hexagonal lattice. (b) Ambiguity circle of radius $g = \lambda + \alpha$ fully contained in the intersection of 3 cells of radius r with separation d.

2.1 Cell Layout Geometry

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. The location of a database match can be off by a distance of up to $\lambda + \alpha$ from a query's reported location. As such, to guarantee that a given query image has at least one cell containing all true matches, any circular region of radius $g = \lambda + \alpha$, referred to as an "ambiguity circle", must be fully contained by at least one cell in the cell grid shown in Figure 3(a). 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 3(b) shows that this condition is satisfied if an ambiguity circle of radius g can be fully contained within the region of intersection of three adjacent cells; a proof of this is presented in Appendix A: SCE . 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.

The largest circular region, with radius g, which fits within the intersection of 3 overlapping cell is shown in Figure 3(b). In particular, we note that it is internally tangent to each cell and its center is 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 Equation (1) becomes:

$$d = r \geq \frac{\sqrt{3}}{\sqrt{3} - 1} g \tag{2}$$

This case is shown in Figure 3(a) and simple geometric inspection reveals that every database image is always contained in either 3 or 4 search cells[3]. Specifically, any database image whose location falls within the “petal” region of the layout scheme, one of which is highlighted in Figure 3(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 2.3 to combine results from multiple cells.

2.2 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 [**7**] features in the query image with their approximate nearest neighbor in the database images using a FLANN [**8**] kd-tree[4] of all features in the local cell. 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.

2.2.1 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

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

[4] We use a single, rather than multiple, kd-tree implementation since we parallelize the approximate nearest neighbor search across multiple cells as outlined in Section 2.3. For a single cell search strategy, a multiple kd-tree implementation is more appropriate.

new method for evaluating the 'goodness' of a feature pair by comparing the nearest neighbor distances of parallel kd-tree queries while enforcing a uniqueness constraint on the match pairs. Intuitively, the distance between a query feature and a correct match should be significantly better than the distance between the query feature and the closest incorrect match. Specifically, provided there are a sufficiently large number of local cells in our database, for any query location we identify a dummy cell c' with image set $I_{c'} = \{I_1, I_2 \dots, I_n\}$ that the query location is known not to reside in. The features in I_c and $I_{c'}$ have been put into an approximate nearest neighbor kd-tree offline. For each feature $f_q \in I_q$ we can, in parallel, compute its nearest neighbors f_c and $f_{c'}$ in I_c and $I_{c'}$ respectively. A feature pair (f_q, f_c) is considered a good match if (a) $\Delta(f_q, f_c) < \Delta(f_q, f_{c'})$ where Δ is the Euclidean distance, and (b) f_c has not already been matched with another feature in the query image.

2.3 Combining Results from Multiple Cells

We now describe the "Select Cells" block in Figure 2. 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 in Equation (2) ensuring that each 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 Equation (2) with $d = r$, the ambiguity circle for a given query image can intersect with at most 9 cells as shown in Figure 4, 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 2. Since each database image is contained in a variable number of cells, it is conceivable that the combined score will be confounded by whether the database match is contained in 3 or 4 cells. This situation can be rectified by either adopting a hexagonal, rather than a circular, cell geometry or by weighing the combined score based on the number of cells the database match is contained in. We have empirically found that in practice, this confound does not significantly affect retrieval performance.

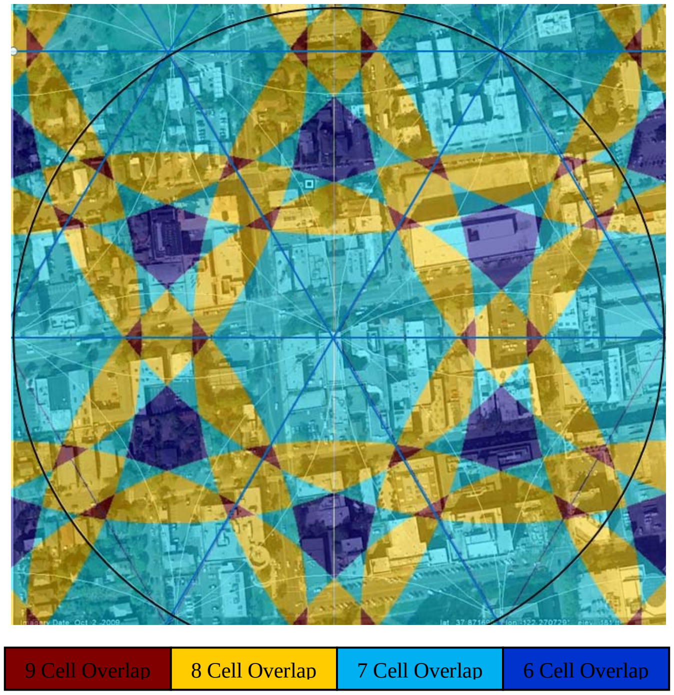


Figure 4: The number of cells intersected by ambiguity circles, as defined by Equation 2, centered at various regions in the hexagonal lattice grid. The regions colored in red, yellow, light blue, and blue corresponds to cases where the ambiguity circle intersects 9, 8, 7, and 6 cells respectively. The boundary of a cell is outlined in black. The vertices of neighboring cells form equilateral triangles as indicated by the blue lines.

2.4 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 epipolar constraints as shown in Figure 5.

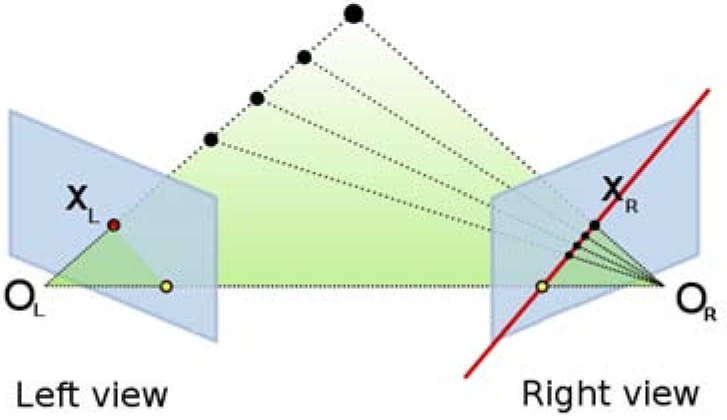


Figure 5: The point X_L observed in the left image must be observed in the right image along an epipolar line shown in red. O_L and O_L denote the focal points of the left and right image respectively.

Specifically, we use a RANSAC [**9**] loop to compute the fundamental matrix and discard all outlier feature matches [**10**][5]. 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 Section 2.3 yields a re-ranked list of candidate image as shown in the "Geometric Consistency and Rotation Re-ranking" block in Figure 2.

[5] For convenience, we provide an outline of Hartley and Zizzerman's algorithm in Appendix B: RANSAC Homography and Fundamental Matrix calculations.

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 Section 2.3 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. $s'_j > s$. In the event that all features are removed as a result of this filtering, we pass S directly to our Bayesian Post Processing outlined below.

2.5 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 user's reported location to the location of the candidate database image. Using this distance δ and the score s' from Section 2.4, we re-rank our results by using a Naive Bayes classifier to generate a match likelihood for each candidate image.

3. TAG TRANSFER APPROACH

Assuming a good image match is retrieved[6], we must transfer the tag information from the matched database image to the query image. As our system is designed for use in urban environments, we assume that both the query and database image contain a

[6] This is a reasonable assumption as we can simply ask the user to generate a new query, possibly from a different vantage point, if the match confidence from Section 2.5 is below a set threshold.

dominant plane – typically this will be in the form of a building facade. Under this assumption, we can derive a homography transformation which is then used to project the tag information associated with the matched database image onto the query image.

3.1 Pre-processing of Database Tags

For our particular application, our tags are stored as text data associated with rays in 3D space, i.e. lat, long, alt, and yaw. To integrate these tags into retrieval system, we first associate every tag with its nearby database images. In particular, because we have the location and pose of every database images, we can project the nearby tags of every database image onto the database image plane. We use points on the image plane rather than the database image itself because a query image might only partially overlap a database image in terms of content. To accurately tag a query image, we must therefore consider not only the tags within a database image, but also tags lying outside the field-of-view of the image. This process is illustrated in Figure 6.

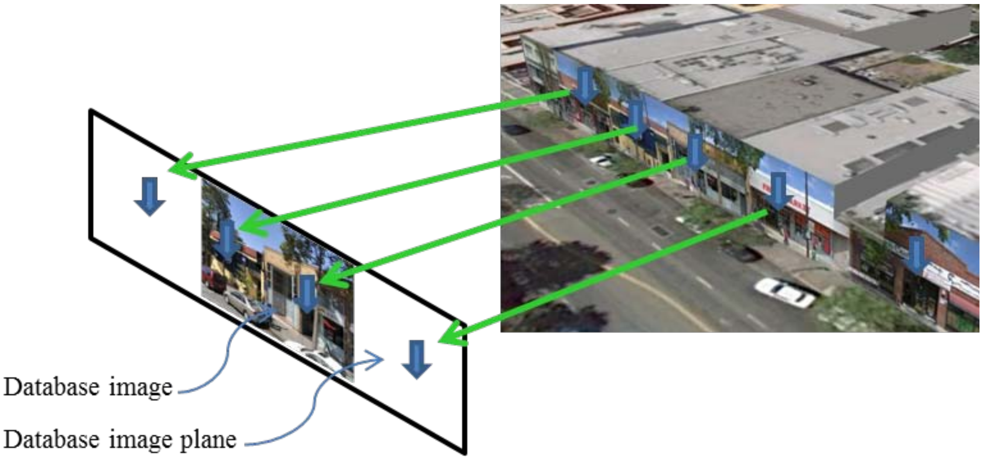


Figure 6: To account for differences in viewpoint between the database and query image, tags are associated with points in the database image plane.

3.2 Generating Additional Feature Matches

Since the kd-tree used in Section 2.2 pairs features in the query against features in the entire database cell, the final set of matched features following the combination and re-ranking steps outlined in Sections 2.3-2.5 might yield too few match pairs to reliably compute a homography between the query and retrieved image. By re-matching features directly between the query and top database image, we can increase the number of good feature match pairs. To re-match features, we prebuild kd-tree indexes for each database image. Experimentally, we find that combining the recomputed feature match pairs with the previous feature matches from the image retrieval pipeline significantly increases the accuracy of the computed homography matrix.

3.3 Computing and Rejecting Homographies

Similar to the fundamental matrix calculations performed in Section 2.4, we use RANSAC to obtain a robust estimation of the homography matrix between the query and database image [**11**][7]. Though the top 1 match from the retrieval pipeline has the highest a priori probability of being a correct match, the quality of the computed homography between the matches also is a strong indication of match success. We quantify the "goodness" of a homography matrix $\boldsymbol{H}$ by computing its determinant. In particular, very large or small values of $\det(\boldsymbol{H})$ are indicative of a degenerate homography [**12**], and negative values of $\det(\boldsymbol{H})$ indicate a reflective component – a physical impossibility in our application. In addition, we perform checks for other known physical impossibilities such as strong skew and rotation. If we cannot find a valid homography under these

[7] For convenience, we provide an outline of Hartley and Zizzerman’s algorithm in Appendix B: RANSAC Homography and Fundamental Matrix calculations.

criteria, we repeat the process on the next highest ranked match until an acceptable homography is found.

3.4 Homography Tag Transfer

Assuming an acceptable homography matrix is found, we apply the homography to all tags associated with the database image and superimpose the textual tags onto the query image. Since the tags are already represented as 2D points on the database image plane from Section 3.1, this is a simple matrix multiplication:

$$p_{query} = \boldsymbol{H} p_{match}$$

Where p denotes the 2D position of the tag on the image plane, and H is the computed homography matrix.

4. 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 [**13**]. 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. This process is illustrated in Figure 7.

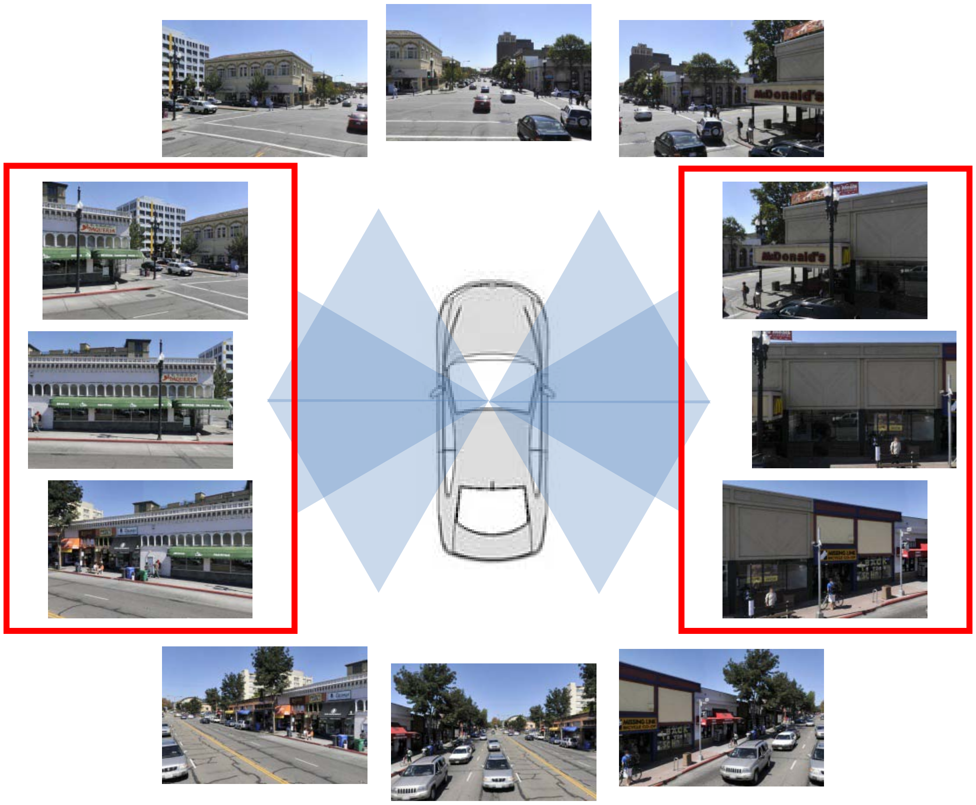


Figure 7: 6 images are extracted from each vehicle capture location. Each image has a 60° field of view and are oriented at 60°, 90°, 120°, 240°, 270°, and 300° angles with respect to the front of the capture vehicle.

For our Berkeley dataset, we have found, through examination, 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 in order to satisfy Equation (2). As such, our database is divided into 25 cells with each cell containing roughly 1500 images.

Set #	Camera	Orientation	Focal Length	Size	Count	Comments
1	SLR	Landscape	Fixed	765x512	100	Same views as set5
2	SLR	Landscape	Fixed	765x512	65	Used for training
3	SLR	Landscape	Varying	765x512	84	
4	Smartphone	Portrait	Fixed	504x840	112	
5v	Smartphone	Landscape	Fixed	512x680	100	Same views as set1
5h	Smartphone	Portrait	Fixed	680x512	100	Same views as set1

Table 1: Query sets used to generate Figure 5 and Figure 6.

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 tagged with their reported GPS location obtained either through the device's built-in GPS unit or via an external GPS receiver. The Naive Bayes classifier uses the 65 query images in set 2 with a total of 5499 candidate database images for training.

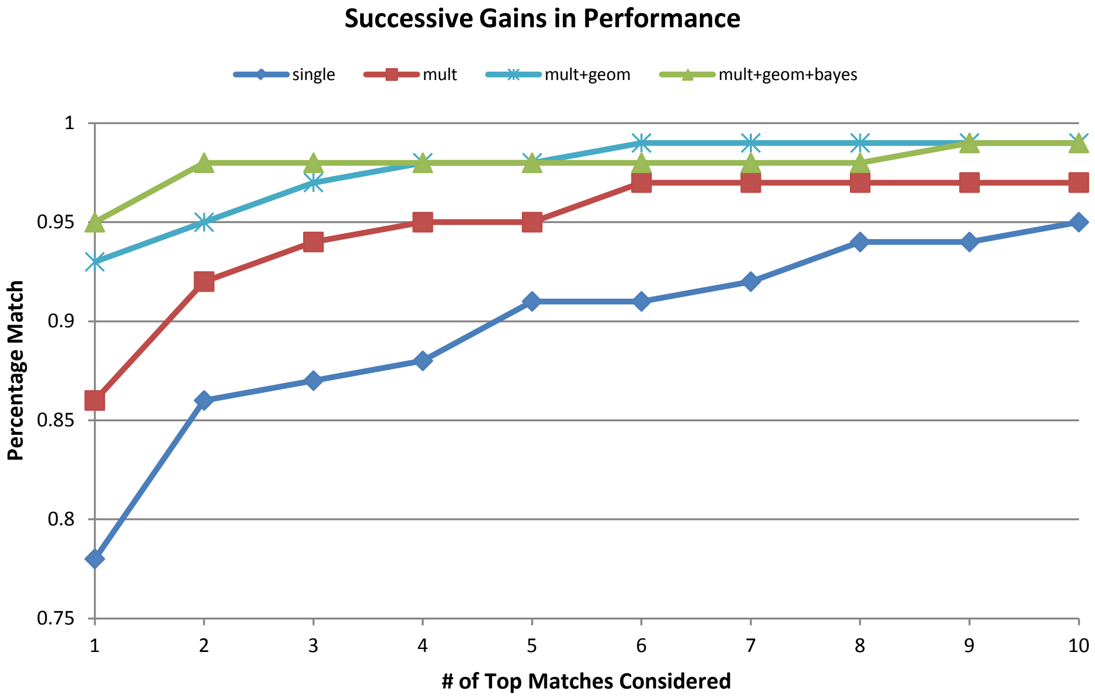


Figure 8: Successive performance gains due to the steps outlined in Sections 2.3-2.5 compared against a single cell baseline.

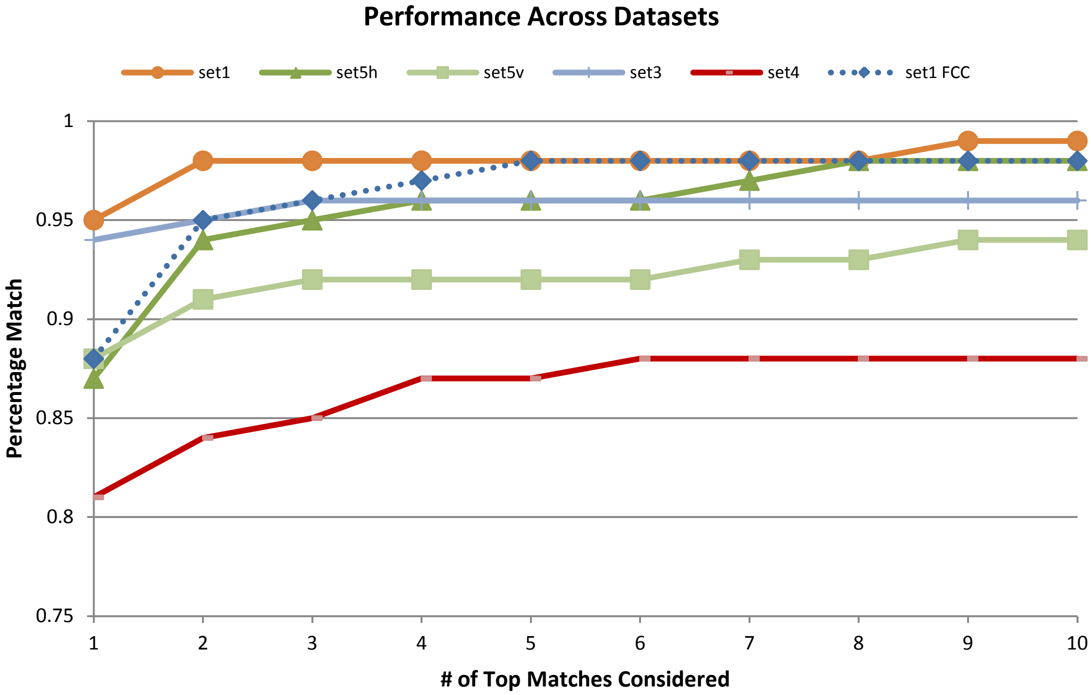


Figure 9: Performance of our image retrieval approach across various datasets.

4.1 Retrieval Performance

Figure 8 shows the incremental gains in performance due to the steps outlined in Sections 2.3-2.5 of our retrieval pipeline for top 1-10 retrieved images using query set 1 of Table 1. As a baseline, we examine the results of querying against a single FLANN kd-tree cell without geometric verification or Bayesian post-processing. 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 2.3-2.5. Specifically, the multi-cell combination step outlined in Section 2.3 yielded a 8% improvement from the baseline, adding the geometric consistency checks outlined in Section 2.4 led to an additional 7% improvement, and the Bayesian post processing step outlined in Section 2.5 led to a final 2% improvement in performance. In general, we find that 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 9. As seen, there is a visible decrease in performance on query sets with portrait orientations. Furthermore, query images acquired from digital SLRs tend to perform better than those acquired from smartphones.

Even though the cell structure used in our experiment 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 in urban settings, 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 and characterize its performance. Our results, not shown here, indicate less than 1% change in performance under such simulated location ambiguity; this shows that the reported location is not significant 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 [**14**]. Results shown in Figure 9 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 5-10 retrieval performance remained roughly unchanged.

We have found that the conditional probability, resulting from the Bayesian Post-Processing set of Section 2.5, for the top match 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 10.

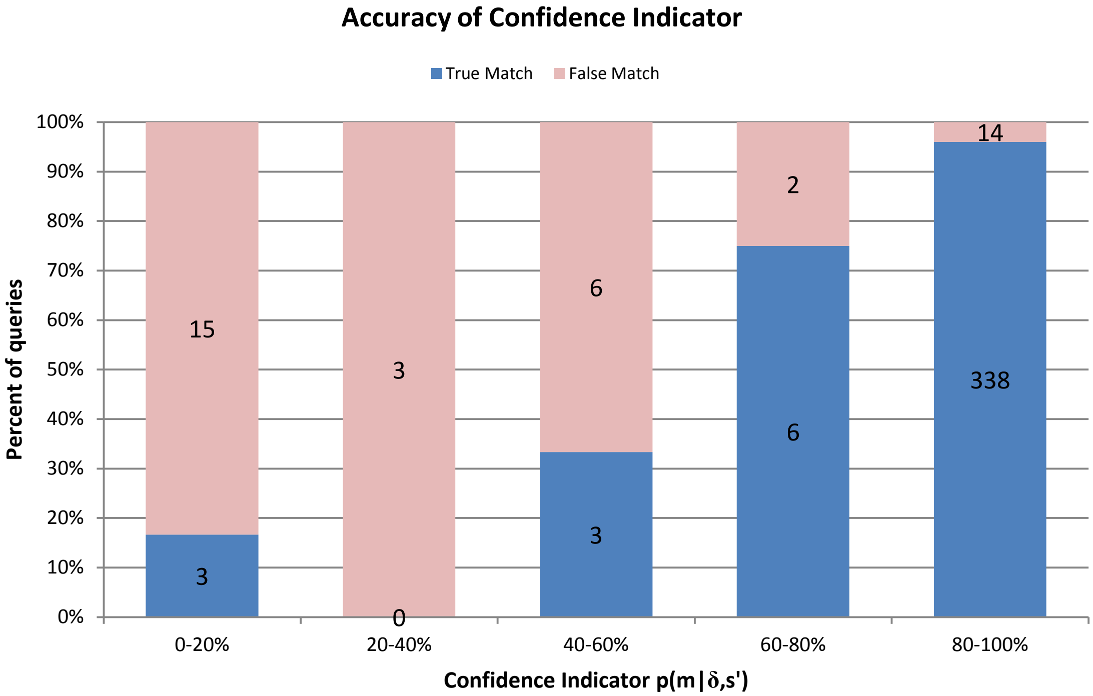


Figure 10: Performance of the image retrieval based on the query's confidence indicator across all test sets.

4.2 Tag Transfer Performance

To characterize the performance of our tag transfer approach, we examine the set of retrieved images for which a valid homography is detected per Section 3.3 and at least one tag is available. We evaluate these tagged images based on the pixel location accuracy of the transferred tags as compared to the database image. A query image whose tags haves been transferred correctly from its database match is considered a correctly tagged image. Results are shown in Table 2.

Set #	# of Tagged Images	# of Correctly Tagged Images	% Correctly Tagged
1	63	54	86%
3	52	44	85%
4	60	51	85%
5v	55	49	89%
5h	53	44	83%
Total	283	242	86%

Table 2: Tag transfer performance on correctly retrieved images across datasets.

Of the images that were tagged, the homography tag transfer method was able to accurately transfer text information onto a query image from a matched database image for 242 out of 283 images. Of the cases where the tag was not projected correctly onto the query image, several were situations where the query image captured an opposing street corner and therefore contained more than one dominant plane in the image. A subset of the images retrieved and tagged by our system is shown in Figure 11.

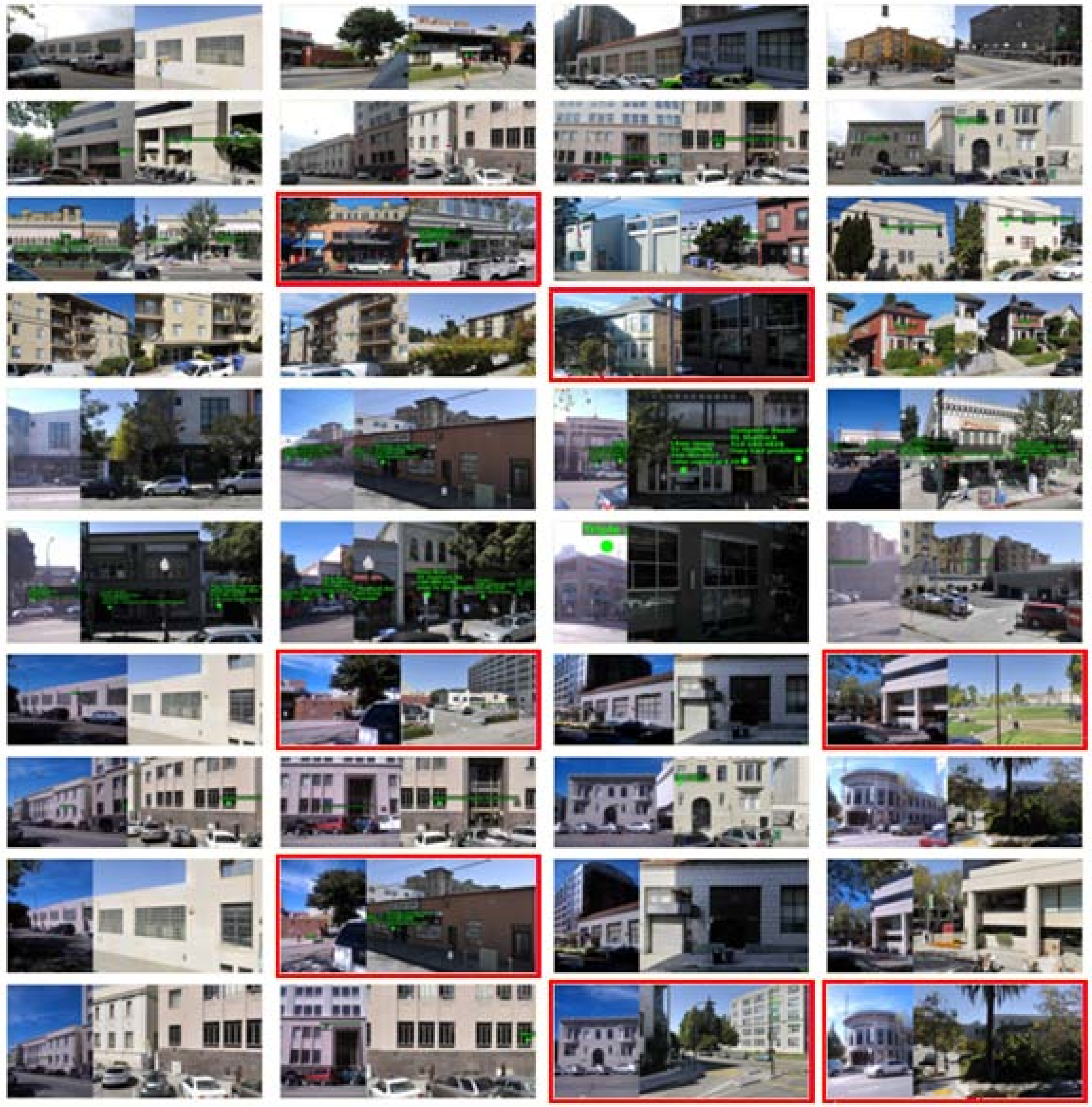


Figure 11: Select query and match pairs with projected tags. For each image pair, the left image is the query and the right image is the corresponding retrieved image from the database. Incorrect image matches are highlighted in red.

5. Conclusions and Future Work

In the report, we have presented a method for large scale retrieval against large sets of geo-tagged images using coarse location information. Using the retrieved image, a homography matrix can be used to project tag information onto the user generated query image in a pixel accurate fashion. 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. Similarly, because images of urban buildings typically contain a single dominant plane, a homography model was an appropriate fit. In situations where there are multiple planes in the image, such a model would not work. Future work involves exploring other feature descriptors and preprocessing methods as well as recovering the user's 6 degrees of freedom pose information.

6. ACKNOWLEDGEMENTS

I would like to acknowledge the Air Force Office of Scientific Research and the Army Research Office for funding this project. I would like to acknowledge my advisor, Professor Avideh Zakhor, for supervising this project. I would like to credit Aaron Hallquist for developing the Naïve Bayes approach mentioned in Section 2.5 as well as Eric Liang for his contribution to the tag transfer process discussed in Section 3. I also would like to thank Aaron Hallquist, Eric Liang, and Eric Tseng for their assistance in data collection and performance evaluation.

7. REFERENCES

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

[2] Grant Schindler, Matthew Brown, and Richard Szeliski, "City-Scale Location Recognition," in *CVPR*, 2007.

[3] Georges Baatz, Kevin Koser, David Chen, Radek Grzeszczuk, and Marc Pollefeys, "Handling Urban Location Recognition as a 2D Homothetic Problem," in *ECCV*, 2010.

[4] Gabriel Takacs et al., "Outdoors Augmented Reality on Mobile Phone using Loxel-Based Visual Feature Organization," in *MIR*, 2008.

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

[6] Aly Mohamed, Peter Welinder, Mario Munich, and Pietro Perona, "Scaling Object Recognition: Benchmark of Current State of the Art Techniques," in *ICCV*, 2009.

[7] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 2, no. 60, pp. 91-110, 2004.

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

[9] Martin A. Fischler and C. Robert Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381-395, June 1981.

[10] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*. West Nyack, NY, USA: Cambridge University Press, 2004, ch. 11.

[11] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*. West Nyack, NY, USA: Cambridge University Press, 2004, ch. 4, p. 123.

[12] Etienne Vincent and Robert Laganière, "Detecting Planar Homographies in an Image Pair," in *ISPA*, 2001.

[13] David Chen et al., "Robust Image Retrieval using Multiview Scalable Vocabulary Trees," in *VCIP*, 2009.

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

8. Appendix

8.1 Appendix A: SCE Equivalence

In this appendix we show that every "ambiguity circle" of radius less than g is fully contained by at least one cell in the cell grid of Figure 3(a) if and only if a circle of radius g can be fully contained within the region of intersection of three adjacent cells.

Letting C represent the set of all cells, A represent the set of all circular ambiguity regions, and $\Delta(a, c)$ denote the Euclidean distance between the centers of cell c and ambiguity circle a, the SCE condition described in Section 2.1 can be stated as such:

$$\forall\ a \in A\colon a_{radius} \leq g\ \exists\ c \in C\ \Delta(a, c) \leq r - g$$

We will show that for any given value of g in our cell arrangement, SCE is satisfied if and only if a circle of radius g can be fully contained within the region of intersection of three adjacent cells whose centers form an equilateral triangle of side length d. For convenience, we will refer to this second condition as Three Cell Containment, or TCC:

$$\exists a \in A\colon a_{radius} = g\ \ \exists c_1, c_2, c_3 \in C\colon c_1 \neq c_2 \neq c_{3,}\ \wedge_{\mathrm{i=1,2,3}} \Delta(c_i, a) \leq r - g, \wedge_{\mathrm{i,j=1,2,3\ i\neq j}} \Delta\left(c_i, c_j\right) = d$$ [8]

Thus, our goal is to show that SCE implies TCC and vice versa, i.e. $SCE \leftrightarrow TCC$.

We show $SCE \rightarrow TCC$ by contradiction:

[8] We use the notation $\wedge_{\mathrm{i=[X]}}\ Y$ to denote a preposition that is true if and only if Y is true for every case in x, $\vee_{\mathrm{i=[x]}}\ Y$ to denote a preposition that is false if and only if Y is false for every case in x, and $\neg Y$ to denote the preposition that is true if and only if Y is false.

1) Assume SCE holds for some values of g, r.

   a) This means that every ambiguity circle a with radius g is fully contained by at least one cell.

2) Assume$\neg TCC$:

$$\forall a \in A: a_{radius} = g \ \forall c_1, c_2, c_3 \in C: c_1 \neq c_2 \neq c_3 \ \vee_{\text{i=1,2,3}} \neg\Delta(c_i, a) \leq r - g \ \wedge_{\text{i,j=1,2,3 i}\neq\text{j}} \Delta\left(c_i, c_j\right) = d$$

3) Let a' be a circle of radius g centered at the centroid of an equilateral triangle in the lattice, such as in Figure 3(b). From (1) we know that there is at least 1 cell that fully contain a'.

4) Because a' is placed at the center of the equilateral triangle whose vertices are also cell centers, from symmetry we know that if one cell contains a', the two other cells whose center lie on the vertex of the equilateral triangle must also fully contain a'.

5) This contradicts (2) which specify that there does not exist any ambiguity circle of radius g that can be fully contained by three cells.

6) By contradiction, $SCE \rightarrow TCC$.

Let SCE_{eq} be the SCE condition with a strict equality on the ambiguity radius:

$$\forall \ a \in A: a_{radius} = g \ \exists \ c \in C \ \Delta(a, c) \leq r - g$$

Our approach to showing $TCC \rightarrow SCE$ is to break it up into two parts: First, we show that $SCE_{eq} \rightarrow SCE$ and then show that $TCC \rightarrow SCE_{eq}$. By combining these two results, we can then conclude that $TCC \rightarrow SCE$.

Let us now show that SCE_{eq} implies SCE. We note if every ambiguity circle of radius g can be contained by at least one cell, then it must hold that every ambiguity circle of radius $g' < g$ can also be contained by at least one cell. Therefore, $SCE_{eq} \rightarrow SCE$.

We show $TCC \rightarrow SCE$ by means of showing $TCC \rightarrow SCE_{eq}$ by contradiction:

1) Assume TCC holds for some values of g, r.
2) For some ambiguity circle a_1 of radius g, let c_1, c_2, c_3 denote the three distinct cells with $\wedge_{\mathrm{i=1,2,3}} \Delta(c_i, a_1) \leq r - g$ as implied by (1).
3) We note that due to the symmetry of our cell layout grid, it is possible to restrict our analysis to the equilateral triangle formed by the centers of these three cells.
4) Assume $\neg SCE_{eq}$:

$$\exists a \in A \colon a_{radius} = g \forall c \in C, \neg \Delta(a, c) < r - g$$

5) This means that there exists some ambiguity circle a_2 with radius g that is not fully contained by any cell.
6) Since both a_1 and a_2 have radius g, it must be the case that either $a_1 = a_2$ or $a_1 \neq a_2$.
7) If $a_1 = a_2$, then we know from (1) that a_2 is contained in at least 3 cells, which contradicts with (4).

8) If $a_1 \neq a_2$, then a_2 must lie some distance away from $a_1$in the equilateral triangle defined in (3).

9) Since the vertices of this equilateral triangle correspond to the centers of cells c_1, c_2, c_3, it must be the case that a_2 is *closer* to one of c_1, c_2, c_3: $\exists c_a \in \{c_1, c_2, c_3\}: \Delta(c_a, a_2) < \Delta(c_a, a_1)$.

10) Since $\wedge_{\mathrm{i=1,2,3}} \Delta(c_i, a_1) \leq r - g$, it must be the case that $\Delta(c_a, a_2) < r - g$, which contradicts (4).

11) By contradiction, $TCC \rightarrow SCE_{eq}$.

Thus we have shown that $TCC \rightarrow SCE_{eq}$. Since we have earlier shown that $SCE_{eq} \rightarrow SCE$, we can combine these two results to conclude that $TCC \rightarrow SCE$.

8.2 Appendix B: RANSAC Homography and Fundamental Matrix calculations

1) **Interest points**: Compute interest points in each image.
2) **Putative correspondences**: Compute a set of interest point matches based on proximity and similarity of their intensity neighborhood.
3) **RANSAC robust estimation**: Repeat for N samples, where N is determined adaptively.
   a. Select a random sample of 4 (7 for Fundamental Matrix) correspondences to generate a candidate Homography H (Fundamental Matrix F).
   b. Calculate the distance $d_\perp$for each putative correspondence.
   c. Compute the number of inliers consistent with H (F) by the number of correspondences for which $d_\perp$ is less than a set threshold of pixels.

   Choose the H (F) with the largest number of inliers. In the case of ties choose the solution that has the lowest standard deviation of inliers.

4) **Estimation:** re-estimate H (F) from all correspondences classified as inliers, by minimizing a cost function using the Levenberg-Marquardt algorithm.
5) **Guided matching:** further interest point correspondences are now determined using the estimated H (F) to define a search region.

Source: Multiple View Geometry in Computer Vision *by Hartley and Zisserman*