Large-Scale Geolocalization of Overhead Imagery

Mehul Divecha and Shawn Newsam
Electrical Engineering & Computer Science
University of California at Merced
mdivecha,snewsam@ucmerced.edu

ABSTRACT

In this paper, we investigate state-of-the-art computer vision techniques to perform large scale geolocalization of overhead imagery through image matching. We consider two types of features: scale invariant feature transform and region-based shape features. Since these features can be high dimensional and an image can contain many of them, using them to perform image matching can be computationally expensive. Therefore, we also investigate two methods for performing efficient matching: aggregating the features at the image level using a bag of words framework and using hashing to perform multiple, efficient matches and then aggregating the results. We show that hashing performs better in terms of accuracy but is expensive computationally compared to bag of words. We also show that shape features may be accurate and efficient for small data sets, but they do not scale well to large data sets.

CCS Concepts

•Information systems → Geographic information systems; •Computing methodologies → *Computer vision;*

Keywords

Overhead Imagery; Geolocalization; Image Matching

1. INTRODUCTION

We study the problem of determining the geographic location of high-resolution overhead imagery on a large scale. Given a query image that could be anywhere within a large geographic region, our goal is to assign at least approximate location. This could then be followed by small-scale geolocalization, through image registration for example, to precisely assign geographic coordinates to the content of the query image.

We formulate this as an image matching problem in which the query image is matched to a reference set. Our motivation for this is twofold. First, high-resolution imagery now exists for much of the Earth's surface. The location of this imagery is known and it thus serves as a reference for our problem. Second, a matching approach allows us to exploit recent advances in computer vision in image feature representation and aggregation that have been developed for related tasks such as image retrieval.

Overhead image geolocalization has received surprisingly little attention. This is probably due to the fact that, to now, overhead imagery, such as from air- or space-borne platforms, has mostly been acquired and archived in a systematic fashion. Traditionally, the location of the image is determined at capture time through the use of a geographical positioning system (GPS) or similar technology, and remains associated with the image as meta-data through data management best practices. We believe, however, that this is changing. Large amounts of overhead imagery is now being captured by camera enabled drones and the like. We argue that *location information is much less likely to exist for this imagery* for a variety of reasons. The acquisition and archival of this imagery is generally done in a much more ad hoc way. The drone might not be location aware or it might not receive reliable GPS signals. Even if the location is known at capture time, it might be lost or considered irrelevant when the imagery is archived or distributed on the web, for example. Or, the person who captured the imagery might want to deliberately obscure its location. We anticipate an explosion of overhead imagery with unknown location in the coming years and feel that our focus on the problem of geolocalizing this imagery is very timely.

While image matching is a conceptually simple and appealing approach to the problem, there are many technical challenges particularly when the geolocalization is done on a large scale. First is image representation. Representations, commonly known as features in computer vision, are needed that capture the salient aspects of the query and reference imagery while remaining invariant to noise, capture conditions, such as illumination, and geometric transformations, such as rotation, etc. We investigate two types of features, 1) keypoint-based scale invariant feature transform (SIFT) features [10] and 2) region-based shape moment features. Both of these are rotation and scale invariant. A high-resolution image will typically contain many of these features so the second challenge is how to aggregate them when performing the matching. We investigate two methods, 1) bag-of-words (BOW) which aggregate the (quantized) features at the image level so that image-to-image matching can be performed through query-to-reference image similarity, and 2) hash-based indexing which performs separate, efficient searches

SIGSPATIAL'16, October 31-November 03, 2016, Burlingame, CA, USA

DOI: http://dx.doi.org/10.1145/2996913.2996980

for each feature in the query image, and then aggregates the results through voting. We explore the four combinations of features–SIFT and shape moment–and aggregation methods–BOVW and hash-based indexing.

We perform experimental evaluation using a large collection of high-resolution overhead aerial imagery covering over 4,000 square kilometers. We explore the tradeoffs between accuracy and efficiency for the four approaches with increasing reference set size. We also explore the effects of varying query image size.

Our contributions are as follows:

- Overhead image geolocalization will soon become an important problem due to the popularity of camera-equipped drones. Our focus on large-scale geolocalization is novel and timely. We expect our paper to stimulate significant interest in the problem.
- We propose a conceptually simple solution based on image matching which can scale globally.
- We investigate four different combinations of image representations and feature aggregation techniques.
- We investigate the tradeoffs between accuracy and efficiency using a large experimental dataset.

2. RELATED WORK

The problem of *refining* the location of an overhead image is a well-studied problem. In this situation, the approximate location of the image is known and so the "search" only occurs over a small-scale region. Examples of this work include image registration, in which two images are brought into accurate alignment (e.g., [12]), and visual simultaneous location and mapping (SLAM) (e.g., [1]) from the robotics community. We focus instead on large-scale geolocalization.

A number of works have investigated *cross-modal matching* for geolocalization. Senlet et al. [15] perform building detection in overhead imagery in order to match against known building footprints in a geographic information system (GIS), and Costea and Leordeanu [3] perform road and intersection detection in order to match against known street networks in OpenStreetMap. While appealing, these approaches suffer from several drawbacks. First, they require high-level analysis of the query imagery which is itself a very difficult problem. Second, they require semantic level reference data in the form of a symbolic map. Such reference data is not available for all regions or, if it is, it might not be sufficiently consistent. In contrast, our image-to-image matching does not require a high-level understanding of the query image and uses widely-available consistent overhead imagery as the reference set.

Our image matching problem is similar to and shares many technical challenges with *content-based geographic image retrieval* (CBGIR) (E.g., [19, 13]). Our goal is different though. CBGIR seeks to retrieve the most similar images to a query image. While images are compared based on visual similarity, the goal is typically to retrieve semantically similar images. This is reflected in the evaluation protocols which use a set of labeled images, such as the popular UC Merced 21 class land use dataset [18], to determine the performance. A retrieved image is considered similar to the query if it is from the same semantic class. (The problem is thus similar to a classification task.) We instead seek the unique reference image region that matches our query. Semantically similar images do not solve our problem (although they might be useful to constrain it).

A fairly recent problem with similar motivation to ours is the geolocalization of *ground-level imagery*. This is the focus of the Intelligence Advanced Research Projects Activity (IARPA) Finder[1] program. Researchers have also framed this as an image matching problem in which the ground-level query image is either matched to ground-level images with known location (e.g., [8]) or matched to georeferenced overhead imagery (e.g., [16]). The technical challenges of this matching are mostly related to the significant variation in viewpoint and thus are not applicable to our problem. Also, the ground-level images do not represent a continuous region like our reference set of overhead imagery and can be treated independently, simplifying the problem. Nonetheless, we see the rise of the problem of ground-level image geolocalization following the increased acquisition of ground-level images due to smart phones, even to the point of getting the attention of a federal funding agency, as predicting the rise of overhead image geolocalization due to the increased acquisition of overhead imagery from drones.

Perhaps the most similar work to ours is the 2008 paper by Wu. et al [17] which performs overhead image geolocalization through image matching. They only consider SIFT features aggregated using BOVW, though, so our work expands and, based on our results, significantly improves on their approach. Our reference search area is also significantly larger: over 4,000 square kilometers versus 192 (both reference image sets have a resolution of one meter per pixel). Also, they only perform 60 manually chosen queries while we perform thousands of randomly selected ones. Interestingly, this seems to have been an isolated investigation into this problem by the authors–they do not appear to have worked on the problem since. We anticipate our work to be the start of a sustained effort.

In summary, while the problem of large-scale overhead image geolocalization is conceptually simple and appealing, it has received no attention with the exception of one nearly decade-old paper. Again, we feel this is in large part because of the nascence of this problem due to the recent availability drones. We feel our focus on large-scale geolocalization via image matching is novel and timely.

3. APPROACH OVERVIEW

Our goal is to use image matching to determine the location of an overhead image. The query image is matched to a reference set of images with known location. The matching is performed using image content as captured through extracted features. Different methods are quantitatively compared by randomly selecting image patches from a large set of overhead images and seeing whether they are correctly matched to their true locations. Both accuracy and efficiency are considered in the evaluation.

4. METHODOLOGY

We compare two different types of image features: SIFT and region-based shape. We also compare two feature aggregation methods: bag-of-words (BOW) and multi-index

[1]https://www.iarpa.gov/index.php/research-programs/finder

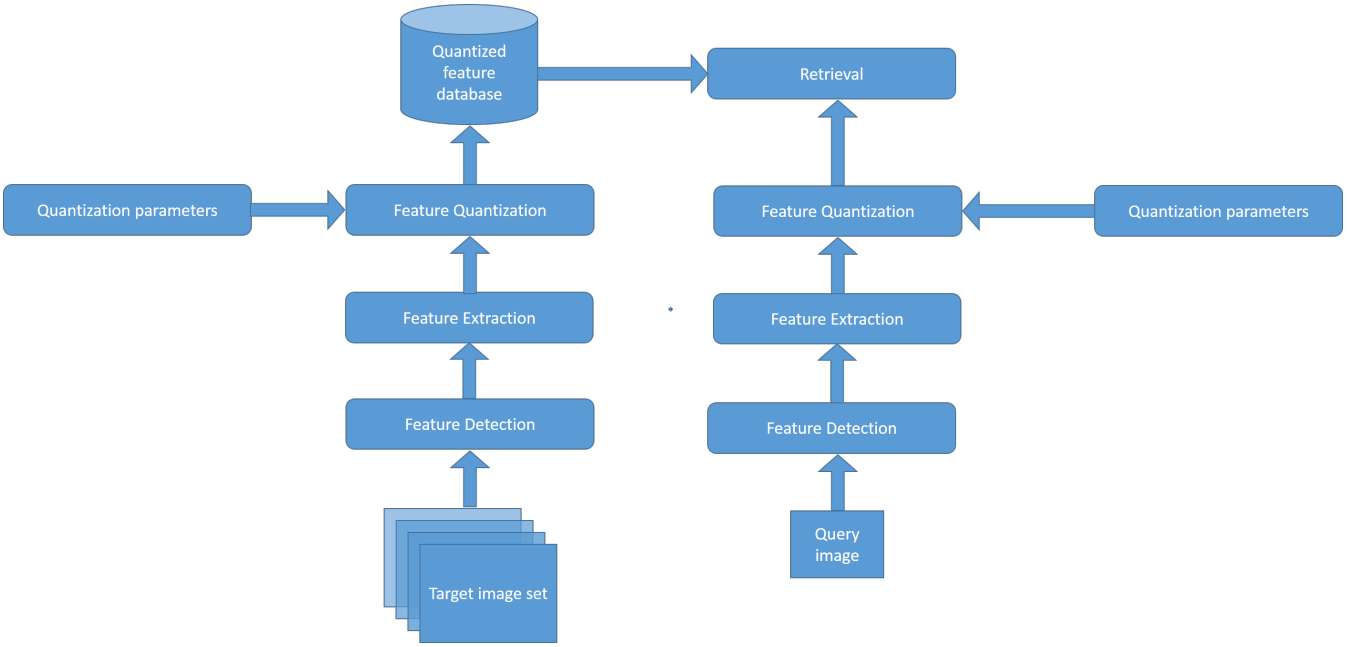


Figure 1: Overview of the framework.

binary hashing. Each of the feature types is paired with each of the aggregation mechanisms to yield four different configurations.

SIFT features describe the local characteristics of an image patch in terms of edge strength and orientation. These features can either be extracted at keypoints as detected based on image saliency or on a dense regular grid. We employ the former. SIFT features are orientation and scale invariant.

Our region-based shape features are based on moments. Shape features require a segmentation step to delineate the regions. They are also orientation and scale invariant.

Both feature extraction methods produce varying numbers of features per image. This raises the question of how to perform matching using all the features. We therefore investigate two approaches. First is to aggregate the features for an image into a single, fixed dimensional descriptor and then use this descriptor to compare the query image with the reference set. An effective way to do this is to quantize the features into a fixed number of canonical features and then count the number of these features that appear in an image. The quantization is performed by clustering a large number of features into a dictionary of words and then assigning each of the features in an image to one of these words. The final descriptor is then simply the word count for an image. Its length is equal to the number of words in the dictionary. This first approach to aggregation is thus called bag of words.

The second aggregation technique performs separate matches for each of the features in the query image and then computes the intersection of the results. This is potentially an expensive operation though and so we use hash-based indexing to make it more efficient. The features are first binarized using locality-sensitive hashing. The binary features are then indexed using a multi-index hash. This allows us to efficiently retrieve the reference images containing binary features that are similar in Hamming space to the binary features in the query image. The number of votes a reference image gets determines the best match.

We measure each configuration's performance with respect to the following metrics:

Accuracy If the location retrieved for a given query image matches the ground truth, the query is considered a success. Accuracy is the computed as the fraction of successful matches over a set of queries.

Efficiency The efficiency of a technique is how fast it can produce a result for a given query. A technique is efficient compared to another if its average response time is less than that of the other technique.

Scalability This metric measures the change in accuracy and efficiency as the search area grows.

We now provide more details on our approach.

4.1 SIFT Features

Scale invariant feature transform (SIFT) [10] is a popular method used to locate and extract local features in an image. SIFT refers to both the keypoint detection as well the feature extraction. For keypoint detection, SIFT computes differences of Gaussians at multiple image scales and detects local maxima. The image patch centered on the keypoint is then normalized with respect to its dominant direction and described by a 128-dimensional vector derived from 8x8 orientation histograms in a 4x4 grid.

4.2 Shape Features

An image must be segmented into regions before shape features can be extracted. We employ the maximally stable extremal regions (MSER) [5] approach to perform the segmentation. MSER finds segments that stay maximally stable over different thresholds, making the results invariant to illumination and baseline changes.

I_0	$\eta_{20} + \eta_{02}$
I_1	$(\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$
I_2	$(\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$
I_3	$(\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$
I_4	$(\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + 3(\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} - \eta_{12})^2 - (\eta_{21} + \eta_{03})^2)]$
I_5	$(\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$
I_6	$(3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$

Table 1: Hu moments.

We then extract shape moments to describe the regions. To make the features as invariant to scale and orientation as possible, we extract normalized central moments.

First, the spatial moments, m_{ji} for a shape or region are computed as

$$m_{ji} = \sum_{x,y} (S(x,y) \cdot x^j \cdot y^i) \tag{1}$$

where S is the shape being considered and x, y are the pixel coordinates. The central moments are then computed from the spatial moments,

$$\mu_{ji} = \sum_{x,y} (S(x,y) \cdot (x - \bar{x})^j \cdot (y - \bar{y})^i) \tag{2}$$

where $(\bar{x},\bar{y})$ is the center of mass: $\bar{x} = m_{10}/m_{00}, \bar{y} = m_{01}/m_{00}$.

Finally, the normalized central moments are computed as,

$$\eta_{ji} = \frac{\mu_{ji}}{m_{00}^{(i+j)/2+1}} \tag{3}$$

Note that $\eta_{00} = 1$ and $\eta_{10} = 0$ and thus are not included in our features. The normalized central moments considered are η_{02}, η_{03}, η_{11}, η_{12}, η_{20}, η_{21}, and η_{30}, thus giving the normalized central moment feature a dimension of seven.

We also compute Hu moments [9] which are based on the normalized central moments. Table 1 shows the construction of Hu moments from the normalized central moments. The Hu moments also have seven dimensions so our final region shape feature has 14 dimensions. We refer to our shape features as MSER features.

4.3 Bag of Words Aggregation

BOW [4] represents an image by way of a histogram of the frequency of the "visual words" in the image. The "visual words" are built using the underlying features and represent classes of similar features. The set of distinct visual words is called the codebook or the vocabulary.

4.3.1 Vocabulary Generation

An important part of a BOW model is its vocabulary. The vocabulary dictates how many words are needed to describe an image, i.e., the dimensionality of the feature, and what those words should be. Ideally, the visual words in the vocabulary should be those that are representative of as many features as possible in the original feature space. In practice, this is achieved through clustering the features from the original feature space typically using k-means.

The number of visual words in the vocabulary (the number of clusters produced by the clustering) is a design decision that determines the dimensionality of the quantized descriptors. This choice depends on many factors such as the underlying dataset and the type of performance required.

4.3.2 Descriptor Quantization

Once we have the vocabulary, each image is then described in terms of the visual words from the vocabulary, essentially making an image a bag of visual words. To do this, each of the image features is mapped to the closest cluster usually through a nearest neighbor search [6] and then assigned the word associated with that cluster. In this work, we use FLANN [11] which is a faster, though approximate variant of the nearest neighbor algorithm. The final image-level descriptor is then the word count. That is, each component of a descriptor indicates the number of times a particular word appears. This descriptor is normalized by the total number of words in the image

4.4 Locality Sensitive Hashing

Locality sensitive hashing (LSH) [7] is a dimensionality reduction technique often used for efficient nearest neighbor search. LSH is actually a family of techniques that encompasses different methods to generate the low dimensional representation, with the important aspect being that any two low dimensional vectors produced by LSH are at least as near to each other as are their high dimensional counterparts. Thus LSH promises a low dimensional version of the data that maintains the similarity measure of the original one.

Of the many techniques to perform LSH, the one we use here is random projections [2]. A number of random hyperplanes are generated with the same dimensionality as the original features. For each of the features, the side on which it lies to a particular hyperplane determines the bit for that hyperplane. Given a vector v from the original feature space and a vector r representing one of the hyperplanes, the bit value $h(r)$ for that hyperplane is determined by,

$$h(r) = sgn(v \cdot r). \tag{4}$$

Again, the number of hyperplanes is a design choice and determines the dimensionality of the final binary features.

We use LSH to convert the SIFT and shape features separately into binary representations for efficient matching through multi-index hashing.

4.5 Multi-Index Hashing

Multi-index hashing (MIH) is a framework proposed in [14] for the fast matching of binary features. Similarity search using binary features can be computed using the Hamming distance and hashing can be used to lookup features that are within a certain Hamming distance of each other. However, this lookup grows combinatorially with the length of the binary features. MIH partitions the binary features into substrings which are individually hashed more efficiently using smaller Hamming distances. The full Hamming distance is then computed using these approximate re-

sults. This cost of this extra step is negligible compared to the savings from the reduced number of hash table lookups.

4.6 Framework

We compare different combinations of feature extraction and aggregation methods. Since the intended application is to search over a very large dataset, typically overhead images covering vast geographical regions, we perform different experiments against an increasing reference dataset size. As most of the large maps and images do not come as a single homogeneous image, but rather as a set of adjacent images, we allow for the framework to accept a set of such images. And as the reference dataset has very large images, they are broken up into tiles of 256×256 or 512×512 pixels, with each tile given a tile ID, and passed as inputs to the pipeline. An overview of the framework is presented in Figure 1

4.6.1 Evaluation Strategy

Given a query image, we need to measure the performance of the different methods in terms of accuracy and efficiency. The query image is randomly sampled from any of the input images that constitute the reference/target set. The ground truth for these randomly sampled query images is easily obtained by calculating the amount of area overlap the query has with the reference set tiles. As shown in Figure 2, for a query size equal to the reference tile size, at most four tiles will have a non-null amount of overlap, and thus there will be at most four tiles in the ground truth. The number of ground truth tiles can vary depending on the query and the reference tile sizes, with a smaller ground truth set for smaller query images and larger set when the query image size is greater than the reference tile size.

The accuracy is measured using top-n matched results. Since in this work we are mainly concerned with retrieving the unique reference image tile, we consider the top-1 accuracy. That is, if the first tile returned by the search pipeline overlaps with the query, we consider the query to be a success.

5. EXPERIMENTS

5.1 NAIP dataset

For the experiments, we use the aerial imagery provided by The National Agriculture Imagery Program (NAIP)[2].

The imagery is acquired at a one-meter ground sample distance (GSD). The images are 3 band natural color (red, green and blue) and are compressed in .jp2 format. For a particular region, the imagery is made available as set of images that can downloaded using the tool provided on the NAIP website. Each image measures 8831×6964 pixels. These images are broken into 256×256 or 512×512 pixel tiles and used as the reference set. Figure 2 shows a sample of the NAIP imagery for the San Francisco Bay Area.

5.2 Experimental Protocol

We test the performance of different combinations of the methods by varying three major parameters:

- Changing the target set size
- Changing the target tile size
- Changing the query tile size

For each of the combinations, an experiment is performed and results collected. Each experiment contains 10 runs with 100 queries each and each run contains a query sampled randomly from the target set. (The query tile is not necessarily aligned with the target tiles as shown in figure 2.) To keep the comparisons reproducible and fair between different runs, the random number generator for each run is initialized with a seed from the same set of seeds used across all experiments. The results reported are an average across all the runs.

6. RESULTS

6.1 Varying the Target Set Size

We tested different configurations using an increasing set of target set sizes. This is to simulate the growing scale of the dataset and to determine the scales at which different approaches start breaking down. For images of the San Francisco Bay Area, our set of target set sizes are 1, 2, 4, 8, 16, 32, 64. A target set size of 64 corresponds to over 4,000 square kilometers or approximately 60K target tiles of size 256×256.

6.1.1 Comparison of SIFT Features

6.1.1.1 Average accuracy.

Figure 3a shows the accuracy of hashing against that of the BOW model for the various target set sizes and the two tile sizes (here we set the query and target tile size to be the same). As can be seen, hashing consistently outperforms BOW while achieving close to near 100% accuracy even on very large target set sizes. This indicates that features binarized through LSH retain most, if not all, of the distinctiveness of the original SIFT features. Additionally it also shows that MIH is very good at correctly retrieving those features.

We can also see that hashing with 512×512 tiles is marginally better than with 256×256 tiles. Although the difference in performance is not very significant for hashing, it is non trivial for BOW. Interestingly for BOW, the smaller tile size seems to perform better for larger target set sizes. As BOW features are global features unlike that of LSH, it might indicate that the quantized features produced by smaller test sizes are more distinguishable than those produced by the larger tile size. It can be argued that this might be caused by the less amount of variance in smaller tiles.

6.1.1.2 Average Query Times.

As can be seen from Figure 3b, querying with MIH using binary features is significantly slower than querying with BOW quantized features. This can be attributed to two factors. Firstly, hashing has to search for a significantly larger number of features as each of the features are local features rather than global ones. Secondly, MIH performs the search over multiple hash tables. This can vastly increase the number of duplicate candidates found and thus the duplicate elimination stage of MIH can contribute significant overhead. Also, if a match is not found for a substring in a hash table bucket, the search continues until either it goes beyond the desired search radius or it reaches the end of

[2]http://www.fsa.usda.gov/programs-and-services/aerial-photography/imagery-programs/naip-imagery/index

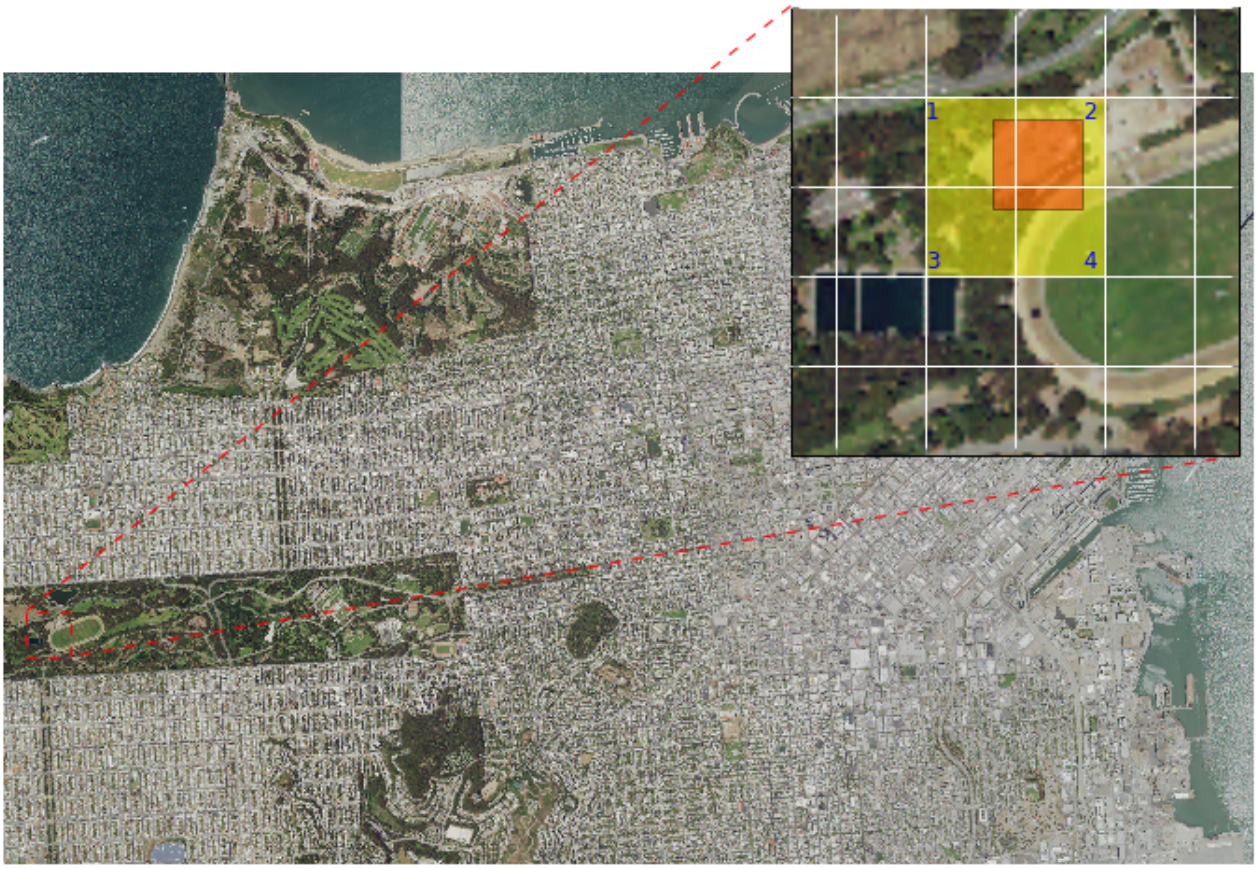


Figure 2: A sample of the dataset, showing part of the San Francisco Bay Area. The image in the inset shows how the evaluation is carried out. For a query patch (red), the ground truth is the four overlapping tiles. If the top-1 returned result from the matching framework is one of these tiles, the query is considered to be a success.

the bucket. For large target set sizes, each individual bucket can be very large.

It is also quite apparent that the rate of increase of the query times with the target set sizes for both methods is large. This can indicate that scaling up the target set size will need better techniques or methods if the performance has to be kept in real time limits.

6.1.2 *Comparison of MSER Features*

6.1.2.1 *Average Accuracy.*

As MSER features are shape features, they are not as localized as SIFT features. This may cause the accuracy of these features to suffer, as can be seen in Figure 3c. Although, here too hashing outperforms BOW on all target sizes. For the single target image case, it performs with an accuracy of 59.2% and 74.8% for the 256 and 512 pixel tile size cases. This indicates two things: 1) The underlying MSER features do provide some degree of location repeatability and 2) even for simple descriptors like moments, the degree of accuracy is good, if not acceptable. On the other hand, BOW quantized features show very poor performance. This can be attributed to the fact that the features derived in this case go through two types of quantization, first via the shape descriptors and second via BOW. This may cause these features to lose most of their distinctiveness.

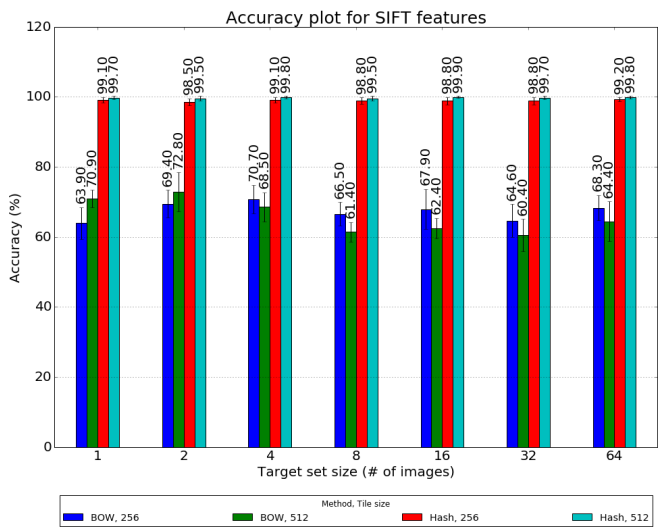
As for the tile sizes, a general trend can be seen wherein tile sizes of 512 pixels perform better than 256 pixels. Interestingly, this trend undergoes a reversal in the case of hashing for target set sizes starting around 16. However, the accuracy for each case is within the standard deviation of the other, so this reversal may not be statistically significant.

6.1.2.2 *Average Query Times.*

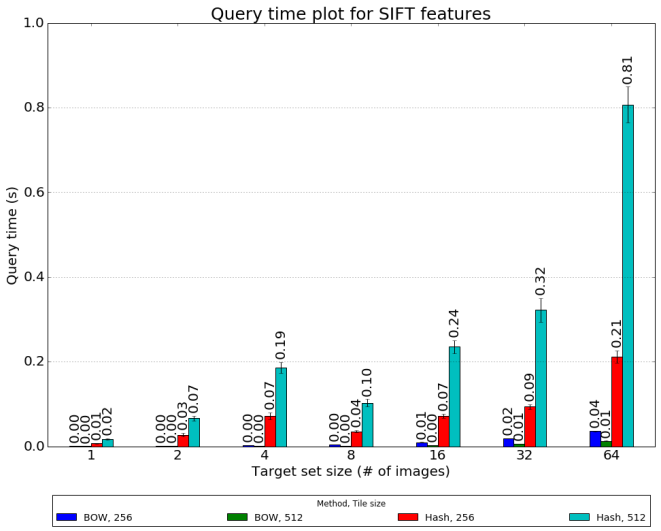
Figure 3d shows the average query times as we increase the target set size. Compared to SIFT features, retrieval over MSER features is much faster. This is due to the fact that there are significantly fewer MSER features and the features themselves are of lower dimension. The same rate of increase in average query times is observed as in the case of SIFT.

6.2 Varying the Query Tile Size

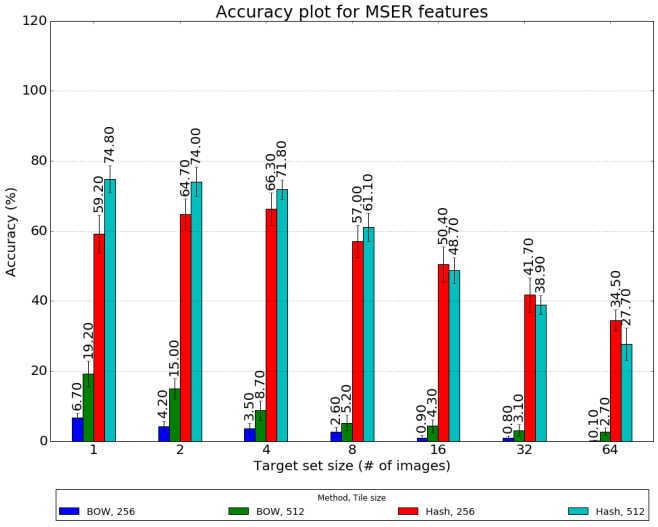
Real world queries that may be presented to the framework may not be of the same size as the reference tile sizes. The framework has to be able to handle different sized query images. For instance, if the query image is smaller than the reference tile size, it will contain fewer features that can be used to match against the dataset. This may lead to



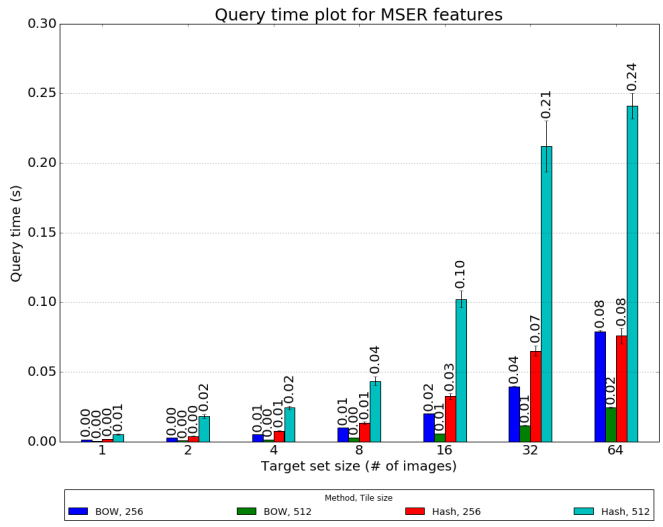
(a) SIFT accuracy



(b) SIFT query time



(c) MSER accuracy



(d) MSER query time

Figure 3: Performance of hashing and BOW for varying target set sizes. Each target image measures approximately 9K×7K pixels.

reduced accuracy. To quantify this, we test the methods against query tile sizes of 64, 128, 256 and 512 pixels. The reference tile sizes are still either 256 or 512.

6.2.1 Comparison of SIFT Features

6.2.1.1 Average accuracy.

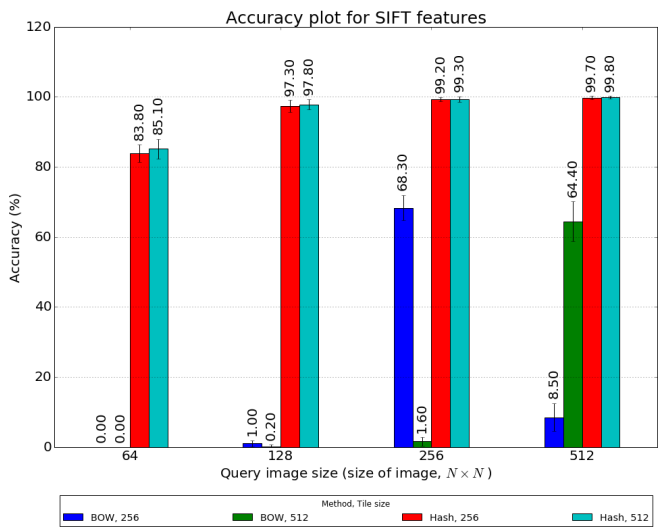
Figure 4a shows that the accuracy of hashing improves as the query tile size increases. This is because large image sizes provide a larger number of features that can be used to accurately match against the target dataset. The gap in performance for the two different reference tile sizes can be seen to be closing as we increase the query tile size. This can indicate that for larger query sizes with the SIFT-MIH combination, the reference tile size may not be a very important parameter. It is also worth noting that the performance is good even for small query tile sizes. For example, for 64×64 query tiles, the accuracy is 83.8% for 256×256 sized target tiles and 85.1% for 512×512 sized target tiles.

Comparatively, BOW performs poorly if the query tile size does not match the target tile size. This can be expected as the BOW features represent the entire tile and thus may differ significantly if the tile sizes are different.

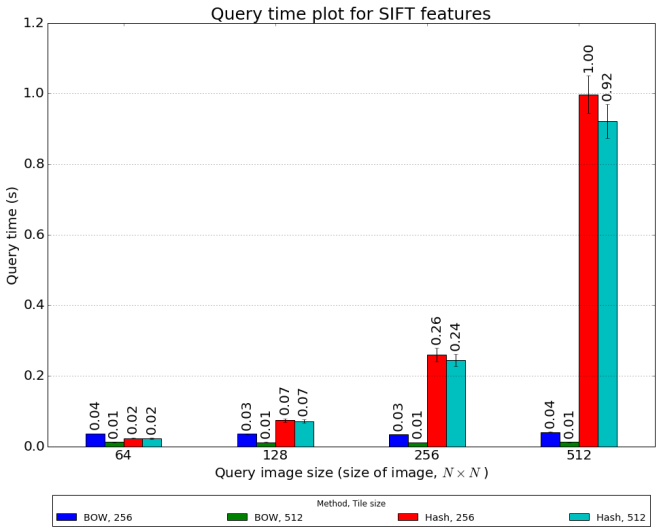
6.2.1.2 Average Query Times.

The query times over the varying query tile sizes mimic that over the target set sizes as shown in Figure 4b. Here again we see a rapid increase in the times in the case of hashing, but not in the case of BOW. Again, MIH using binary features slows down significantly as the query size increases. This is expected since the number of features will grow quadratically with image size. The reasons behind this rate of growth in average query times are the same as in the varying target set size case.

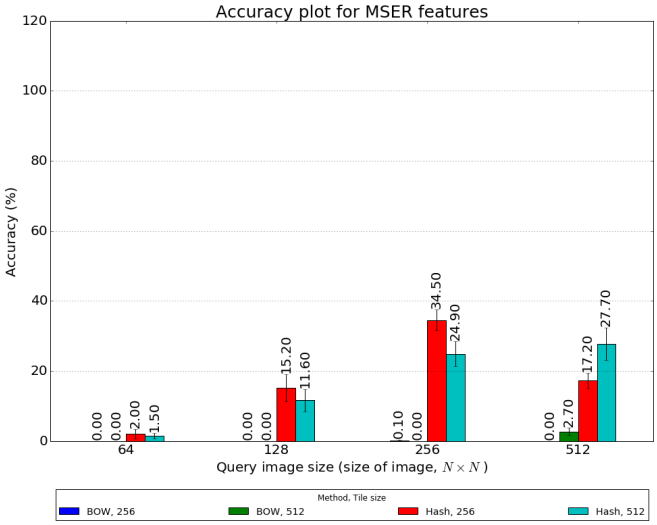
6.2.2 Comparison of MSER Features



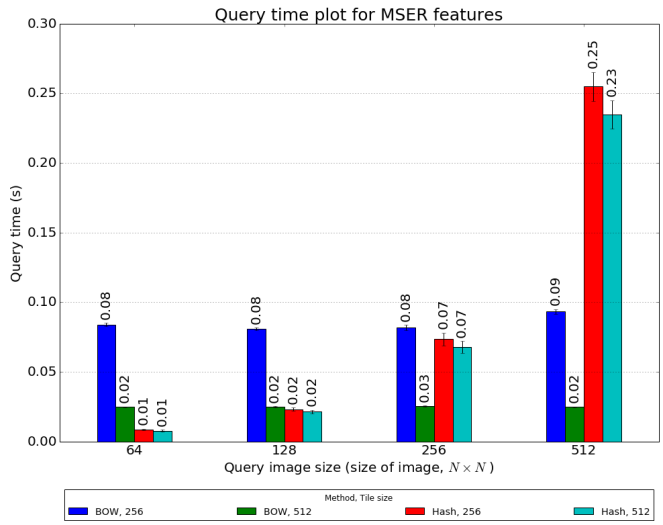
(a) SIFT accuracy



(b) SIFT query time



(c) MSER accuracy



(d) MSER query time

Figure 4: Performance of hashing and BOW for varying query tile sizes.

6.2.2.1 *Average accuracy.*

Figure 4c shows the accuracy for MSER with hashing and BOW. BOW fares poorly consistently, while the accuracy of hashing tends increases with query tile size. This shows that moment features quantized through BOW tend to lose practically all their distinctiveness, especially for smaller query tile sizes.

6.2.2.2 *Average Query Times.*

The average query times for MSER features are shown in Figure 4d. The general trend of increase in query times with the increase in the query tile size is once again observed. Hashing tends to be as fast as BOW in the case of query sizes of 64 and 128 pixels, but then tends to slow down exponentially from queries of size 256. BOW on the other hand stays consistent throughout. Also, as the blue bars show, BOW with 256 pixel target tile size is quite slow compared to 512 pixel target tile size. This is due to the fact that for the same sized target set (in terms of square kilometers), 256 pixel tiles will produce significantly more tiles, leading to a larger number of descriptors.

7. CONCLUSION

In this paper, we conducted an empirical study of state-of-the-art image retrieval algorithms for the goelocalization of overhead imagery on a large scale. We compared two feature extraction algorithms, SIFT and MSER, combined with two quantization and retrieval methods, hashing and BOW. We measured and quantified the performance of these methods in terms of accuracy, efficiency, and scalability. The results indicate that, in terms of accuracy, hashing outperforms BOW. However, hashing suffers at very large target set sizes in terms of efficiency.

In future work, we plan to investigate hierarchical feature representations for large scale image geolocalization. We anticipate these methods will be beneficial in maintaining good efficiency while achieving near perfect accuracy.

8. REFERENCES

[1] F. Caballero, L. Merino, J. Ferruz, and A. Ollero. Vision-based odometry and SLAM for medium and high altitude flying UAVs. *Journal of Intelligent and Robotic Systems*, 54(1):137–161, 2009.

[2] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 380–388, New York, NY, USA, 2002. ACM.

[3] D. Costea and M. Leordeanu. Aerial image geolocalization from recognition and matching of roads and intersections. *CoRR*, abs/1605.08323, 2016.

[4] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[5] P. E. Forssen. Maximally stable colour regions for recognition and matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[6] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226, Sept. 1977.

[7] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pages 518–529, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[8] J. Hays and A. A. Efros. IM2GPS: Estimating geographic information from a single image. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[9] M.-K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962.

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[11] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *In VISAPP International Conference on Computer Vision Theory and Applications*, pages 331–340, 2009.

[12] J. M. Murphy, J. L. Moigne, and D. J. Harding. Automatic image registration of multimodal remotely sensed data with global shearlet features. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3):1685–1704, March 2016.

[13] S. Newsam, D. Leung, O. Caballero, J. Floreza, and J. Pulido. CBGIR: Content-based geographic image retrieval. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 526–527, New York, NY, USA, 2010. ACM.

[14] M. Norouzi, A. Punjani, and D. J. Fleet. Fast search in hamming space with multi-index hashing. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3108–3115, June 2012.

[15] T. Senlet, T. El-Gaaly, and A. Elgammal. Hierarchical semantic hashing: Visual localization from buildings on maps. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2990–2995, Aug 2014.

[16] S. Workman, R. Souvenir, and N. Jacobs. Wide-area image geolocalization with aerial reference imagery. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3961–3969, Dec 2015.

[17] C. Wu, J.-M. Frahm, F. Fraundorfer, and M. Pollefeys. Image localization in satellite imagery with feature-based indexing. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2008.

[18] Y. Yang and S. Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 270–279, New York, NY, USA, 2010. ACM.

[19] Y. Yang and S. Newsam. Geographic image retrieval using local invariant features. *IEEE Transactions on Geoscience and Remote Sensing*, 51(2):818–832, Feb 2013.