Offline Mobile Instance Retrieval with a Small Memory Footprint

Jayaguru Panda[1] Michael S. Brown[2] C. V. Jawahar[1]
[1]CVIT, IIIT Hyderabad, India [2]NUS, Singapore

Abstract

Existing mobile image instance retrieval applications assume a network-based usage where image features are sent to a server to query an online visual database. In this scenario, there are no restrictions on the size of the visual database. This paper, however, examines how to perform this same task offline, where the entire visual index must reside on the mobile device itself within a small memory footprint. Such solutions have applications on location recognition and product recognition. Mobile instance retrieval requires a significant reduction in the visual index size. To achieve this, we describe a set of strategies that can reduce the visual index up to 60-80 $\times$ compared to a standard instance retrieval implementation found on desktops or servers. While our proposed reduction steps affect the overall mean Average Precision (mAP), they are able to maintain a good Precision for the top K results (P_K). We argue that for such offline application, maintaining a good P_K is sufficient. The effectiveness of this approach is demonstrated on several standard databases. A working application designed for a remote historical site is also presented. This application is able to reduce an 50,000 image index structure to 25 MBs while providing a precision of 97% for P_{10} and 100% for P_1.

1. Introduction

Mobile image retrieval allows users to identify visual information about their environment by transmitting image queries to an online image database that has associated annotations (e.g. location, product information, etc) with the images. However, this is reliant on a network connection to transmit and receive the query and retrieved information.

This paper examines mobile image retrieval for *offline* use when a network connection is limited or not available. In this scenario, the entire visual search index must reside on the mobile device itself. More specifically, we are interested in "instance retrieval", where the annotations associated with the images (e.g. building's name, object information) are returned by the query and not the images themselves. Figure 1 shows an example use case where mobile camera photos are used to identify buildings and landmarks without the need for a network connection.

Figure 1. Example use of our method where landmark location can be performed offline by indexing an image collection within a small memory footprint for use on a mobile device.

While our targeted instance retrieval does not need to store the images, the entire visual index structure needs to fit on a mobile device, ideally within a small footprint (e.g. 16-32MB). This small memory footprint serves two purposes. First, while mobile phones have up to 16-32GB of storage, this is mainly in the form of slower flash memory that is an order of magnitude slower than RAM. Having the entire index within tens of MBs makes it possible for use in a resident application on the phone's RAM. Second, this small size is inline with common practices for mobile applications; e.g. the iPhone average application size is currently 23MB. Additionally, iPhone apps less than 50MB can be downloaded using 3G/4G, anything large must be downloaded using a wireless connection [1].

Contribution We introduce a series of pruning steps that can significantly reduce the index size and incorporate partial geometry in it. These include a simple vocabulary pruning based on word frequency, exploiting a 2-word-phrase based index structure, placing constraints on feature geometry and encoding the relative geometry of visual words. Each step is discussed in detail and a running example using the Oxford building dataset is used to show the effects on the index size and performance. While this reduction does come at a cost of a lower mAP, our reduction strategies do not adversely affect the mean precision at K (P_K).

In fact, our collective strategies are able to reduce the visual index up to 60-80 $\times$ with virtually no change in P_{10}, P_5 and P_1. As we show, this is perfectly suited for mobile instance retrieval application. Higher precision for smaller K is important since, the eventual goal of the solution is to transfer the information from the top-K results to the query image.

1.1. Related Work

Image recognition and retrieval for mobile devices already has several successful examples. Commercial apps like Google Goggles, Amazon Snaptell, Nokia's Point and Find, are current popular examples. Most of these mobile apps use a client-server model and communicate with a remote server for matching within large image databases [12, 6, 5, 7]. Some of these applications send the image to the server, while other implementations send a compressed feature representation over to the server [4, 10, 18]. We address this problem in a far more challenging setting, where the entire computation must happen on the mobile device. This requires the entire database of information to reside on the device storage. Prior work targeting this offline scenario [11, 13, 27, 30] have only been demonstrated on image databases of the order of hundreds of images. We target databases at least 10-100 times larger where the mobile device should store a compact search index to instantly recognize the query and annotate it. In order to avoid any time-consuming post-processing, we need the search index to effectively borrow the advantages of geometric verification into index structure.

Image recognition and retrieval primarily involves matching local image features which represent an image's appearance and geometry, to those in the database. Using Bag-of-Words approaches [20, 23, 26], scalability is achieved with the help of (i) quantization, which enables the efficient computation of similarity and (ii) order-less description of image features, making the representation invariant to many popular transformations. Variations of this approach have been motivated either towards building a compact and efficient search index or, a geometry induced search framework.

Decreasing the memory footprint of the search index has been earlier addressed targeting desktop environments [15, 17, 32]. Identifying a useful subset of training features that are robust and distinctive and using only these for specific retrieval tasks can achieve significant memory savings without loss in matching performance [29]. Hamming embedding techniques for feature representation in the form of binary strings have been introduced to efficiently work with smaller vocabularies [14]. Such techniques have also been employed for binary representation of global image features like GIST [28]. A popular alternative approach to inverted files in large-scale search is the min-hashing technique [9, 8]. Usage of Fisher Vectors [22] and vector of locally aggregated descriptors (VLADs) [16] for large-scale image search has been demonstrated with compact image vectors using joint optimization of dimensionality reduction and indexing for precise vector comparisons.

Words alone may not be meaningful to retrieve semantically related images. In such cases, one could use larger constructs (e.g. phrases) defined over multiple words as the basic primitive [19, 25]. A few attempts have introduced novel techniques which can help in phrases with index structures like min-hash [8, 34]. Introduction of geometry constraints into the phrase or index definition [21, 34, 35] often helps in improving the recall of the rare objects in large databases. However, with introduction of geometry, the search index size grows. Our approach attempts to achieve a compact geometry preserving indexing by efficiently utilizing the advantages of visual-phrase representation of images. We prune the largest possible 2-word-phrase space to design a compact solution.

2. Reducing the Index Footprint

There are two broad directions in image retrieval: (a) category retrieval and (b) instance retrieval. This work addresses the latter, which focuses on retrieving images and/or associated information related to a specific instance/object, where the query can be from a widely varying imaging condition (like scale, view-point, illumination, etc) compared to those in the database. Instance retrieval literature has been dominated by the Bag-of-Words (BoW) method and its variations [9, 20, 23, 26].

In this section, we start by analysing the memory and storage requirements of the instance retrieval solution implemented as an inverted index search with a BoW representation. We experiment on the Oxford-5K dataset [23] to demonstrate the quantitative performance of various indexing strategies. This dataset contains 5,062 images from 11 different landmarks, each represented by 5 possible queries.

Local features (often SIFT vectors) are first extracted at interest points and quantized with the help of a precomputed vocabulary. With quantization, a 128-Byte SIFT representation gets converted to a compact visual word representation of 4 Bytes (or smaller depending on the vocabulary size). Visual word representations are attractive for their invariance to the common transformations and compactness. Database images get represented as a histogram of visual words that are indexed with an appropriate index structure (often an inverted index). During the retrieval, SIFT vectors are extracted at interest points from the query image. They are quantized into visual words, and a set of images having most of these visual words in common, are retrieved from the database. Weighting visual words based on a TF-IDF measure has shown to improve the performance [26]. Even then, this list could contain many false positives. As such, the retrieved images are matched with the query image, and

re-ranked based on the matching scores. Often, fitting a fundamental matrix or geometric verification (GV) of the two images is the basis for matching.

The choice of quantization scheme is a trade off between accuracy vs efficiency. Clustering with a flat k-means approach gives better results, but the quantization is linear in terms of the vocabulary. A vocabulary tree built with hierarchical k-means (HKM) [20] has logarithmic order and highly speeds up the vocabulary assignment, with small compromise on clustering results. Approximate k-means (AKM) [23] also has similar time and memory complexity as HKM. We choose to use HKM for our experiments.

For many instance retrieval tasks, the vocabulary size can be as high as 1M. If there are 5K images, the histogram representation could be as large as 5GB ($5\text{K} \cdot 1\text{M} \cdot 1\text{Byte}$). An efficient implementation with an inverted index can do this in around 110MB. An additional variable length encoding (VLE) technique could be used to further compress the index size. This, however, incurs additional decoding costs which may be undesirable on a mobile device. We report and compare the memory footprint without using VLE compression techniques for all our experiments. In addition, we need 2.3 GB memory to store the SIFT vectors with keypoint information, which are required during the final Geometric Verification (GV). Note, this does not count the storage of the image database. In Table 1, we present the space requirement as well as the retrieval performance of a desktop (BoW-D) implementation of a retrieval system on Oxford Building dataset. We show performance with and without GV. While GV improves the mAP and P_K, it requires significant computational resources, in addition, the resulting memory requirement is unacceptable for mobiles.

2.1. Baselines with Pruned Visual Words

We first reduce the memory requirement with two simplifications. First, for the problem of our interest, we do not store the images on the phone (we only need the annotation information or tags). This limits the storage to only the features and representations. Second, as done by other retrieval systems, instead of using SIFT vectors for GV, we match the visual word indices in the corresponding images. This reduces the memory requirement by a factor of 32 (i.e., $\frac{128}{4}$), without visible reduction in performance, as can be seen in Table 1 (see row:BoW-WGV).

A large vocabulary is one of the primary reason for the large index size; e.g. instance retrieval solutions need large (closer to 1M) vocabulary for finer quantization [23, 24]. Smaller vocabularies reduce the search index, but at the cost of performance. We desire a smaller vocabulary with finer quantization. We do this by pruning the vocabulary by discarding some of the visual words while indexing.

Instead of directly computing a smaller vocabulary, we prune the large vocabulary and define a smaller one by recursively removing the least influencing visual words. Pruning is done by identifying and eliminating visual words that are the least useful for the retrieval task. Borrowing the idea behind TF-IDF, visual words that occur in a large number of images or conversely, in few outlier images are treated less discriminative. A visual word occurring in greater than T_H images or, less than T_L images is removed (T_L and T_H are experimentally determined). That is, instead of weighing the visual words based on TF-IDF, we discard (i.e., assign a zero weight) the least scored visual words and work only with the informative visual words (with weight as one). Using such a vocabulary pruning, we reduce the vocabulary size from 1M to 750K, with only a minor reduction in performance as can be seen in Table 1. Results of the pruned vocabulary (BoW-Pr and BoW-PrGV) are comparable to the original.

	Voc	Index	mAP	P_{10}	P_5	P_1
BoW-D	1M	110MB	.580	.75	.87	.88
BoW-D-GV	1M	2.4GB	.618	.79	.92	.93
BoW-WGV	1M	510MB	.616	.78	.92	.93
BoW-Pr	750K	80MB	.555	.75	.88	.91
BoW-PrGV	750K	480MB	.582	.77	.91	.92
Base-Phr	530K	58MB	.592	.78	.92	.93

Table 1. Baseline instance retrieval on the Oxford Buildings 5K dataset using the BoW framework. Last two rows reflect results with pruning and using 2-word-phrases respectively.

2.2. 2-word-phrase Search Index

Using a set of neighboring words (often referred to as phrases) is often more informative than isolated words. This has been attempted in various forms [3, 8, 25, 31]. Focus of most of the previous work has been on obtaining higher quantitative performance (i.e., mAP) during the retrieval. In this work, we first define a phrase as a pair of visual words, and show how that can help in the instance retrieval task.

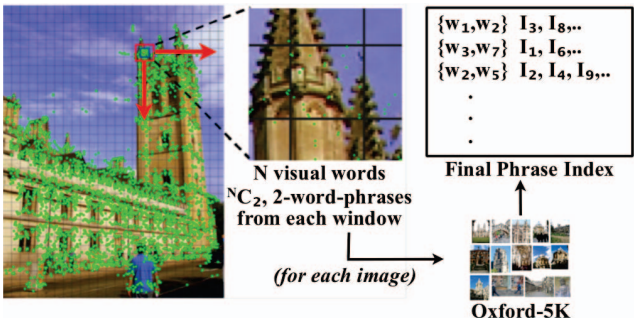

Figure 2. Building a 2-word-phrase index: Each image, represented by BoW quantized features, is processed to obtain phrases by dividing the image into "overlapping" grids and indexing 2-word combinations in each grid. In the final index, only phrases that index more than one image are retained.

A 2-word-phrase is detected from an object and is indexed in search records, if it occurs repetitively in images containing that particular object. Such phrases can be represented as $p_k = \{w_i, w_j\}$ such that w_i is in the neighborhood of w_j, and indexed in a similar manner to that of visual words. For implementation, we consider the neighborhood as a fixed size of 50 pixels. To avoid duplicate 2-word-phrases in the index structure, we constrain $w_i \leq w_j$. p_k is indexed in a hash-map with $\{w_i, w_j\}$ as a key. The map data structure is typically implemented as binary search trees with logarithmic look-up time. Extracting higher order phrases increases the computational requirements during query processing. We find the additional computations are not affordable on low-processing devices. Further, indexing higher order features (≥ 3 words) is harder and do not always enhance the retrieval performance [33].

With a vocabulary size of 1M, the number of possible 2-word-phrases could be as large as 10^{12}, however, this does not happen in practice. We first find a set of phrases that are prominently present in the database. For example, phrases formed with a vocabulary as small as a 500K, one can obtain performance comparable to that of 1M visual words (see Table 1) without any explicit geometric verification. Even in this case, the subset of relevant phrases are computed based on the TF-IDF scores.

To speed up the computation, we divide the image into overlapping rectangular grids (see Figure 2) and choose combinations of visual word pairs from each grid. The overlap must be sufficient to ensure that true-positive 2-word-phrases are not missed out at grid-boundaries. A $40-50\%$ overlap is ensured between consecutive rectangular grids. Table 1 shows the comparison with traditional approaches. In row:Base-Phr, one can observe that the vocabulary is pruned to half the original size and the index requirement reduces by almost half to $58MB$. The mean average precision (mAP) increases by ~1% as compared to baseline BoW method (BoW-D) and reduces by ~2% as compared to BoW with geometric verification (BoW-D-GV). However, P_{10} and P_5 are comparable with the latter. Since we are motivated to look for the most similar image and not bother about the rank-list, the drop in mAP is ignored.

3. Geometry-aware 2-word-phrases

For our problem of instance retrieval, such as location recognition or product search, further constraints on the geometry can be assumed. For example, the variation of depth within an object will likely be small, even if the object/product is not planar. In this section, we demonstrate how the geometry can be effectively used in selecting a much smaller subset of phrases for our instance retrieval problem. In general, we constrain the phrases selected, based on (i) the scale of keypoints at the visual word location, (ii) the relative placement of the words, and (iii) the possibility of it contributing reliably for a geometric verification. We compare the results with the standard desktop version of BoW with GV (BoW-D-GV), and with the state-of-the-art image search techniques with compact representations: the Fisher Vectors (FV) and Vector of locally aggregated descriptors (VLADs) [17]. Note we are using the basic implementation for VLADs in [17], recent improvement gains in results have been demonstrated [2].

	Voc	Index	mAP	P_{10}	P_5	P_1
ScA-Phr	462K	41MB	.581	.78	.92	.93
SpA-Phr	280K	36MB	.579	.78	.92	.93
Scp-Phr	250K	28MB	.571	.78	.91	.92
SV-Phr	180K	16MB	.567	.77	.90	.91
BoW-D-GV	1M	2.4GB	.618	.79	.92	.93
FV	64	4.5MB	.298	.50	.66	.76
VLAD	256	23MB	.309	.49	.63	.69

Table 2. Results on using the Geometry aware 2-word-phrases. ScA-Phr, SpA-Phr, Scp-Phr and SV-Phr Phr refer to the the 2-word-phrase index methods with scale-aware, space-aware, both scale-space aware and spatial validity constraints respectively. The last three rows refer to the state-of-the-art Bag-of-words (as BoW-D-GV in Table 1), Fisher Vectors and VLAD approaches [17].

3.1. Scale-aware constraints

We define phrases only in a small neighborhood (50 pixels in practice) and assume both the visual words are at the same scale. This is mostly true except for situations of serious depth discontinuity. However, for situations like product search or location search, this is a reasonable assumption. Phrases defined using words with the same scale are referred to as scale-aware phrases. This also helps in selecting phrases that will be reliable for viewpoint variations.

During feature extraction, SIFT keypoints are detected at scale-space extrema points using the DOG-space pyramid, constructed at scale-levels that are multiples of 2. We separate scale according to log-scale $(log(s_i))$ value corresponding to the feature keypoints detected at s_i scale. Two visual words with keypoint scales s_i and s_j, are combined into a 2-word-phrase, if $|(log(s_i) - log(s_j))|$ is less than a particular threshold τ. For our implementation we use τ as *3*. The separation of scale allows to enlarge our rectangular window size to $70-100$ pixels as the number of 2-word-phrase combinations get reduced significantly. This accommodates more true-positives and reduces the probability of false-positives. In Table 2, with scale-aware phrases (ScA-Phr) the search index is reduced with a further pruned vocabulary, while precision P_{10}, P_5 and P_1 are preserved.

3.2. Space-aware constraints

Phrases capture larger geometry of the configuration of visual words. We constrain the geometry of occurrence of two visual words further by encoding the relative placement

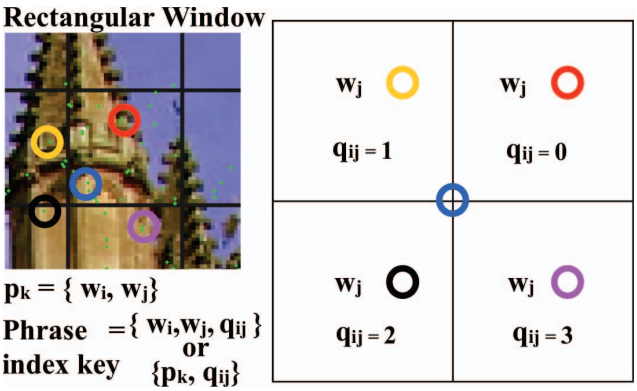

Figure 3. Space aware constraint is enforced with a two bit information q_{ij} along with the 2-word-phrase $\{w_i, w_j\}$ as shown.

of visual words in a neighbourhood. We embed a loose spatial constraint using just 2-bits of information. In a phrase p_k defined by $\{w_i, w_j\}$, where $w_i \leq w_j$, we choose w_i as the origin and determine in which rectangular quadrant w_j lies in (See Figure 3). Depending on the quadrant where w_j lies, a two-bit value of $q_{ij} = 0, 1, 2, or, 3$, the 2-word-phrase key is modified as $\{p_k, q_{ij}\}$ i.e. $\{w_i, w_j, q_{ij}\}$.

With just two-additional bits per index key, we further prune the vocabulary size and reduce the phrases from the search index. In Table 2, row:SpA-Phr corresponding to space-aware phrases, shows a decrease in the size of search index. Using both the scale and space constraints (row:Scp-Phr) together, we achieve a search index structure as small as $28MB$ with a vocabulary pruned to $250K$.

3.3. Spatial Validity Constraints

Motivated by the approach presented in [29], we retain useful features by validating the training features in each image using an unsupervised pre-processing. Useless features that come from occlusions and unstable keypoints are likely to exist in only a single image, while useful features are found in more than one image of the same object. To identify useful features, each image is used as a query and geometric verification is performed on the top M retrievals based on visual word matching. The features that score as inliers in this procedure are considered useful and hence retained. Other features are discarded. Using $M = 100$, we observe that this brings down the average number of features per image by 90% (See Figure 4).

Figure 4. Spatially validated features in example images. The red dots represent the rejected features after geometric validation based pre-processing, while green dots are the useful features. We can observe the unstable features due to occlusion are removed.

We now prune 2-word-phrases that have a component visual feature rendered useless by the above technique. Hence, the phrases obtained by the combination of space and scale aware constraints are further processed. This brings down the vocabulary size to $160K$ and the search index is reduced to $16MB$. As shown in Table 2 (row:SV-Phr), phrases constrained with spatial validity further compromise the mAP over the conventional BoW-D-GV approach but help in significant reduction of index size.

3.4. mAP vs P_K

Image retrieval research is often focused on achieving a superior mAP of the retrieved images. This is achieved by pulling out rare images or images which contain the object or location of interest in only a small part of the image. This is relevant when the objective is to retrieve from a "casual" database of images. We are working on a database of images with annotations (which are often prepared and specially annotated by a user). Given a specific view of a product or building, we are not interested in getting all the images of the same object. Instead, we desire a similar view in a reliable manner and possibly transfer the annotation to the query image. This needs highly similar images coming at the top of the retrieved list. Our retrieval process is designed to retain the highly similar images at the top of the list. This is done by (i) using the scale information while defining phrases and (ii) spatial constraints preferred images with similar view angles to be at the top of the list.

4. Results and Discussions

We demonstrate that the method of using 2-word-phrases in our search index, significantly reduces the size of the Inverted look-up index. By identifying 2-word-phrases that inherit robust geometrical constraints, we are able to further reduce the search index by 3-4 times. In the previous sections, we make use of Oxford-5K as our primary evaluation dataset. We now show results on three other datasets.

Evaluation Criteria: Image retrieval systems measure the performance in terms of the mAP. Average precision is the area under the precision-recall curve. This is computed for each query based on the ranked list of retrieved images, and the average performance on all queries give us the mAP. We are interested in search applications that are not concerned with the ranked list, but only the top image. Hence, although we take care of sharp drop in AP, only the Precision at K (P_K) matters. Hence, we measure the mean Precision at $10, 5, 1$ (P_{10}, P_5, P_1) to evaluate our performance.

4.1. Performance on Multiple Datasets

Paris Buildings [24] : This dataset has 6,386 high-resolution (1024×768) images of 11 iconic landmarks from

Dataset	#Images	#Queries
Oxford Buildings	5,062	55
Paris Buildings	6,383	55
UkBench Dataset	10,200	10,200
Heritage Structures	50,292	40

Table 3. Number of images in each dataset and the number of queries used for evaluation.

Figure 5. Random Sampling of images from each of the four datasets used for evaluation: Oxford Buildings, Paris Buildings, UkBench, and Heritage structures.

France. Five query images are taken from each of the 11 buildings to give a total of 55 queries. The images are marked by architecturally distinct landmarks unlike identical buildings in Oxford-5K, which creates a difference in the vocabulary used for quantization. With a vocabulary size of $1M$, we observe that the mAP is affected with our 2-word-phrase indexing technique. However, we were able to reduce the memory footprint significantly, compared to BoW approach with SIFT-based GV (BoW-D-GV).

UkBench dataset [20] : This dataset has 10,200 medium-resolution (640×480) images with 2,550 groups of 4 images each. Each group has images of a particular object from different viewpoint and angle variations. Since, the objects are very distinct, this dataset has large variations in appearance. A vocabulary of size $1M$ was built using features from all the objects. The evaluation criteria here is P_4 and P_1. Each image in the dataset is taken as a query image and performance is measured. It was observed that on average, we usually get more than 2 out of the 4 positives in the top-4. The size of the search index is reduced by 75 times, compared to BoW-D-GV.

Heritage Dataset : We introduce a novel data set of images from a world heritage site. This dataset has 50,292 medium-resolution (480×360) images of specific buildings, architectural details, interiors and exteriors of heritage monuments, etc. (See Figure 5, last two rows). Four query images are taken from each of the 10 distinct and iconic monumental structures we identified to evaluate on, giving us a total of 40 queries. This dataset has similarities with the Oxford-5K in terms of the identical buildings and structures, marked by a unique style of architecture. We measure performance in terms of precision at 10, 5, 1. We reduce the search index size significantly, however, P_K is preserved.

Dataset	Method	Voc	Index	P_{10}*	P_1
Paris (6,086)	BoW-D-GV	1M	1.9GB	.98	1.0
	Base-Phr	300K	51MB	.91	.95
	Geom-Phr	200K	34MB	.91	.98
ukbench (10,200)	BoW-D-GV	1M	1.2GB	.76	.92
	Base-Phr	300K	22MB	.65	.88
	Geom-Phr	200K	16MB	.68	.91
Heritage (50,292)	BoW-D-GV	1M	1.6GB	.97	1.0
	Base-Phr	350K	37MB	.96	1.0
	Geom-Phr	200K	25MB	.97	1.0

Table 4. Performance on the remaining three datasets. *: For the Ukbench dataset, we evaluate P_4 instead of P_{10}. Base-Phr refer to baseline 2-word-phrases and Geom-Phr shows results with geometry-aware 2-word-phrase indexing.

4.2. Analysis

Selecting a phrase or co-occurrence of visual words from a query image to match with a similar co-occurrence in the database brings in spatial constraints during the matching procedure. Along with further addition of loose-geometry to the pair of visual words in a phrase, the constraints become stronger in order to avoid false matches.

With the help of 2-word-phrases as a key in the search index, our method tries to achieve the same precision at ranks 10, 5 and 1 as obtained by the conventional BoW-based approach with the GV post-processing. Matching spatially and geometrically constrained 2-word-phrases partially incorporates the advantages of GV. In Figure 6, we can observe that while the raw individual visual word matches include a lot of false-positives, on matching 2-word-phrases, we mostly have true-positive matches.

We also observe that the 2-word-phrases help in significantly reducing the size of the search index. This can be realised if we understand that only stable feature keypoints are preserved and indexed in the search records. Lone keypoints that cannot be paired with others in the chosen neighborhood, as well as those set of pairs that occur in single

Figure 6. Visual word vs. 2-word-phrases matches between two images of the same object captured from very distinct view points. While word matches require a geometric verification to remove false positive matches, most of the matched 2-word-phrases are true. A few two-word-phrases-matches are highlighted with same-colored ellipses around the keypoint pairs.

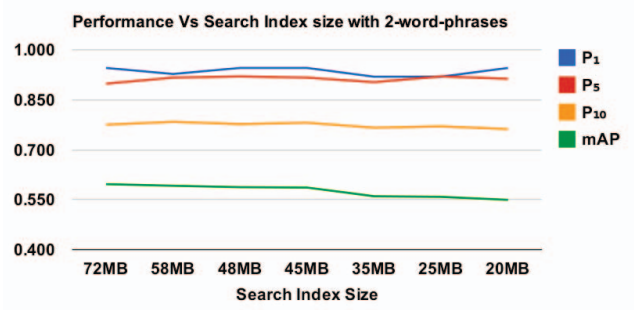

Figure 7. Graph of Performance Vs Index size with the 2-word-phrases indexing. The mAP decreases with size of the search index. However, Precision at 10, 5, 1 remain largely unaffected.

images, are rejected. This results in pruning of the original vocabulary. As the vocabulary size decreases, the memory footprint for the search index also decreases. This affects the mAP. However, we observe that the Precision at $10, 5, 1$ remain mostly unaffected (See Figure 7). This is because, while the ranked-list is affected with smaller search index, the top results are in general, preserved.

4.3. Mobile App

We now discuss an implementation for a mobile instance search on a mid-end mobile phone. Our application provides details of the monuments based on the image captured on a mobile phone. The app is implemented for Android devices and designed to allow a tourist point his mobile at an object and know its details without using a network connection. Examples of the recognition is shown in Figure 8.

The app is designed for the heritage data set discussed in the section 4.1. The search index is built on a dataset with 50,292 medium resolution (480×360) images, which amounts to $5.3GB$. Note that the images are never stored on the mobile devices. We extract SIFT features from all images and use hierarchical k-means to build a corresponding visual vocabulary of size $100K$. The memory occupied by the vocabulary tree during query processing is $17MB$. With our 2-word-phrase indexing technique, we get a compact search index of $25MB$. These along with the annotations for the database images are all that we store as application data on the mobile device.

The heritage site is marked by architecturally similar monumental structures and towers spread over a large area. We did real-time testing of our app at this site and analyzed how our solution performs in terms of correctness of retrieved annotations, memory usage and the speed of annotation delivery. The app successfully returned correct annotations at a wide-variety of locations and for different monuments. On average, the annotation was delivered within a second or two (See Table 4.3). Since, this is a tourist location, we did face failure cases, when a large part of the monument was occluded by crowd, however, overall we found the app to be successful at its task.

Event	**Time**(in seconds)
SIFT keypoint Detection	0.250
Extracting SIFT descriptors	0.270
Quantizing to Vocabulary	0.010
Compute 2-word-phrases	0.470
Index Search	0.080
Total	1.080

Table 5. Average computational time analysis on a mobile phone with 1GHz processor. Feature extraction and constructing 2-word-phrases take up significant time.

5. Conclusion

This paper presents an indexing strategy for reducing the memory footprint for instance retrieval along with exploiting the spatial geometry of 2-word-phrases. Precision at K is preserved even with significant reduction in the size of search index. We successfully demonstrate the usage of 2-word-phrases as key in the inverted search index and how this helps in pruning the vocabulary as well as compacting the search index. This facilitates successful application in low-memory, slow-processing mobile environments.

Acknowledgements

This research is carried out as part of Indian Digital Heritage Project of DST, Govt. of India. It is also carried out in part at the SeSaMe Centre, supported by the Singapore NRF under its IRC@SG Funding Initiative and administered by the IDMPO.

References

[1] ABIresearch http://www.abiresearch.com/press/average-size-of-mobile-games-for-ios-increased-by-. accessed: 2-April-2012. 1

[2] R. Arandjelović and A. Zisserman. All about VLAD. In *CVPR*, 2013. 4

[3] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang. Spatial-bag-of-features. In *CVPR*, 2010. 3

[4] V. Chandrasekhar, D. M. Chen, Z. Li, G. Takacs, S. S. Tsai, R. Grzeszczuk, and B. Girod. Low-rate image retrieval with tree histogram coding. In *MobiMedia*, 2009. 2

[5] V. Chandrasekhar, Y. Reznik, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod. Quantization schemes for low bitrate compressed histogram of gradients descriptors. In *CVPR Workshops*, 2010. 2

[6] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, Y. Reznik, R. Grzeszczuk, and B. Girod. Compressed histogram of gradients:a low-bitrate descriptor. *IJCV*, 2012. 2

[7] D. M. Chen, S. S. Tsai, V. Chandrasekhar, G. Takacs, J. P. Singh, and B. Girod. Tree histogram coding for mobile image matching. In *DCC*, 2009. 2

[8] O. Chum, M. Perdoch, and J. Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. In *CVPR*, 2009. 2, 3

Figure 8. An application of the compact search index structure for tourism. Mobile phone localizes and gives details of what it sees without the help of external network. The first row shows usage at different locations and monuments; the second row demonstrates the robustness of the approach to view-changes of a particular monumental structure.

[9] O. Chum, J. Philbin, and A. Zisserman. Near duplicate image detection: min-hash and tf-idf weighting. In *BMVC*, 2008. 2

[10] J. Feng. Mobile product search with bag of hash bits and boundary reranking. In *CVPR*, 2012. 2

[11] P. Föckler, T. Zeidler, B. Brombach, E. Bruns, and O. Bimber. Phoneguide: museum guidance supported by on-device object recognition on mobile phones. In *MUM*, 2005. 2

[12] B. Girod, V. Chandrasekhar, D. M. Chen, N. man Cheung, R. Grzeszczuk, Y. Reznik, S. Tsai, G. Takacs, and R. Vedantham. Mobile visual search. *IEEE SPM*, 2011. 2

[13] N. Henze, T. Schinke, and S. Boll. What is that? object recognition from natural features on a mobile phone. In *MIRW*, 2009. 2

[14] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008. 2

[15] H. Jégou, M. Douze, and C. Schmid. Packing bag-of-features. In *ICCV*, 2009. 2

[16] H. Jegou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010. 2

[17] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *PAMI*, 2012. 2, 4

[18] R. Ji, L.-Y. Duan, J. Chen, H. Yao, Y. Rui, S.-F. Chang, and W. Gao. Towards low bit rate mobile visual search with multiple-channel coding. In *ACM MM*, 2011. 2

[19] Y. Jiang, J. Meng, and J. Yuan. Randomized visual phrases for object search. In *CVPR*, 2012. 2

[20] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006. 2, 3, 6

[21] M. Perd'och, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, 2009. 2

[22] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, 2010. 2

[23] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 2, 3

[24] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008. 3, 5

[25] S. Romberg, M. August, C. X. Ries, and R. Lienhart. Robust feature bundling. In *Advances in MIP*, PCM, 2012. 2, 3

[26] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 2

[27] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W.-C. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod. Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In *ACM MIR*, 2008. 2

[28] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *CVPR*, 2008. 2

[29] P. Turcot and D. G. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems, 2010. 2, 5

[30] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. In *ISMAR*, 2008. 2

[31] Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling features for large scale partial-duplicate web image search. In *CVPR*, 2009. 3

[32] X. Zhang, Z. Li, L. Zhang, W. Ma, and H.-Y. Shum. Efficient indexing for large scale visual search. In *ICCV*, 2009. 2

[33] Y. Zhang and T. Chen. Efficient kernels for identifying unbounded-order spatial features. In *CVPR*, 2009. 4

[34] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR*, 2011. 2

[35] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian. Spatial coding for large scale partial-duplicate web image search. In *ACM MM*, 2010. 2