Combining Dense Features with Interest Regions for Efficient Part-based Image Matching

Priyadarshi Bhattacharya and Marina L. Gavrilova
Dept. of Computer Science, University of Calgary, 2500 University Drive, NW, Calgary, Canada

Keywords: Recognition, Part-based Match, Dense Sampling, Interest Regions.

Abstract: One of the most popular approaches for object recognition is bag-of-words which represents an image as a histogram of the frequency of occurrence of visual words. But it has some disadvantages. Besides requiring computationally expensive geometric verification to compensate for the lack of spatial information in the representation, it is particularly unsuitable for sub-image retrieval problems because any noise, background clutter or other objects in vicinity influence the histogram representation. In our previous work, we addressed this issue by developing a novel part-based image matching framework that utilizes spatial layout of dense features within interest regions to vastly improve recognition rates for landmarks. In this paper, we improve upon the previously published recognition results by more than 12% and achieve significant reductions in computation time. A region of interest (ROI) selection strategy is proposed along with a new voting mechanism for ROIs. Also, inverse document frequency weighting is introduced in our image matching framework for both ROIs and dense features inside the ROIs. We provide experimental results for various vocabulary sizes on the benchmark Oxford 5K and INRIA Holidays datasets.

1 INTRODUCTION

Object recognition is perhaps one of the most important areas of computer vision. It remains a highly challenging problem because of the large variations in scale, viewpoint and illumination between two instances of the same object. Occlusion and background clutter can also significantly hinder reliable recognition. The bag-of-words (BoW) image representation, although highly popular for object recognition, has some disadvantages. It is a global representation of an image and thus, is unsuitable for sub-image retrieval problems where the query object may occupy only a small part of the corpus image. Any noise or background clutter invariably influences the histogram representation and as a result the similarity computation between two images. The other disadvantage is the lack of spatial information of visual words in the representation. This results in many false matches which need to be rectified by performing geometric verification. But this process is computationally expensive and can only be applied to a limited number of images. Many genuinely similar images may not be make it to the shortlist for geometric verification.

In (Bhattacharya and Gavrilova, 2013), we have proposed a novel image matching framework based on matching interesting regions in an image individually instead of matching entire images at a time based on histogram approach. Our motivation is that even if two images having the same query object do not match well in entirety, a high level of similarity between certain regions in the images can still be expected. Figure 1 illustrates this with a sample query and corpus image from Oxford 5K dataset (Philbin et al., 2007). To this end, in (Bhattacharya and Gavrilova, 2013), we propose a novel method for combining dense features with interest points. We compute interest points in scale-space over a restricted range of sigma. We also compute dense features at a fixed spatial stride and a number of scales. For each interest point, we consider only those dense features that are contained inside and have a radius that is approximately half the radius of the interest point. This strategy has the advantage of achieving invariance to scale while at the same time not requiring dense features to be computed at multiple scales which will potentially greatly increase the number of dense features to be considered per interest point. Any low contrast dense features are discarded. In addition, the number of dense features inside an interest point is limited to a maximum of 49 by using a uni-

Bhattacharya P. and Gavrilova M..
Combining Dense Features with Interest Regions for Efficient Part-based Image Matching.
DOI: 10.5220/0004684000680075
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (VISAPP-2014), pages 68-75
ISBN: 978-989-758-004-8

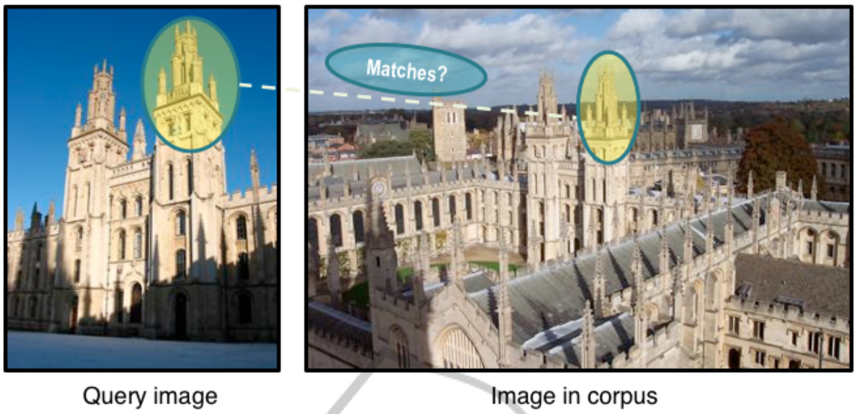


Figure 1: Part-based image matching (landmark images from Oxford 5K (Philbin et al., 2007)).

form sampling strategy.

In this paper, we introduce improvements to the feature selection stage. We utilize Harris-Laplace (Mikolajczyk and Schmid, 2004) in place of Laplacian of Gaussian (LoG) to detect interest points as it localizes better and eliminates repetitive detections along edges. We also experimented with Hessian-Laplace (Mikolajczyk and Schmid, 2004) but Harris-Laplace produced better results. From the interest points which can have significant overlap and number in thousands, we select a maximum of 200 regions of interest (ROIs). The selection strategy is detailed in the methodology section in this paper. We only consider dense features that are contained inside these ROIs and have a radius equal to half the radius of the ROI. This was found to exhibit superior recognition performance in (Bhattacharya and Gavrilova, 2013). In contrast to (Bhattacharya and Gavrilova, 2013) which uses two separate vocabularies - one built from the interest points and another from the dense features, we utilize a single vocabulary built from the dense features. This reduces quantization error and improves retrieval quality. Using a voting mechanism that requires the dense features to vote for the ROIs in which they are contained using an inverted index structure, we are able to quickly determine which images in corpus have ROIs that share similar dense features to a query image ROI. The spatial arrangement of dense features is considered in estimating a match score. A cumulative score is computed by summing up the match scores for all ROIs in query image and the corpus images are sorted based on descending order of this score. RootSIFT introduced in (Arandjelović and Zisserman, 2012) and inverse document frequency weighting are used to further improve recognition results. With these improvements, the highest mAP reported on 200K vocabulary size in (Bhattacharya and Gavrilova, 2013) jumps by 12.8% while using only up to 200 ROIs per image (80% less). It also reduces query time per image by more than 50%.

2 RELATED WORK

Several improvements over bag-of-words have been proposed over the years. Increasing the vocabulary size helps compensate for some of the quantization error and has been used to improve recognition accuracy (Mikulk et al., 2010)(Philbin et al., 2007). But building a vocabulary approaching a million or more visual words is prohibitively time consuming and the storage requirements also increase. Soft quantization (Philbin et al., 2008) has been suggested to improve recognition accuracy but can significantly increase query time and storage requirements as each word is mapped to *k* nearest visual words. These techniques do not effectively address the two weaknesses of bag of words - lack of spatial information and susceptibility to noise and background clutter.

Spatial information is considered in (Cao et al., 2010) by considering several ordered bag-of-words per image. But it requires tuning of parameters and an offline boosting-based method to learn the most effective features and is most effective for very large vocabularies only.

(Lin and Brandt, 2010) uses a family of histograms that depend functionally on a bounding rectangle. The method uses more spatial information than bag-of-words but is considerably more complicated than bag-of-words and the grid size selection can be problematic.

Instead of matching each feature to nearest visual word, (Jegou et al., 2010) computes a signature for each feature and maps to a visual word only if the hamming distance between the signatures is higher than a threshold. This achieves higher recognition rates than bag-of-words for reasonable vocabulary sizes but in order to obtain the binary signature, an offline training process is required.

Several variations of query expansion have been proposed in literature (Arandjelović and Zisserman, 2012)(Chum et al., 2007) but as observed in (Jegou et al., 2010), this significantly increases query time by several orders of magnitude and only works well when there are several images of the same object in corpus.

(Wu et al., 2009) proposed grouping of SIFT features inside Maximally Stable Extremal Regions (MSER) and applied to logo and web image retrieval. This does not work well for real-world photographs with large changes in viewpoint and illumination and is mostly applicable for 2D images.

Our method is able to consistently produce significantly better recognition rates than bag-of-words for a wide range of vocabulary sizes and incurs no offline training or learning overheads. Experiments reveal that the proposed approach can be effective for recognition in real-world photographs involving large changes in viewpoint and occlusion and sub-image retrieval problems. Also, the retrieval is several orders of magnitude faster than performing geometric verification or query expansion.

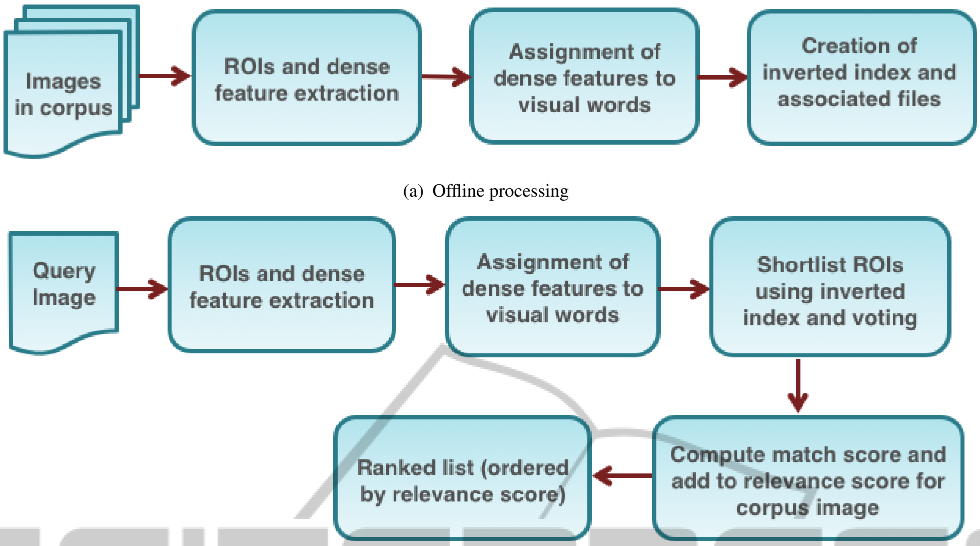
3 METHODOLOGY

Figure 2 provides an overview of proposed method. The offline processing module (subfigure 2(a)) prepares the inverted index and files that store spatial information from the images in corpus. It also computes the inverse document frequency weights for visual words inside the ROIs. The online processing module (subfigure 2(b)) inputs the query image and computes the regions of interest (ROIs) and dense features. It assigns dense features to visual words using the codebook computed from the dense features in the corpus images. A voting mechanism is used to determine ROIs in corpus that share common visual words. An array whose size equals the number of ROIs in corpus is first initialized to zero. Using an inverted index structure, each visual word in a ROI in the query image votes for ROIs in corpus in which it occurs. Figure 3 illustrates the voting mechanism. After the voting, the counts in the array represent the number of visual words each corpus ROI has in common with the query ROI. For corpus ROIs that have count < 2, the match score is set to zero. For the rest, a match score is computed based on the number of visual words in common and agreement in their spatial layout. The match score is then added back to a second array that stores the cumulative match score for all the corpus images. The corpus images are then ordered based on descending order of the cumulative match score. For all our experiments, we use RootSIFT (Arandjelović and Zisserman, 2012) instead of SIFT. With our matching framework, this is observed to increase mAP by about $1\% - 2\%$ from using L2 distance for SIFT comparison.

3.1 ROI Computation

In this section, we provide details of the feature extraction. We compute Harris-Laplace interest points using the LIP-VIREO toolkit of (Zhao, 2010). In contrast to our previous work (Bhattacharya and Gavrilova, 2013) which detects interest points using LoG and also computes the descriptors, we simply compute the interest points using Harris-Laplace. We discard overly large and overly small interest points as these are likely to result in false matches. Specifically, for all our experiments, we discard any interest points with radius < 15 or > 51 pixels. The number of interest points still number in the thousands. We sort the interest points in descending order of radius. Using a kd-tree, we efficiently determine the nearest interest points to any given interest point. We discard any nearby interest points for which the distance between the interest point centres is $< \triangle D$ and difference in radius is $< \triangle R$. The motivation is to discard overlapping interest points that are similar in scale and hence likely to represent similar image structure. If the number of ROIs is > 200, we sort the ROIs based on saliency in descending order and select up to the top 200 ROIs. Subfigure 4(a) displays the interest points detected using Harris-Laplace while subfigure 4(b) displays the ROIs extracted using the simple technique just outlined. We set $\triangle D$ and $\triangle R$ to 20 pixels for all our experiments. It is important to note that if the value of $\triangle D$ is set too high, then the localization accuracy of the ROIs will not be good, resulting in degradation in recognition performance.

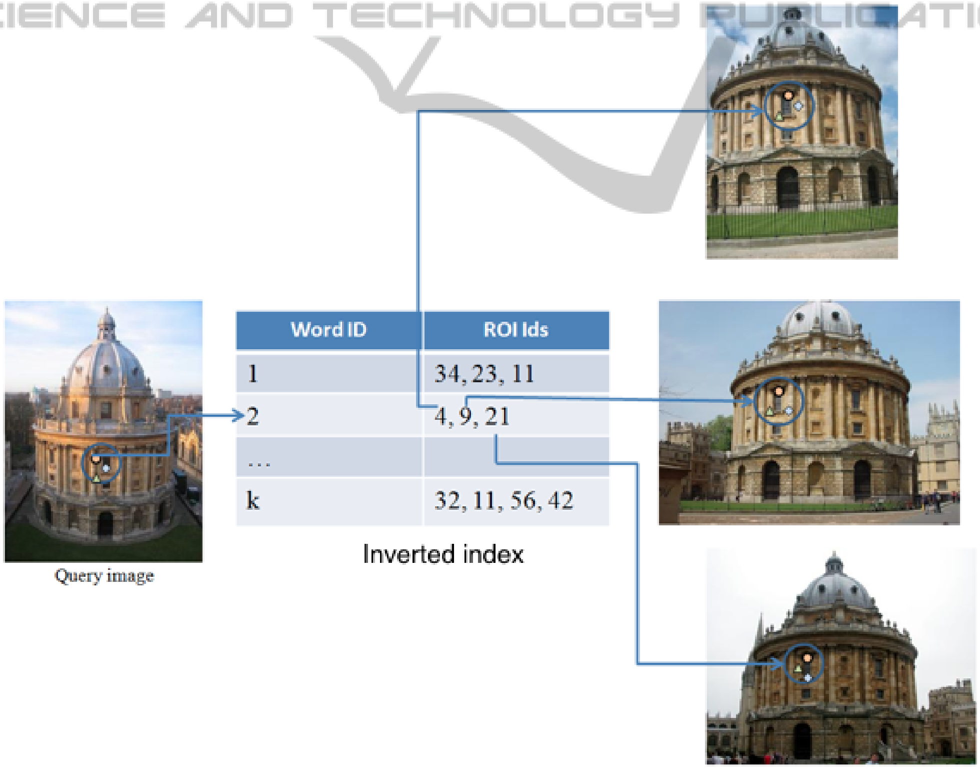
We next compute dense features at a spatial stride of 5 pixels and 6 scales $\{9,12,15,18,21,24\}$ using the *vl_phow* command of (Vedaldi and Fulkerson, 2012). This has been used for category recognition in (Chatfield et al., 2011). It is fast to compute and takes well below a second per image. For ROIs with radii in the range $[15,21]$, we only consider dense features that are contained inside and have a radius of 9 pixels (roughly half the ROI radius). Similarly, for ROIs with radii in the range $\{[22,27],[28,33],[34,39],[40,45],[46,51]\}$, we consider dense features that have a radius of $\{12,15,18,21,24\}$ pixels respectively. Since the ra-


Images in corpus
ROIs and dense feature extraction
Assignment of dense features to visual words
Creation of inverted index and associated files
Query Image
ROIs and dense feature extraction
Assignment of dense features to visual words
Shortlist ROIs using inverted index and voting
Compute match score and add to relevance score for corpus image
Ranked list (ordered by relevance score)

(a) Offline processing

(b) Online processing

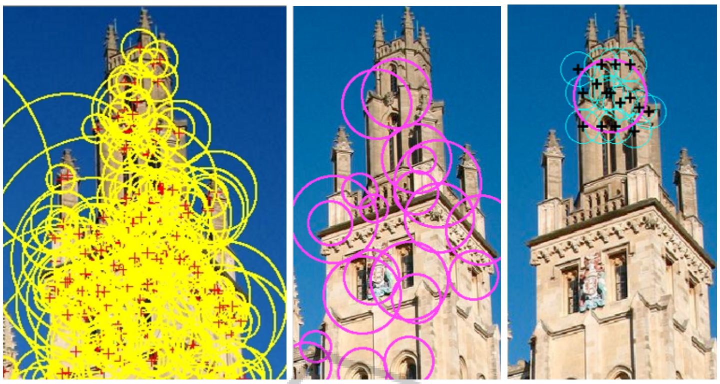
Figure 2: Flowchart for proposed method.



Word ID	ROI Ids
1	34, 23, 11
2	4, 9, 21
...	
k	32, 11, 56, 42

Figure 3: Visual words representing dense features inside an ROI voting for ROIs in corpus (landmark images from Oxford 5K (Philbin et al., 2007)). Blue circles represent the ROIs and the shapes inside represent dense features mapping to various visual words.

dius of dense features considered inside a ROI is based on the radius of the ROI, which is derived in scale-space, the dense features are scale-invariant without requiring to compute them at multiple fixed scales as prevalent in category recognition. This limits the number of dense features to be considered per ROI. We have observed through experiments that the recognition quality is not impacted much if the num-



(a) Harris-Laplace interest points (b) Computed ROIs (c) Dense features inside one of the ROIs

Figure 4: ROIs and dense features (landmark image from Oxford 5K (Philbin et al., 2007)).

ber of scales is ≥ 5.

3.2 Inverted Index and other Files

Codebook is built by clustering the dense features inside the corpus ROIs. We use the approximate K-means of (Philbin et al., 2007) for clustering and consider up to 16 million dense descriptors sampled uniformly. The dense features are then mapped to the codebook. All files generated are in binary format for fast random access. The inverted index file contains an entry for each word in the codebook: the number of ROIs in corpus in which it occurs and then the ROI ids in sequence (each ROI is assigned a unique id). Since a record in the inverted index can be of variable length, we maintain an index file on the inverted index which simply stores the offset of each word in the inverted index file. There is a separate file created for storing information about the ROIs. It contains two entries per ROI - the offset in a binary file where the visual words inside the ROI are stored and another offset to a second binary file which stores the x and y coordinates of the words. A binary file is also created which stores the inverse document frequency weights of each word. It is defined as $log(N/n_i)$ where N is the number of images in corpus and n_i is the number of images in which the word occurs. This penalizes words that occur in many images and are hence less informative. We additionally create a stop list of words. We count the number of corpus ROIs in which each word occurs and compute the maximum. Any word that occurs more than 50% of the maximum is assigned to the stop list.

3.3 Querying

Given a query image, the ROIs and dense features are extracted in a similar manner as previously described for corpus images. The dense features are then mapped to the codebook to obtain the representative visual words. We iterate over all the ROIs in the image and for each ROI, we use the inverted index file to determine which are the corpus ROIs that share visual words. If the number of common visual words is < 2, we set ROI match score to zero. If the number is ≥ 2, we compute the match score S_i for i^{th} ROI in query image based on formula introduced by us in (Bhattacharya and Gavrilova, 2013) with some modifications:

$$S_i = T \times log\left(\frac{CP_{max}}{CP}\right) \tag{1}$$

where T is the sum of inverse document frequency weights of the common visual words between the two ROIs being matched, CP is the cumulative penalty for the sequence, CP_{max} is the maximum CP. The computation of CP and CP_{max} can be found in (Bhattacharya and Gavrilova, 2013). We use the second technique - *match score using relative order* which does not require a descriptor computation for the ROIs. The cumulative score (CS) for a corpus image is then computed over all query ROIs as:

$$CS = \sum_{i=1}^{m} S_i \, * \, P_i \tag{2}$$

where $P_i = log(N/n_i)$

Here N is the number of images in corpus and n_i is the number of images to which the i^{th} ROI contributes a match score > 0 (this information is obtained as part of the earlier match score computation) and m is the number of ROIs in query image. It penalizes ROIs that yield matches with many images in corpus.

Table 1: Performance (mAP) for different vocabulary sizes on Oxford 5K dataset.

Vocab. size	BoW	SBOF (Cao et al., 2010)	HE + WGC + wts. + MA (Jegou et al., 2010)	Proposed
10K	0.358	–	–	0.546
20K	0.385	–	0.605	0.568
50K	0.473	0.523	–	0.592
100K	0.534	0.571	–	0.620
200K	0.561	–	0.615	0.641

4 EXPERIMENTS

We experimented with two popular benchmark datasets for image retrieval - the Oxford 5K (Philbin et al., 2007) and INRIA Holidays (Jegou et al., 2008). The Oxford 5K dataset has a total of 5062 images collected from Flickr of 11 different landmarks in Oxford area and a large portion of distractor images. Mean Average Precision (mAP) (as defined in (Philbin et al., 2007)) is used for evaluation. There are 55 queries in total (5 for each landmark) with bounding boxes of query regions given. The INRIA Holidays dataset has 1491 images in total divided into 500 image groups of various natural and man-made scenes. The first image of each group is the query and correct retrieval results are the other images of that group. The evaluation is again based on Mean Average Precision and we use the evaluation tools available online from the authors of (Jegou et al., 2008). Since our method uses new kind of features instead of standard SIFT, we compute all features ourselves. The vocabularies are computed from the dense features in each dataset.

4.1 Results on Oxford 5K Dataset

Table 1 presents the retrieval results for Oxford 5K dataset for various vocabulary sizes. The improvement over bag-of-words (BoW) is quite significant. For smaller vocabularies, the improvement percentage is more pronounced. Our method exhibits improvement in recognition accuracy with increase in vocabulary size similar to bag-of-words and it is expected that with larger vocabulary sizes, the retrieval accuracy will improve further. Although (Jegou et al., 2010) achieves a higher mAP for 20K vocabulary compared to our method, we obtain superior results for 200K vocabulary size. The best accuracy using Local BoF mentioned in (Lin and Brandt, 2010) for Oxford 5K is 0.647 but the authors do not specify the vocabulary size they use.

4.2 Results on INRIA Holidays Dataset

For 20K vocabulary, the mAP achieved by our method is 0.638 and for 200K vocabulary, the mAP is 0.685. This is significantly higher than the bag-of-words mAP of 0.469 and 0.572 (from (Jegou et al., 2010)) for 20K and 200K sized vocabularies. We expect the mAP of our method to increase further with larger vocabulary sizes. Our mAP is lower than the best reported mAP of 0.75 in (Jegou et al., 2008) but the method in (Jegou et al., 2008) requires offline learning and uses priori knowledge of image orientation. (Perdoch et al., 2009) achieved a mAP of 0.715 on this dataset using a one million vocabulary. This method requires learning of geometry representations.

4.3 Analysis of Algorithm Parameters

In order to analyze the impact of parameter selection, we created a subset of Oxford 5K (Philbin et al., 2007) which consists of only the *Good* images for each of the landmarks. We remove *Ok* images from consideration in the evaluation similar to *Junk* images. The query images are the same as in the original dataset. We first analyze the impact of maximum number of ROIs on the mAP. Figure 5 plots the mAP for 4 different choices of the maximum number of ROIs. With more ROIs, there is greater coverage of the image and we observe a steady increase in mAP as the maximum number of ROIs per image increases from 100 to 200. From 200 to 250, the increase is less dramatic and for higher values (not shown in figure), it is even less. Figure 6 plots the total query time for all 55 queries. As can be expected, the query time increases as more ROIs are considered per image. There is clearly a trade-off here between query time and mAP. We set the maximum number of ROIs to 200 for all our experiments as it seems to be a good fit both with respect to retrieval quality and query time. If query time is more important however, setting ROIs

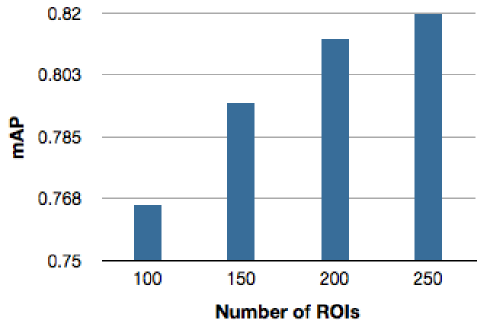


Figure 5: mAP versus Number of ROIs.

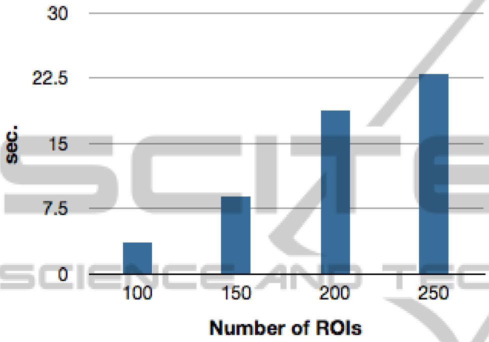


Figure 6: Total query time (for 55 queries) versus Number of ROIs.

Table 2: Performance (mAP) for different values of $\triangle D$ ($\triangle R = 20$).

$\triangle D$	10	30	50	70	100
mAP	0.826	0.832	0.821	0.814	0.800

Table 3: Performance (mAP) for different values of $\triangle R$ ($\triangle D = 20$).

$\triangle R$	10	30	50	70	100
mAP	0.828	0.840	0.844	0.844	0.844

to 150 will be beneficial with a slight reduction in retrieval quality.

With the maximum number of ROIs set to 200, we analyze the impact of $\triangle D$ and $\triangle R$ on the mAP. From Table 2, it can be observed that a value of $\triangle D$ between 10 and 50 pixels yields similar results. With a higher value of $\triangle D$, less ROIs are detected per image which can reduce query time. As $\triangle D$ increases more, the retrieval quality deteriorates with increasing inaccuracy in ROI localization. As evident from Table 3, $\triangle R$ has less of an impact on the retrieval quality. It should be set to a reasonably high value to make sure we do not select ROIs that represent similar visual elements. For all our experiments on Oxford 5K and INRIA Holidays, we set both $\triangle D$ and $\triangle R$ to 20 pixels. The mAP for ($\triangle D = 20$, $\triangle R = 20$) is 0.836 for this reduced dataset.

4.4 Query Time and Storage Costs

We performed our experiments on a Macbook Pro laptop with 2.4 GHz Intel Core i5 processor and 16 GB RAM. The query time averaged over 55 queries for Oxford 5K and 500 queries for INRIA Holidays is shown in Table 4 for a 200K vocabulary. With increase in vocabulary size, the number of ROIs in which a visual word occurs will be less. Since match score is computed for only those ROIs which have at least 2 visual words in common, the number of ROIs for which match score needs to be computed reduces significantly with larger vocabularies. This result in faster queries. The query times are significantly lower than performing geometric verification or query expansion to improve bag-of-words retrieval results.

Table 4: Average query time for 200K vocabulary.

Dataset	Oxford 5K	INRIA Holidays
Avg. time (sec.)	0.62	0.18

The storage costs are dependent on the number of ROIs detected per image and the number of dense features inside each ROI. Table 5 shows the number of ROIs and dense features computed for the two datasets. With the current settings, the memory footprint per image is approximately 80 KB. This implies a corpus size of about 100K images can be handled in 8 GB memory for querying. For more images, we prefer a distributed computing/cloud based approach where the image corpus is distributed across N number of machines. Each machine retrieves up to m similar images and then the results are merged in time linear on the size of the returned results.

Since our algorithm returns a match score, images can be easily sorted across machines based on that score. This is cheaper than renting a high configuration machine on the cloud with very large RAM. Also, it will be faster as the machines will work in parallel.

5 CONCLUSIONS

In this paper, we developed a part-based image matching framework that combines dense features with interest regions in a novel way to dramatically improve image retrieval quality over bag-of-words. The method proposed by us is simple and easy to implement and does not require any offline learning or training overheads. We make new contributions in the feature extraction and image similarity computation stages of the recognition pipeline. Future research will involve looking into approaches for further re-

Table 5: Number of ROIs and dense features.

Dataset	*ROIs (total)*	*ROIs (avg. per image)*	*Dense (total)*	*Dense (avg. per ROI)*
Oxford 5K	932975	184	33788806	36
INRIA Holidays	259726	174	9534068	36

ducing the number of ROIs to be considered per image by possibly doing foreground-background separation or by eliminating ROIs completely from consideration that lead to too many false matches. Since dense features are not rotation invariant, possible rotation of the image area before computation can be investigated to further improve recognition quality.

ACKNOWLEDGEMENTS

We are thankful to Natural Sciences and Engineering Research Council of Canada and Alberta Innovates Technology Futures for continued support of this research.

REFERENCES

Arandjelović, R. and Zisserman, A. (2012). Three things everyone should know to improve object retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Bhattacharya, P. and Gavrilova, M. L. (2013). Spatial consistency of dense features within interest regions for efficient landmark recognition. *The Visual Computer*, 29(6-8):491–499.

Cao, Y., Wang, C., Li, Z., Zhang, L., and Zhang, L. (2010). Spatial-bag-of-features. In *CVPR*, pages 3352–3359.

Chatfield, K., Lempitsky, V., Vedaldi, A., and Zisserman, A. (2011). The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*.

Chum, O., Philbin, J., Sivic, J., Isard, M., and Zisserman, A. (2007). Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. In *ICCV*, pages 1–8.

Jegou, H., Douze, M., and Schmid, C. (2008). Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, pages 304–317.

Jegou, H., Douze, M., and Schmid, C. (2010). Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, 87(3):316–336.

Lin, Z. and Brandt, J. (2010). A local bag-of-features model for large-scale object retrieval. In *European conference on Computer vision*.

Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.

Mikulk, A., Perdoch, M., Chum, O., and Matas, J. (2010). Learning a fine vocabulary. In *European Conference on Computer Vision*, volume 6313 of *Lecture Notes in Computer Science*, pages 1–14. Springer.

Perdoch, M., Chum, O., and Matas, J. (2009). Efficient representation of local geometry for large scale object retrieval. In *CVPR*, pages 9–16.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *CVPR*.

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

Vedaldi, A. and Fulkerson, B. (2012). VLFeat: An open and portable library of computer vision algorithms. Available at http://www.vlfeat.org/.

Wu, Z., Ke, Q., Isard, M., and Sun, J. (2009). Bundling features for large scale partial-duplicate web image search. In *CVPR*, pages 25–32.

Zhao, W. (2010). LIP-VIREO: Local interest point extraction toolkit. Available at http://www.cs.cityu.edu.hk/∼wzhao2/.