A Local Bag-of-Features Model for Large-Scale Object Retrieval

Zhe Lin and Jonathan Brandt

Adobe Systems, Inc.
{zlin,jbrandt}@adobe.com

Abstract. The so-called bag-of-features (BoF) representation for images is by now well-established in the context of large scale image and video retrieval. The BoF framework typically ranks database image according to a metric on the global histograms of the query and database images, respectively. Ranking based on global histograms has the advantage of being scalable with respect to the number of database images, but at the cost of reduced retrieval precision when the object of interest is small. Additionally, computationally intensive post-processing (such as RANSAC) is typically required to locate the object of interest in the retrieved images. To address these shortcomings, we propose a generalization of the global BoF framework to support scalable local matching. Specifically, we propose an efficient and accurate algorithm to accomplish local histogram matching and object localization simultaneously. The generalization is to represent each database image as a family of histograms that depend functionally on a bounding rectangle. Integral with the image retrieval process, we identify bounding rectangles whose histograms optimize query relevance, and rank the images accordingly. Through this localization scheme, we impose a weak spatial consistency constraint with low computational overhead. We validate our approach on two public image retrieval benchmarks: the University of Kentucky data set and the Oxford Building data set. Experiments show that our approach significantly improves on BoF-based retrieval, without requiring computationally expensive post-processing.

1 Introduction

We address the problem of retrieving images containing an object of interest, specified by a visual query, from a large image database. We are interested not only in ranking the database images but also locating the relevant objects in the top matching images.

Perhaps the most common and effective approach to large-scale image retrieval is the bag-of-features (BoF) framework (see, for example [15, 11, 13, 2]). The BoF representation for an image is a global histogram of visual word occurrences where each "visual word" is a quantized local feature descriptor. The set of all possible visual words, or visual vocabulary, is learnt via various clustering algorithms, such as k-means [15, 2], hierarchical k-means (HKM) [11], and approximate k-means (AKM) [13].

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 294–308, 2010.

Fig. 1. An example of small object retrieval. Left: a query image with a region of interest. Right: the top 8 retrieved images using our approach and the baseline Global BoF approach. Red rectangle represents the query object of interest, and Green rectangles represent the returned object bounding boxes using our approach. In contrast, the baseline method cannot return bounding boxes and need additional totally different criterion (*e.g.* RANSAC) to localize objects.

Large vocabularies, typically containing a million or more visual words, tend to be more discriminative therefore more effective in locating specific objects. The large vocabulary size results in sparse histograms for particular images which can be efficiently represented and searched using inverted files [15].

Since the BoF representation contains no spatial information, post-processing to verify the spatial consistency of the retrieved images tends to improve retrieval accuracy provided the underlying spatial model is appropriate. Approaches to spatial verification include spatial neighborhood counting [15], as well as RANSAC-based spatial matching [13].

Retrieval based on the global BoF representation, although being very scalable, has the shortcoming that objects become difficult to retrieve as the amount of surrounding clutter in an image increases. That is, small objects are hard to find using a global histogram representation. Post-processing based spatial verification partially addresses this, but with an added computational cost that effectively limits the total number of images to be considered for post-processing. We propose a generalization of the global BoF framework to support scalable local matching without post-processing. By *local matching*, we mean matching the query to a locally bounded BoF, as opposed to a global BoF, which limits the effect of clutter, and also localizes the object. The generalization is to represent each database image as a family of histograms that depend functionally on a bounding rectangle. Integral with the image retrieval process, we identify bounding rectangles whose consequent local histograms optimize query relevance, and rank the images accordingly.

Ideally, we aim to localize the best region (namely, the one that has the maximum similarity to the query) in all the database images. As a simplification, we constrain our problem to the set of all possible subrectangles. Each image is therefore represented as a BoF histogram parameterized by a subrectangle. We can certainly go beyond rectangles through a post refinement process as in [19].

In order to maintain scalability, we use a spatial quantization-based indexing mechanism to compute sparse feature energies (norms) offline, and compute similarities over a coarse grid of rectangles to the query online. Integral images, enabled by a binary approximation of the BoF model, allow the localized similarities to be computed efficiently, and a full BoF comparison is done for the final ranking. In this way, we are able to match a query BoF against a broad set of sub-rectangle BoFs for each of the database images.

An example of the effectiveness of our algorithm for small object retrieval is shown in Fig. 1, where our approach returns more consistent results than traditional global BoF methods, and can localize objects simultaneously.

2 Related Work

Most common approaches to object localization in the BoF retrieval framework include neighborhood counting [15], and RANSAC-based methods [5,13]. Neighborhood counting uses the total number of neighboring word correspondences to rerank images. It is largely dependent on the size of the neighborhood and cannot capture spatial relationship in wider configurations. The RANSAC-based approaches can capture wider spatial consistency but are typically limited to near planar objects in order to avoid an overly complex spatial model, and are applied only to top hundreds of images [13, 2] due to RANSAC's computation complexity. During re-ranking, RANSAC-based verification computes similarities as the number of inliers, which is very different from the ranking criterion used in the first-phase BoF retrieval process (*i.e.* BoF similarity).

There have been approaches grouping pairs of or multiple local features in a larger spatial neighborhood as a new 'feature', *e.g.* the geometric min-Hash [1], bundling features [17], and multi-samples [18], to increase feature discriminative power. These approaches capture visual word co-ocurrence information in an early stage of retrieval, but still need an additional post-processing to localize objects in the top retrieved images.

BoF matching can also be formulated as a voting framework [2], where each matching pair of features between query and database will generate a vote (score) to be accumulated to query-to-database image distances. A fast weak geometric consistency scheme is introduced in [2] by voting for rotation angles and log-scale ratios during the first-phase retrieval process, but this model does not provide localization capability and cannot be easily extended to localize objects in arbitrarily rotated images.

We propose a local BoF model to simultaneously rank images and localize relevant objects under arbitrary rotations, significantly different viewpoints, and in the presence of clutter. By localized BoF matching, the model encodes weak spatial constraints implicitly during the retrieval process to improve the ranking

accuracy and localize objects simultaneously. The model is fully integrated with an inverted file-based search to support large-scale object search and localization.

The localization process uses a simple greedy optimization method due to the potentially large scale nature of our problem. Although it is not guaranteed to find the global optimum, we have found through experiment that the retrieval accuracy of our approach is nearly identical to the result obtained by exhaustive search, as can be seen in Table 1. Also, our optimization method could be replaced with branch-and-bound [7] to guarantee a global optimum, but we found the added computational expense to be unnecessary.

Our approach is closely related to [6] in formulating the problem as a combination of image retrieval and object localization. Lampert [6] applied the branch-and-bound search to problems of subimage retrieval in large image and video sets, but the approach differs from our method in that it does not leverage the fast inverted file for localization. In this way our method is more scalable than [6].

In the context of a complete retrieval systems, our method can be regarded either as an improved BoF matching for the first-phase retrieval process [11], or as an improved weak spatial verification alternative to RANSAC-based schemes. Our contributions are three-fold:

1. A computationally efficient local BoF model for re-ranking database images and localizing objects in a large number images.
2. A local spatial pyramid model for combining the local BoF model and the spatial pyramid-based representation.
3. An efficient, integrated system for local BoF model and inverted file index for large scale object retrieval.

3 Local BoF Retrieval

3.1 Global BoF Model

The BoF representation begins with detection of local image features and extraction of each of the features as high dimensional descriptors $f \in \mathcal{R}^n$. Each of the descriptors is quantized according to a quantization function, $\mathcal{C} : \mathcal{R}^n \rightarrow \{1, 2, 3, \ldots, V\}$, to generate a set of "visual words" representing the image. The global BoF representation for an image is the normalized histogram of visual words, typically using either the L_1 or the L_2 norm, with components weighted by term frequency-inverse document frequency (TF-IDF). (Term frequency (TF) $\tau(i)$ is defined as the number of occurrences of word i in an image, and the inverse document frequency (IDF) α_i is defined as $\alpha_i = \log \frac{N}{N_i}$, where N is the total number of images and N_i is the number of images containing the word i.)

Let q denote the BoF for a query image and d denote the BoF for a database image. The relevance of d to query q is the distance, $D(q, d) = \|q - d\|_p^p$, where $p \in \{1, 2\}$ [11, 13, 2]. For search, the database images are ranked in ascending order of the distance to the query.

Since the BoF becomes very sparse when the vocabulary is large, the distance D can be evaluated efficiently by considering only the non-zero elements of q and d. For the L_2 norm, the simplification is as follows [15]:

$$D(q,d) = \|q-d\|_2^2 = 2 - 2 \sum_{i|q_i \neq 0 \wedge d_i \neq 0} q_i d_i. \quad (1)$$

We define the L_2 norm-based BoF similarity as:

$$S(q,d) := \sum_{i|q_i \neq 0 \wedge d_i \neq 0} q_i d_i. \quad (2)$$

In case of L_1 norm, the simplification is as follows (see [11] for the derivation):

$$D(q,d) = \|q-d\|_1^1 = 2 - \sum_{i|q_i \neq 0 \wedge d_i \neq 0} \left(q_i + d_i - |q_i - d_i|\right). \quad (3)$$

We define the L_1 norm-based BoF similarity as:

$$S(q,d) := \sum_{i|q_i \neq 0 \wedge d_i \neq 0} \left(q_i + d_i - |q_i - d_i|\right). \quad (4)$$

In any case, the search relevance of a database image I_d to a query image I_q is the BoF similarity, $\mathcal{S}(I_q, I_d) = S(q,d)$.

3.2 Local BoF Model

We can extend the global BoF model (denoted Global BoF) to a local model (Local BoF) by introducing a parameterization on the database BoF representation d. Specifically, let the Local BOF representation be a function $d(R)$ of a rectangle $R \in \mathbf{R}$, where R is parameterized by its bounding top/bottom/left/right image coordinates (t, b, l, r). $\mathbf{R}$ denotes the set of all subrectangles in an image. That is, for any database image, and for any subrectangle of the image, $d(R)$ is the normalized histogram of visual words occurring inside the subrectangle.

We define the image similarity as the global maximum of BoF similarity over the set of all possible subrectangles for the image.

$$\mathcal{S}(I_q, I_d) = \max_{R \in \mathbf{R}} S(q, d(R)), \quad (5)$$

$$R^*(I_d) = \arg\max_{R \in \mathbf{R}} S(q, d(R)), \quad (6)$$

where $S(q, d(R))$ is the localized object similarity, and $R^*(I_d)$ is the detected bounding box for image I_d. Note that R^* is not unique in general. We take a smallest one among the set of rectangles of equal similarity value.

We can solve the above problem by brute force simply by evaluating the similarity for all possible rectangles in all images as in the sliding window approach to object detection. We can also reduce the number of rectangles to consider by utilizing the branch-and-bound approach [7, 6]. However, by exploiting the sparsity of BoF vectors and the inverted file index storage representation, we can achieve the goal even more efficiently.

The approach is to fit the similarity equations (Eq. 2 and 4) into an integral image computation framework. Integral images have been widely used in the

object detection literature, *e.g.* Vedaldi *et al.* [16] used the integral image idea to improve the efficiency of object category detection significantly. Specifically, by converting from sum-over-word index to sum-over-feature form, and also factor the BoF normalization term out of the summation. We analyze L_1 and L_2 cases separately here. Let $\tilde{q}$ and $\tilde{d}$ denote the (*unnormalized*) TF-IDF weighted BoF histograms. Consequently, $q = \tilde{q}/\|\tilde{q}\|$ and $d = \tilde{d}/\|\tilde{d}\|$.

L_2 Case. Eq. 2 can be rewritten as follows:

$$S(q,d) = \sum_{i|\tilde{q}_i \neq 0 \wedge \tilde{d}_i \neq 0} \frac{\tilde{q}_i \tilde{d}_i}{\|\tilde{q}\|\|\tilde{d}\|} = \frac{1}{\|\tilde{q}\|\|\tilde{d}\|} \sum_{i|\tilde{q}_i \neq 0 \wedge \tilde{d}_i \neq 0} \tilde{q}_i \tilde{d}_i. \tag{7}$$

Similarly, the localized similarity $S(q, d(R))$ can be written as follows:

$$S(q,d(R)) = \frac{1}{\|\tilde{q}\|\|\tilde{d}(R)\|} \sum_{i|\tilde{q}_i \neq 0 \wedge \tilde{d}_i(R) \neq 0} \tilde{q}_i \tilde{d}_i(R). \tag{8}$$

Since $\|\tilde{q}\|$ is constant,

$$S(q,d(R)) \propto \frac{1}{\|\tilde{d}(R)\|} \sum_{f|\tilde{q}_{\mathcal{C}(f)} \neq 0 \wedge f \in R} \tilde{q}_{\mathcal{C}(f)} \alpha_{\mathcal{C}(f)}, \tag{9}$$

where f denotes a feature in image I_d, and $f \in R$ means the feature f is located inside the region R. α_i is the IDF weight for word i.

From Eq. 9, we can see that the similarity is represented as the sum over votes from individual feature points in database images. For an arbitrary subrectangle, it is now straightforward to use the inverted file to accumulate the summation term in Eq. 9 for non-zero query words, and use an integral image to rapidly evaluate the term for an arbitrary subrectangle. The integral image $\mathcal{G}_{q,d}(x,y)$ of the summation term for query q and image d can be written as:

$$\mathcal{G}_{q,d}(x,y) = \sum_{f|\tilde{q}_{\mathcal{C}(f)} \neq 0 \wedge f \in (0,y,0,x)} \tilde{q}_{\mathcal{C}(f)} \alpha_{\mathcal{C}(f)}. \tag{10}$$

Under the binary TF histogram assumption, $\mathcal{G}_{q,d}(x,y)$ simplifies to the form:

$$\mathcal{G}_{q,d}(x,y) = \sum_{f|\tilde{q}_{\mathcal{C}(f)} \neq 0 \wedge f \in (0,y,0,x)} \frac{\alpha^2_{\mathcal{C}(f)}}{\tau_d(\mathcal{C}(f))}, \tag{11}$$

where $\tau_d(\mathcal{C}(f))$ is the TF of word $\mathcal{C}(f)$ in I_d, which distributes the contribution of multiple features f corresponding to the same visual word uniformly so that to ensure the binary assumption of the global TF histogram of I_d.

But in order to evaluate the full similarity in Eq. 9 we need an approximation for $\|\tilde{d}(R)\|$ since the L_2 norm does not accumulate linearly. For very large vocabularies, the L_2 norm of a BoF vector can be approximated as the square root of

the L_1 norm [2]. (This follows from the observation that for large vocabularies, almost all TF histogram entries are either 1 or 0.) Hence, we replace the L_2 norm, $\|\tilde{d}(R)\|_2$, with the L_1 norm, $\|\tilde{d}(R)\|_1$, which can be computed efficiently for any subrectangle using an integral image. And, the integral image $\mathcal{H}_d(x, y)$ of $|\tilde{d}(R)|_1$ can be written as:

$$\mathcal{H}_d(x, y) = \sum_{f|f\in(0,y,0,x)} \frac{\alpha_{\mathcal{C}(f)}}{\tau_d(\mathcal{C}(f))}. \tag{12}$$

L_1 Case. Eq. 4 can be rewritten as follows:

$$S(q, d) = \sum_{i|\tilde{q}_i\neq 0\wedge\tilde{d}_i\neq 0} \left(\frac{\tilde{q}_i}{|\tilde{q}|} + \frac{\tilde{d}_i}{|\tilde{d}|} - \left| \frac{\tilde{q}_i}{|\tilde{q}|} - \frac{\tilde{d}_i}{|\tilde{d}|} \right| \right). \tag{13}$$

Similarly, the localized similarity $S(q, d(R))$ can be written as follows:

$$S(q, d(R)) = \sum_{i|\tilde{q}_i\neq 0\wedge\tilde{d}_i(R)\neq 0} \alpha_i \left(\frac{\tau_q(i)}{|\tilde{q}|} + \frac{\tau_{d(R)}(i)}{|\tilde{d}(R)|} - \left| \frac{\tau_q(i)}{|\tilde{q}|} - \frac{\tau_{d(R)}(i)}{|\tilde{d}(R)|} \right| \right), \tag{14}$$

where $\tau_q(i)$ and $\tau_{d(R)}(i)$ are the TFs of word i for q and $d(R)$, respectively, and α_i is the IDF weight.

We can again exploit the fact that for large vocabularies, most TF histogram entries are 0 or 1, and therefore we can approximate the BoF with its binary counterpart, where all non-zero entries are replaced by the IDF weights similar to the binary assumptions used in [4, 1]. Under this assumption, $\tau_q(i) = 1$ and $\tau_{d(R)}(i) = 1$ for all i such that $\tilde{q}_i \neq 0 \wedge \tilde{d}_i \neq 0$. Breaking Eq. 14 into two cases, $|\tilde{q}| >= |\tilde{d}(R)|$ and $|\tilde{q}| < |\tilde{d}(R)|$, will remove the absolute sign and the results of the two cases can be combined by using the max operator:

$$S(q, d(R)) = \frac{2}{\max(|\tilde{q}|, |\tilde{d}(R)|)} \sum_{i|\tilde{q}_i\neq 0\wedge\tilde{d}_i(R)\neq 0} \alpha_i, \tag{15}$$

or, dropping the constant,

$$S(q, d(R)) \propto \frac{1}{\max(|\tilde{q}|, |\tilde{d}(R)|)} \sum_{f|\tilde{q}_{\mathcal{C}(f)}\neq 0\wedge f\in R} \alpha_{\mathcal{C}(f)}. \tag{16}$$

The above simplification results in factoring out of norms and a summation over features f, which is exactly what needed for integral image-based framework. Specifically, the norm $|\tilde{q}|$ is fixed with respect to R, while $|\tilde{d}(R)|$ and the summation term can be computed efficiently for all R using the integral images. Similar to the L_2 case, the integral image $\mathcal{G}_{q,d}(x, y)$ of the summation term for query q and image d can be written as:

$$\mathcal{G}_{q,d}(x, y) = \sum_{f|\tilde{q}_{\mathcal{C}(f)}\neq 0\wedge f\in(0,y,0,x)} \frac{\alpha_{\mathcal{C}(f)}}{\tau_d(\mathcal{C}(f))}, \tag{17}$$

where τ_d denotes the TF histogram as in the L_2 case. The L_1 norm $|\tilde{d}(R)|$ is computed efficiently using the integral image $\mathcal{H}_d(x, y)$ in Eq. 12.

Although the binary TF histogram assumption does not take advantage of the full histogram information during the retrieval process, we can rerank the retrieved images according to their exact histograms based on Eq. 14, rather than the binarized approximation.

Another important detail in forming the integral images is to spatially distribute multiple instances of a particular word while not violating the binarization assumption. We have found that if we uniformly distribute the whole vote α_i to different instances, *i.e.*. if there are K instances of word i, each instance gets a vote of α_i/K, we do not introduce a spatial bias by arbitrarily selecting a particular word instance, while respecting the binarization assumption. This is accomplished by the presence of $\tau_d(\mathcal{C}(f))$ in Eq. 11, 12 and 17.

In contrast to generic object category detection where slight shifts and scalings of the window greatly affect the classification scores due to feature misalignment, in our problem, a coarse grid can be used without affecting accuracy. We have found that a 80×80 or 160×160 grid is sufficient for queries larger than 200×200 pixels. If the grid is 80×80, and the image size is 480×640, memory or storage requirement for 1 million images is only 96MB which is negligible compared to the size of the vocabulary and inverted file. In this case, the integral images $\mathcal{G}_{q,d}$ and $\mathcal{H}_d$ are defined on the grid, instead of at all pixels.

3.3 Optimization

Given integral images of norms and similarities between query and database images, we need an efficient optimization scheme for Eq. 5. Here, we simply use a greedy search (see Algorithm 1). In each iteration, we sequentially optimize individual coordinate in the order of (t, b, l, r), and stop the iteration process when the returned bounding rectangle in the current iteration is the same as in the previous iteration or the maximum iteration limit is reached. From experiments, we found that our approach finds global optima in about 66% of the cases and the process generally converges in less than 3 iterations as shown in Fig. 4.

3.4 Local BoF Algorithm

We follow the same general image retrieval framework as described, for example, in [15, 13, 11]. For training and indexing, we (1) extract local interest regions and descriptors for all database images, (2) construct the visual vocabulary by clustering, (3) quantize all descriptors into visual words, and (4) construct an inverted file, indexed on the visual words, and including feature geometry with the index. During the testing stage, we (1) extract interest regions and descriptors in query image, (2) compute distances (or similarities) between query and all database images using the inverted file, (3) apply our Local BoF-based search to localize and rerank the top K results.

The Local BoF retrieval algorithm is briefly described in Algorithm 2. We assume a feature quantizer is given and all features are indexed based on the

Algorithm 1. Greedy Query Localization: (n_x, n_y) is the grid width and height. M is the maximum iterations. $S(u, v, w, z) =: S(q, d(R))$, $R = (u, v, w, z)$.

$(t, b, l, r) \leftarrow (0, n_y - 1, 0, n_x - 1)$
for $i = 1$ to M **do**
  $t' \leftarrow \arg\max_{j=0,\ldots,b-1} S(j, b, l, r)$ and $b' \leftarrow \arg\max_{j=t'+1,\ldots,n_y-1} S(t', j, l, r)$
  $l' \leftarrow \arg\max_{j=0,\ldots,r-1} S(t', b', j, r)$ and $r' \leftarrow \arg\max_{j=l'+1,\ldots,n_x-1} S(t', b', l', j)$
  if $(t, b, l, r) = (t', b', l', r')$ **then**
    break
  end if
  $(t, b, l, r) \leftarrow (t', b', l', r')$
end for
return $R^* \leftarrow (t, b, l, r)$ and $\mathcal{S} \leftarrow S(t, b, l, r)$

Algorithm 2. Local BoF Retrieval

/*——————**Offline**——————*/
Quantize local descriptors and construct the inverted file.
for each database image $\{I_i\}_{i=1,2\ldots N}$ **do**
  Compute $\mathcal{H}_{d_i}$ using Eq. 12.
end for
/*——————**Online**——————*/
Given the query BoF q, use the Global BoF method to rank the images.
for all top-K images $\{I_{T_j}\}_{j=1,\ldots,K}$ on the ranked list **do**
  Compute $\mathcal{G}_{q,d_{T_j}}$ based on Eq. 11 or Eq. 17.
  Compute $R^*(I_{T_j})$ and $\mathcal{S}(I_q, I_{T_j})$ using Algorithm 1.
end for
for all top-K images $I_{T_1}, \ldots, I_{T_k}$ on the ranked list **do**
  Given $R^*(I_{T_j})$, recompute $\mathcal{S}(I_q, I_{T_j})$ using the non-binarized BoF.
end for
Rerank the top K images based on $\mathcal{S}(I_q, I_{T_j})$.
return $R^*(I_{T_j})$ and the reranked image list.

quantizer, and the indices are organized into an inverted file. Offline, we compute integral norm images over the coarse uniform grid for both binary and full BoFs. Online, we first compute the BoF for the query region, sort the images based on the standard BoF algorithm, and then perform the local BoF optimization to estimate the optimal rectangle, compute similarities, and rank images.

3.5 Local Spatial Pyramid Model (LSPM)

Inspired by Lazebnik *et al.* [8], we can extend the Local BoF model by imposing a weak spatial consistency constraint using a local spatial pyramid model. Specifically, we decompose the query region into different spatial quantization levels $P \times P$ ($P = 1, 2...$). In each pyramid level, we compute the similarity vote for each grid cell in this spatial quantization and average them, and average the similarities at all pyramid levels to obtain the pyramid-based Local BoF similarity. For data sets such as the Oxford Building data set, where objects are mostly

upright in the images, more levels of the spatial pyramid are more discriminative and hence result in better average precision. In our experiments we found that $P = 2$ is a good tradeoff between accuracy and complexity.

4 Results

We test the approach on two image retrieval data sets: the University of Kentucky data set(Ukbench)[1] [11], and the Oxford Building (5K) data set (Oxbuild)[2] [13]. Ukbench contains 10200 images of 2550 objects where each object has exactly four images. The evaluation metric is the average number of correct top-4 images for all 10200 queries. Oxbuild contains 5062 images of Flickr images and 55 standard test queries of 11 landmarks. The performance is evaluated as the mean average precision (mAP) score. We implemented our own retrieval system consisting of affine invariant region detection [10], SIFT [9] description, and hierarchical quantization methods, but for fair comparison to other approaches, our results here are based on the same features as [13] and [11] which are publicly available at the data set URLs. We used a fixed grid size (grid spacing meaning the size of one grid cell) of 80×80 pixels and performed reranking for top K images (where K=400 for Oxbuild and K=20 for Ukbench) in all experiments except the ones in Sec. 4.4, where the effects of these parameters are tested.

4.1 Results on Oxbuild

Since our method is aimed at improving retrieval on smaller queries, we have evaluated its performance as a function of query region size. For Oxbuild, we perform two types of 'query resize' experiments: (1) performance w.r.t. the resize ratio to the original query rectangle, *i.e.* test mAP values by varying the standard 55 query rectangles by fixing their center points and scales them uniformly by a set of constant factors ranging from 0 to 1; (2) performance w.r.t. the area (pixel size) of query subrectangle, *i.e.* test mAP values by choosing fixed-size query subretangles (the same number of pixels for all queries) with the same center and aspect ratio to the original query rectangles. Note that we resize query rectangles instead of the underlying query images. Fig. 2 (top) shows the comparison results of those two experiments for the L_1 case. As can be seen from the figure, both versions of our approach consistently outperform the Global BoF approach, improving mAP on average by 12.7% across all query resize ratios and 13.6% across all absolute query subrectangle sizes (pixels). In general, LSPM (level 2) showed better performance than Local BoF for larger scales due to its better discriminative power. For smaller queries, the advantage of using LSPM is not obvious due to the sparseness of the query features. More interestingly, the smaller the absolute query size, the more benefit is observed using our localized algorithms as shown in Fig. 2 (top-right).

[1] `http://www.vis.uky.edu/~stewe/ukbench/`

[2] `http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/index.html`

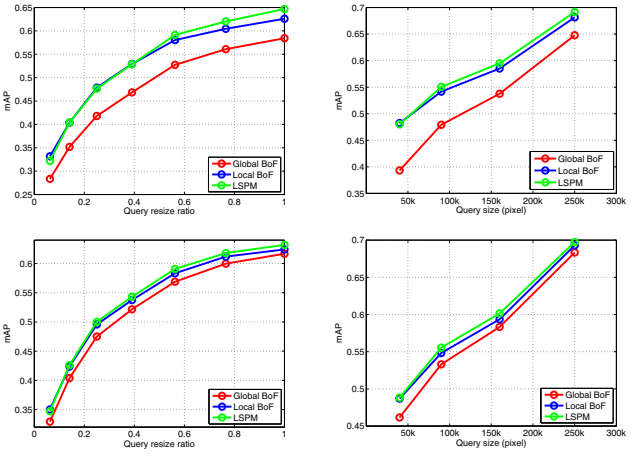

Fig. 2. Performance evaluation on Oxbuild for different approaches (Global BoF, Local BoF and LSPM) w.r.t. the query size. Top-Left: comparison w.r.t. the query size in ratios to the original query regions (the L_1 case). Top-Right: comparison w.r.t. the query size in pixels (the L_1 case). Bottom-Left: comparison w.r.t. the query size in ratios to the original query regions (the L_2 case). Bottom-Right: comparison w.r.t. the query size in pixels (the L_2 case).

Fig. 2 (bottom) shows the same experiments for the L_2 case. We can observe similar consistent improvement of our approach over the baseline but the improvement is less than the L_1 case, *i.e.* on average by 4.2% across all query resize ratios and 3.8% across all absolute query sizes (pixels). This is probably because the assumption that the L_2 norm becomes similar to the square root of the L_1 norm is less accurate due to the repeatitive structures in the data set.

In comparison to previous approaches, mAP of our approach of using the original 55 queries is 0.647 which is significantly better than the L_1 Global BoF (0.582) and L_2 BoF (0.618). And, our local BoF obtained almost identical result to the Global BoF with RANSAC-based reranking [13]. Note that our approach is significantly faster than RANSAC-based verification, and can be applied to thousands of the database images in less than 50ms during retrieval time (see Fig. 5 (right)). Another advantage of our approach is that it is not limited to rigid, mostly planar objects as in the RANSAC-based approach.

4.2 Results on UKbench

We performed the same query-size experiments for Ukbench. We resized each of the original query regions (entire image regions) by fixing its center to the center of the original image since there are no query regions are given. Since the vocabulary tree structure is not provided, we use our own HKM algorithm and the SIFT features (provided on the data set web page) to build a hierarchical vocabulary of 6 levels with the branching factor 10, and obtained the top-4 score of 3.29 which is the same as the best result of [11].

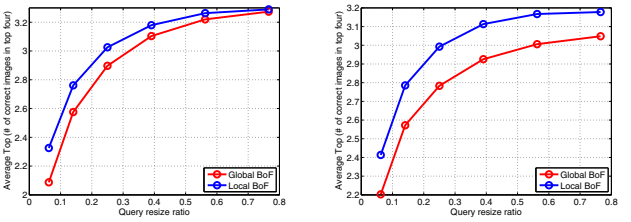

Fig. 3. Performance evaluation on Ukbench: average top-4 w.r.t. the query size. Left: the L_1 case. Right: the L_2 case.

Fig. 3 shows the results of the standard Global BoF and our Local BoF approaches. The improvement of our approach over the Global BoF is most significant for smaller queries, *i.e.* in general the smaller the query the more improvement we achieved. Specifically, the absolute improvement of the top-4 rate is 0.19 for the L_1 case and 0.21 for the L_2 case which are significant considering the strict true positive criterion used for the data set. For larger queries, our approach achieved relatively smaller improvement in retrieval because most of the images in this data set are close-up shots of objects. Comparing the L_1 and L_2 cases, the improvement for L_2 is more consistent over all query sizes.

4.3 Analysis of the Optimization Approach

We evaluated the number of greedy iterations needed for convergence during the local BoF search process using Oxbuild. As shown in Fig. 4 (left), retrieval performance improves with increasing iterations, but the improvement slows significantly after 2 iterations. Surprisingly all of the optimization for 22000 test images over 55 queries coverage in less than 4 iterations. For about 95% images, the process converges in only 2 iterations. Fig. 4 (right) validates proximity of our solutions to the global optimum when changing the area overlap ratio γ[3] to the globally optimum rectangle. Note that in about 66% of cases out of 22000 total localization tasks, our approach achieved the exact global optimum.

As shown in Table 1, we also compared the retrieval performance of our greedy-based approach and the globally optimum-based approach (branch-and-bound or brute-force search) with respect to the different query resize ratio β. Evidently, the greedy approach results in no significant degradation in mAP.

4.4 Analysis of the Algorithm Parameters

We first analyze the effect of changing the grid size (grid spacing) from 20×20 to 320×320, for a range of query sizes, using Oxbuild. From Fig. 5 (left) we can observe that the retrieval performance of the LSPM tends to be increasing and

[3] The area overlap ratio $\gamma(R_1, R_2)$ between two rectangles R_1 and R_2 is defined as: $\gamma = \frac{A(R_1 \bigcap R_2)}{A(R_1 \bigcup R_2)}$, where $A(Q)$ is the area of region Q.

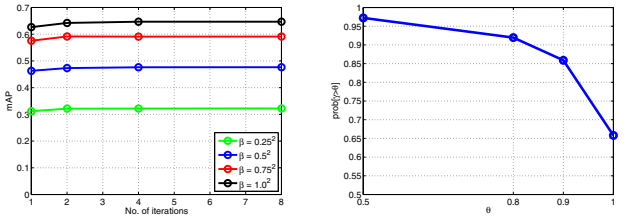

Fig. 4. Analysis of optimization on Oxbuild. Left: mAP w.r.t. the number of greedy iterations, β denotes the query resize ratio. Right: comparison of the greedy solution with the global optima: $Prob(\gamma > \theta)$ is computed as the percentage of localization tasks where the greedy solution rectangle overlaps with the global optimum rectangle when changing the area overlap ratio threshold θ.

Table 1. Performance (mAP) comparison of our greedy solution and global optimum localization-based approaches on Oxbuild

mAP	$\beta = 0.25^2$	$\beta = 0.5^2$	$\beta = 0.75^2$	$\beta = 1.0^2$
Greedy solution	0.322	0.476	0.591	0.647
Global optimum	0.329	0.481	0.591	0.644

leveling off with smaller grid sizes. While for larger grid sizes, the degradation in accuracy is more significant for small-size queries. This is reasonable because more precise localization can be achieved using smaller grids, and when the query size is smaller or similar to the grid size, LSPM becomes too sparse.

We also evaluated the performance by varying K, the number of top images to rerank, for a range of query sizes, using the same data set. Fig. 5 (middle) shows the mAP of the Local BoF and the LSPM w.r.t. K and query size. It is interesting to find that all curves level off with increasing reranking images. We can also observe a consistent mAP improvement of our approaches when moving from $K = 25$ to $K = 800$, *i.e.* the LSPM improves by 9.7% and the Local BoF improves by 8.4% on average. This figure indicates that the method is robust to K. Also, for most query sizes, the LSPM outperformed the Local BoF by about 2% on average but the LSPM resulted in a lower mAP than the local BoF for the smallest query due to the relatively coarser grid size (80×80).

4.5 Complexity and Scalability

Our approach only needs to store feature locations (x, y) (1 byte per feature coordinate) as the geometric information in the inverted file and does not require storing affine geometry parameters. Indexing 1 million images, averaging 500 features per image, requires about 1GB. In addition to the inverted file, our method requires storing L_1 or L_2 norm integrals of all database image BoFs. If each image has 48 grids, storing each integral requires 100 bytes. For 1 million images, this only amounts to 100MB, which is insignificant compared to the size

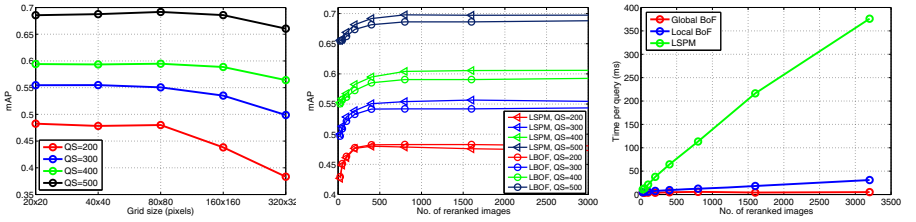

Fig. 5. Analysis of parameters using Oxbuild. Left: mAP w.r.t. the grid size (grid spacing in pixels) using the LSPM, 'QS=n' means the query region size is $n \times n$ pixels. Middle: mAP w.r.t. the number of reranked images, 'LBOF' stands for Local BoF and 'QS=n' means the query size is $n \times n$ pixels. Right: the query time comparison w.r.t. the number of reranked images.

of the inverted file. At retrieval time, our Local BoF is only slightly slower than the Global BoF approach when the optimization was run for the top 400 images, see Fig. 5 (right), and is only twice as slow when run on the top 1000 images. Typically the average query (or retrieval) time (ignoring the query feature extraction) for reranking the top 3200 images is less than 30ms compared to around 5ms for the global BoF approach. From the Local BoF curve, we can predict that our approach can spatially verify, rerank, and localize objects for 100k images in less than 1 second which is significantly faster than RANSAC-based spatial verification for the same number of reranking images. The speed is mainly due to the combination of a coarse grid, the integral image-based computation enabled by binary BoF approximation and greedy optimization.[4]

5 Conclusions

We have presented a local BoF model and its optimization method for efficient object retrieval. Our new contributions include (1) the generalization of the Global BoF to spatially localized models, Local BoF and LSPM, for reranking images and localizing objects in a unified framework, (2) the integration of the localized models with an inverted file index for efficient object retrieval in large image sets. Efficiency was achieved by introducing the binary BoF approximation. We have demonstrated consistent improvement over the baseline BoF on the average retrieval precision using our method. We have also shown that the method is much faster than alternative methods such as RANSAC.

Our local BoF framework is a general module that can be combined easily with numerous already published techniques including (1) different local features and their sample combinations [18], (2) varying quantization methods, such as HKM [11], AKM [13], Soft AKM [14], (3) specific cues, such as query expansion [13], (4) RANSAC [13], (5) gravity vector assumptions [13, 12], and (6) compression-based schemes, such as [12, 3]. We expect that combination with

[4] All the reported times are measured on a 3GHz machine with 16GB RAM.

such techniques will further improve object retrieval accuracy for large sets of images using the Local BoF framework.

References

1. Chum, O., Perdoch, M., Matas, J.: Geometric min-hashing: Finding a (thick) needle in a haystack. In: CVPR, pp. 17–24 (2009)
2. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
3. Jegou, H., Douze, M., Schmid, C.: On the burstiness of visual elements. In: CVPR, pp. 1169–1176 (2009)
4. Jegou, H., Douze, M., Schmid, C.: Packing bag-of-features. In: ICCV, pp. 1–8 (2009)
5. Ke, Y., Sukthankar, R., Huston, L.: Efficient near-duplicate detection and subimage retrieval. In: ACM Multimedia, pp. 869–876 (2004)
6. Lampert, C.H.: Detecting objects in large image collections and videos by efficient subimage retrieval. In: ICCV, pp. 1–8 (2009)
7. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR, pp. 1–8 (2008)
8. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, pp. 2169–2178 (2006)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
10. Matas, J., Chum, O., Urba, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC, pp. 384–396 (2002)
11. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: CVPR, pp. 2161–2168 (2006)
12. Perdoch, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. In: CVPR, pp. 9–16 (2009)
13. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR, pp. 1–8 (2007)
14. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR, pp. 1–8 (2008)
15. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV, pp. 1470–1477 (2003)
16. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV, pp. 1–8 (2009)
17. Wu, Z., Ke, Q., Isard, M., Sun, J.: Bundling features for large scale partial-duplicate web image search. In: CVPR, pp. 25–32 (2009)
18. Wu, Z., Ke, Q., Sun, J., Shum, H.Y.: A multi-sample, multi-tree approach to bag-of-words image representation. In: ICCV, pp. 1–8 (2009)
19. Yeh, T., Lee, J.J., Darrell, T.: Fast concurrent object localization and recognition. In: CVPR, pp. 280–287 (2009)