Interactive Visual Object Search through Mutual Information Maximization

Jingjing Meng¹, ², Junsong Yuan², Yuning Jiang², Nitya Narasimhan¹, Venu Vasudevan¹, Ying Wu³

¹ Applied Research Center, Motorola, Schaumburg, IL, USA
² School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore
³ Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA

ABSTRACT

Searching for small objects (e.g., logos) in images is a critical yet challenging problem. It becomes more difficult when target objects differ significantly from the query object due to changes in scale, viewpoint or style, not to mention partial occlusion or cluttered backgrounds. With the goal to retrieve and accurately locate the small object in the images, we formulate the object search as the problem of finding subimages with the largest mutual information toward the query object. Each image is characterized by a collection of local features. Instead of only using the query object for matching, we propose a discriminative matching using both positive and negative queries to obtain the mutual information score. The user can verify the retrieved subimages and improve the search results incrementally. Our experiments on a challenging logo database of 10,000 images highlight the effectiveness of this approach.

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]: Retrieval models

General Terms
Algorithms, Experimentation, Theory

1. INTRODUCTION

The development of invariant local features and the fast search algorithms allow us to search for small visual objects within large image databases [1] [2] [3] [4] [5]. Now, consider the cases where a user crops a visual object from one image, then searches for occurrences of that object within images in a large database. In such cases, we need to not only retrieve matching images, but also accurately locate the target object within each match.

Despite previous work, such small object search remains a challenging problem mainly due to two reasons. First, we have the matching problem. The target object may differ from the query object in scale, viewpoint, lighting or style, or be partially-occluded. Second, we have the localization problem. The presence of other objects or cluttered backgrounds can make it non-trivial to mark the target object efficiently and accurately. In addition, since a large number of candidate images need to be processed, the computational cost of localization is critical.

We propose an efficient object search approach that addresses the two challenges mentioned above. Given a query object, we formulate visual search as a localization problem to find the subimage with maximum mutual information toward the query object. Characterizing each image as a collection of local feature points, we first measure the point-wise mutual information score employing the non-parametric nearest neighbor classifier. The relevance score of the subimage is then computed as the summation of the relevance scores of its local features. We refer to this as the discriminative mutual information score (Fig. 1) because it involves both positive and negative examples for a discriminative relevance evaluation. Finally, upon user verification of the returned results, we update the relevance scores and re-rank the subimages.

This approach has the following advantages: (1) It does not rely on the visual vocabulary to index and match local features; thus there is no quantization loss during matching. The search results only depend on the matching of local features; thus this is more accurate and robust. (2) The proposed relevance score is compatible to the branch-and-bound search framework proposed in [1], where the complexity of the localization can be sub-linear in the number of images, as we do not have to exhaustively check every image and every subimage. However, because we do not adopt the vi-
sual word histogram to characterize subimages, we have a reduced memory cost when building the integral images. (3) It readily incorporates relevance feedback, where users can interactively clarify their preferences to further refine subsequent search results. The relevance score can be updated incrementally without requiring retraining of the classifier.

2. ALGORITHM

Given an image database \( \mathcal{D} = \{ I_i \} \), the goal of object search is to find the subimage \( I \subseteq \mathcal{I} \), where \( \mathcal{I} \subseteq \mathcal{D} \), so it has the maximum relevance score toward the query object. § 2.1 gives the definition of relevance score based on mutual information between the query and a subimage. § 2.2 describes how to derive the quality bound for the relevance score to enable efficient subimage retrieval via branch-and-bound. § 2.3 describes how to incrementally update relevance score with relevance feedback.

2.1 Discriminative Mutual Information Score

We characterize each image \( I \in \mathcal{D} \) as a collection of local invariant features \([6]\). Each feature point \( p = (x, y, d_p) \), where \((x, y)\) is the location and \( d_p \in \mathbb{R}^N \) is the feature descriptor. Given any subimage \( I = \{ p_i \} \), we want to measure its relevance to the query object.

We denote by \( \Omega^+ \) the positive class. Initially, \( \Omega^+ \) contains only one query object. After relevance feedback, \( \Omega^+ \) would expand to contain more examples that have been detected from the images. In general, we denote by \( \Omega^+ = \{ q_i \} \) the positive training dataset. The query object is represented by the collection of the interest points from all the positive sample images: \( \Omega^+ = \{ d_j \} \). In the feature space, this collection of feature points approximates the distribution of the query object class. For the negative training dataset \( \Omega^- \), we randomly pick some background images to initialize it. It is later updated by adding false detections through relevance feedback. \( \Omega^- \) is also characterized as the collection of feature points from all of its training images. In contrast to previous search methods that use only one positive query example for matching, our matching scheme is more discriminative by using both \( \Omega^+ \) and \( \Omega^- \). By incrementally updating \( \Omega^+ \) and \( \Omega^- \) online, our method can better adapt to the query example.

Inspired by the previous work in \([7]\), we measure the relevance score of a subimage \( I \) by the discriminative pointwise mutual information. Applying the Naive-Bayes assumption and assuming interest points are independent of each other, the relevance score of \( I \) is:

\[
s(I) = MI(\Omega^+, I) = \log \frac{P(I|\Omega^+)}{P(I)}
\]

\[
= \log \prod_{p \in I} \frac{P(d_p|\Omega^+)}{P(d_p)} = \sum_{p \in I} \log \frac{P(d_p|\Omega^+)}{P(d_p)}
\]

\[
= \sum_{p \in I} \log \frac{P(d_p|\Omega^+)}{P(d_p|\Omega^+|I)P(\Omega^+)} + P(d_p|\Omega^-)P(\Omega^-)
\]

\[
= \sum_{p \in I} \log \frac{1}{P(\Omega^+)} \cdot \frac{P(d_p|\Omega^+)}{P(d_p|\Omega^-)} + \frac{P(d_p|\Omega^-)}{P(d_p|\Omega^-)}P(\Omega^-) = \sum_{p \in I} s(p),
\]

where \( s(p) \) is the relevance score of an individual point \( p \). To evaluate the likelihood ratio for each \( p \in I \), we follow the same strategy in \([8]\) and apply the kernel density estimation based on the training data \( \Omega^+ \) and \( \Omega^- \). Applying the nearest neighbor approximation for the Gaussian kernel estimation, we have:

\[
\frac{P(d|\Omega^-)}{P(d|\Omega^+)} = \frac{1}{|\Omega^-|} \sum_{d_j \in \Omega^-} K(d - d_j) \approx \exp \left( -\frac{1}{2\sigma^2} \|d - \hat{d}_{NN}\|^2 \right). \tag{2}
\]

Here \( \hat{d}_{NN} \) is the nearest neighbor of \( d \) in \( \Omega^- \) and \( \hat{d}_{NN}^+ \) is the nearest neighbor of \( d \) in \( \Omega^+ \). We omit \( \frac{1}{|\Omega^-|} \) since only one nearest neighbor is considered in both classes. Such an approximation has been shown to be effective in image matching in \([8]\). To speed up the search of nearest neighbors, we find approximate \( \epsilon \)-nearest neighbors (\( \epsilon \)-NN) by locality sensitive hashing (LSH) \([9]\). The E2LSH software package is used here and the probability for correct retrieval is set to \( p = 0.9 \). We also set \( 2\sigma^2 = 8 \) in our experiments.

2.2 Branch-and-Bound Search

Our objective of object search is to find a subimage \( I^* \subseteq \mathcal{I} \), where \( \mathcal{I} \subseteq \mathcal{D} \), such that it has the maximum relevance score toward the query object:

\[
I^* = \arg\max_{I \subseteq \mathcal{I}, I \subseteq \mathcal{D}} MI(\Omega^+, I) \tag{3}
\]

\[
= \arg\max_{I \subseteq \mathcal{I}, I \subseteq \mathcal{D}} \sum_{p \in I} s(p) = \arg\max_{I \subseteq \mathcal{I}, I \subseteq \mathcal{D}} s(I),
\]

where \( s(I) = \sum_{p \in I} s(p) \) is the objective function.

Considering a single image \( I \) of size \( m \times n \), the total number of subimages of \( \mathcal{I} \) is in the order of \( O(n^2m^2) \). To avoid an exhaustive search, we also apply the branch-and-bound search as in ESR \([1]\), which enables object search in sub-linear time. Despite different objective function \( s(I) \), we will explain that branch-and-bound search is still feasible in our case. The details of the branch-and-bound search can be found in \([1]\) and below we only derive the upper bound function. Let \( I \) be the collection of all subimages in a given image \( I \). Assume there exist two subimages \( I_{min} \) and \( I_{max} \) such that for any \( I \subseteq I \subseteq I \subseteq I_{max} \). Then we have \( s(I) \leq s^+(I_{max}) + s^-(I_{min}) \), where \( s^+(I) = \sum_{p \in I} \max(s(p), 0) \) contains only positive votes, while \( s^-(I) = \sum_{p \in I} \min(s(p), 0) \) contains only negative ones. We denote the upper bound of \( s(I) \) for all \( I \subseteq I \) by:

\[
\hat{s}(I) = s^+(I_{max}) + s^-(I_{min}) \geq \max_{I \subseteq I} s(I). \tag{4}
\]

It is easy to see that if \( I \) is the only element in \( I \), we have the following equality:

\[
\hat{s}(I) = s(I). \tag{5}
\]

Eq. 4 and Eq. 5 thus meet the two requirements discussed in \([10]\) and serve as the effective upper bound for branch-and-bound search. It is worth mentioning that our new objective function \( s(I) \) has two advantages over the histogram-based functions used in ESR. First, local feature matching becomes more accurate, since we do not quantize the visual features. Second, memory cost is lower when building the integral images for branch-and-bound search, because our method only needs to store one number for each pixel, instead of a histogram per pixel as in ESR.
2.3 Incremental Updates of Relevance Score via Relevance Feedback

Given the training dataset $\Omega^+$ and $\Omega^-$, branch-and-bound search returns the subimages of highest relevance scores. After user verifies the returned results, we update the training datasets by adding correct detections $\{\Gamma^+\}$ to $\Omega^+$ and false detections $\{\Gamma^-\}$ to $\Omega^-$. Since the training datasets are updated, the relevance score $s(I)$ needs to be updated accordingly. As we apply the non-parametric nearest neighbor classifier, the relevance score $s(I)$ only depends on the relevance score $s(p)$ of individual point $p$. This enables us to update $s(I)$ incrementally without re-training the classifier. Specifically, according to Eq. 2, the score $s(p)$ relies on the two distances: (1) the distance from $p$ to the positive point set $\Omega^+$, $d_{\Omega^+}$, and (2) the distance from $p$ to the negative point set $\Omega^-$, $d_{\Omega^-}$. As a result, to update $s(p)$, we only need to update $d_{\Omega^+}$ and $d_{\Omega^-}$. Assuming the current training datasets are $\Omega^+$ and $\Omega^-$, and the newly verified subimages form $\Omega^+_n$ and $\Omega^-_n$. Now the new training datasets become $\Omega^+ = \Omega^+_n \cup \Omega^+$, and $\Omega^- = \Omega^- \cup \Omega^-_n$. Let $d_{\Omega^+_n}^{\Omega^+_1}$ and $d_{\Omega^-_n}^{\Omega^-_1}$ denote the nearest neighbors of $p$ in $\Omega^+_n$ and $\Omega^-_n$ respectively, we only need to calculate $d_{\Omega^+_n}^{\Omega^+_1}$ and $d_{\Omega^-_n}^{\Omega^-_1}$ from $\Omega^+_n$ and $\Omega^-_n$, respectively. Finally we can update $d_{\Omega^+} = \min\{d_{\Omega^+_n}^{\Omega^+_1}, d_{\Omega^+_n}^{\Omega^+_2}\}$ and $d_{\Omega^-} = \min\{d_{\Omega^-_n}^{\Omega^-_1}, d_{\Omega^-_n}^{\Omega^-_2}\}$.

3. EXPERIMENTS

3.1 Implementation

We test our algorithm on the challenging BelgaLogos dataset [4], which consists of 10,000 images covering diverse categories of objects and events. As in [4], images in the dataset are resized so the maximum value of height and width is 800 pixels, preserving the original aspect ratio. We extract scale and affine invariant interest points from each image in the dataset using Harris-Affine detector [11], and characterize them using 128-D SIFT descriptors for point matching [6]. After feature extraction, the entire dataset results in a total of 24,172,440 SIFT points.

As in [4], 6 external logos from Google first result page are used as queries to test our algorithm (see Table 1 and supplementary materials). Initially, $\Omega^+$ only contains the single query, while $\Omega^-$ consists of two subimages, each containing a face(Fig. 1), which are cropped out from two images randomly selected from the dataset.

Our implementation includes two stages. The first stage prunes irrelevant images before subimage search. So for each feature point in $\Omega^-$, we find its $\epsilon$-nearest neighbors in the BelgaLogos dataset using LSH. Then we filter images that contain less than $3\% \times M$ matches to the points in $\Omega^-$ ($M$ is the total number of feature points in $\Omega^-$). The second stage performs interactive subimage search on the remaining images. It includes the following steps, and a user can iterate the process until they are satisfied with the results:

1. For each feature point in the dataset, find their matches in current $\Omega^+$ and $\Omega^-$ using LSH;
2. Calculate point-wise relevance score $s(p)$ (Eq. 1);
3. Use branch-and-bound search (§ 2.2) to retrieve subimages, whose relevance score $s(I)$ is greater than a given detection threshold $\tau$. $\tau$ controls the number of detected subimages. Abnormal subimages with an aspect ratio greater than 10 are ignored.
4. User verifies the top $n$ retrieved subimages, adds correct detections to $\Omega^+$ and false detections to $\Omega^-$. Since the number of false detections is usually much greater than the number of correct detections in a huge dataset, to balance the sizes of $\Omega^+$ and $\Omega^-$, we use $K$-means to cluster feature points from all false detections and add the cluster centers to $\Omega^-$. In our experiments, the LSH range query parameter $\epsilon$ is set to 300 for the Presidential Seal and 230 for the other five queries. The detection threshold $\tau$ is set to 3.0 for all six queries. Both parameters stay the same in each round. We only iterate the search twice for each logo.

3.2 Results Evaluation

To evaluate the performance in each round, we calculate the precision and recall scores of detected subimages. Precision is the percentage of correct detections in all detections, while recall is the percentage of correct detections against the total number of ground truth images. Since BelgaLogos dataset does not provide locations of a given logo in its ground truth images, we estimate the recall based on the total number of images that contain a query logo. If more than 100 subimages are returned, only the top 100 results are used as detection results for user verification. Otherwise, we verify all returned subimages. All detected subimages are manually verified to ensure that they indeed contain the query.

Interestingly, we have also detected logo instances that are not included in the ground truth files [4] for Mercedes (Fig. 3) and Peugeot. In both cases, we update the total number of ground truth images and calculate the recall accordingly. If multiple logos are found in the same image, we only count them once for our recall calculation. Table 1 summarizes our results for the 6 logos. After one round of relevance feedback, both precision and recall scores have been improved significantly.

To demonstrate the effectiveness of our method, we also compare it with ESR [1]. Using the same set of local features, we first build a codebook of 10,000 words by a hierarchical clustering of 5% of the SIFT points. We then perform ESR using normalized histogram intersection (NHI) as the quality function, which is shown to be effective [1]. We apply the code provided in [1], but exhaustively search the bounding boxes with multiple scales, ranging from 0.25 to 8 times
of the original query size, with a step factor of 1.1. The precisions and recalls are also shown in Table 1. To make a fair comparison, for the precision score of each logo, we check the same number of top detections as those returned from our algorithm, while the recall is based on the total number of ground truth images.

Experimental results on the 6 logos demonstrate that our algorithm performs slightly better than ESR even without user feedback in the first round, while the second round shows more significant improvements. The detection results (Fig. 2, Fig. 3) demonstrate that our method can effectively handle many challenging cases for object retrieval, like changes in scale and viewpoint, deformations, blurring, as well as severe partial occlusions (e.g., 1\textsuperscript{st} image in 2\textsuperscript{nd} row, Fig. 2). Our method can also detect multiple objects in a single image (Fig. 3) and handle multiple query images (§ 2.3).

4. CONCLUSION

Retrieving small visual objects from a large collection of images is a challenging problem due to the possible appearance variations of objects, as well as the large computational cost to search the image dataset. Our method combines subimage search with relevance feedback to search and locate small objects. Using the branch-and-bound search, it ranks the subimages (instead of whole images) by the mutual information score toward the query object. Once the user verifies the relevant and irrelevant subimages, the mutual information score is updated incrementally and the search results can be further improved. Our experiments on a challenging logo dataset validate the effectiveness and efficiency of our method. Our future work includes how to further reduce the search cost when handling millions of images.

Acknowledgments

This work is supported in part by the Nanyang Assistant Professorship (SUG M58040015) and National Science Foundation grant IIS-0347877 and IIS-0916607.

5. REFERENCES


