Retrieval of images with objects of specific size, location and spatial configuration

Niloufar Pourian  B.S. Manjunath
Department of Electrical and Computer Engineering
University of California, Santa Barbara, United States
npourian@ece.ucsb.edu manj@ece.ucsb.edu

Abstract
An approach to image retrieval using spatial configurations is presented. The goal is to search the database for images that contain similar objects (image-patches) with a given configuration, size and position. The proposed approach consists of creating localized representations robust to segmentation variations, and a sub-graph matching method to compare the query with the database items. Localized object representations are created using a community detection method that groups visually similar segments. Extensive experimental results on three challenging datasets are provided to demonstrate the feasibility of the approach.

1. Introduction
Searching for images with a specific visual content has been a topic of intense research in recent years [3]. However, majority of the prior work focuses on searching for images using global (whole image) attributes, and lacks discrimination based on localized objects or their relative spatial positioning in the images. We consider here a more generalized problem in which the objects of interests are provided by a query set that includes multiple image-patches or images along with the desired spatial configuration, size, and location of such patches in the target image.

Our goal is to develop localized representations that would enable queries similar to the one shown in Figure 1. Here the user provides objects/image-patches, together with specifications on their spatial configuration, size and position in the image. From such a specification, an image query is generated and matched against the database. To achieve this goal, localized representations are needed. We propose a robust graph-based representation that is learned from image-part groupings, and encodes size, location, and spatial configurations of objects/patches. Sub-graph matching is used to search and retrieve similar configurations.

Conventional methods usually represent a query (for instance a “dog”) through a single image or a set of images, possibly along with some textual description of the query [1, 34]. Many approaches focus on the global image representations [1, 16, 23], while some encode the spatial information of image features to improve the discriminative power of the feature representations [13, 11, 19]. In [19, 15, 33], a large number of key-point based descriptors are computed and their relative spatial relationships are encoded. Also, [34] calculates the location offset of two matched features. [14] utilizes the spatial co-occurrence information of visual words mined from database images to boost the retrieval performance. The work in [11] incorporates spatial layout by introducing a Gaussian location model per visual word and encoding only the absolute spatial information. Utilizing localized grids into the feature representation is also a common approach to integrate spatial information [2, 10, 13, 24]. These methods often result in high-dimensional representation and rely on a predefined partitioning of the image which is independent of its content. Moreover, they are not generally concerned with retrieving an exact spatial configuration that exists between objects of interest.

Alternatively one can compute localized features using segmented image regions. [4] investigates object segmen-
2. Proposed approach

In this section, we describe the details of the our model as illustrated in Figure 2.

2.1. Attributed graph structure

Suppose $I$ is an image in a set containing all training images $\{1, 2, \ldots, D\}$, and $G(I) = (V(I), E(I))$ is a graph with $V(I)$ and $E(I)$ representing the nodes and edges of this graph, respectively. Each node corresponds to an image part indicated by a segmented region. Two nodes are connected by an edge if the corresponding regions are adjacent, i.e. $E_{ij}(I)$ is 1 iff $i$ is adjacent to $j$ and 0 otherwise.

To retrieve images with similar spatial configurations, a query is also defined using a graph-based representation denoted by $G(Q) = (V(Q), E(Q))$, where the nodes correspond to the provided image-patches and edges represent adjacent regions defined by the specified configuration.

The segmentation is computed using the method of [7]. To represent regions (nodes), we extract densely sampled SIFT features [18] from each image, and map each 128-dimensional feature vector to a segment that they belong to. Each node is then represented by vector $h(i)$ using the Bag of Words (BOW) [28] model. In what follows, the appearance of node $i$ is denoted by $h(i)$. 

With their desired spatial configuration, size, and/or location in an image (Figure 1). We use an attributed graph for each of the training images based on segmented regions to capture the relative spatial information and adopt an algorithm to collectively learn image parts across all training images. This is done by discovering different groups (communities) of related image parts based on spatial and visual characteristics using a spectral clustering approach. This provides a way to compensate for variations in segmentation. Each segmented region in an image is represented by a community with the highest strength of association. Based on these communities, a robust graph representation is derived for sub-graph matching between the query configuration and training images. The highest matching scores would correspond to images that are most similar to the constructed query through a formulation that will be discussed in Section 2.3.

In summary, our contributions is twofold: First, a new graphical image representation based on segmentation is proposed. Second, an approach to a query retrieval problem using image-patches and spatial configuration is presented.

The remainder of this paper is organized as follows. In Section 2, we describe the overall framework of the proposed retrieval system. The applicability of the proposed approach is illustrated in Section 3 through a query retrieval problem on three challenging datasets. Finally, we conclude the paper with some final remarks and directions for future research in Section 4.
2.2. Learning image parts

The proposed model utilizes a spatially localized feature representation that captures attribute similarity with the relative spatial information without a strong dependence on segmentation. In this section we follow the localized feature representation introduced by [20] and provide a brief summary. We define a network of segmented image regions to integrate the visual similarity between segmented regions across all training images with the localized spatial information. In this network, two nodes $i$ and $j$ are considered related if they are spatially adjacent or if node $i/j$ belongs to the set of $T$ most similar nodes to node $j/i$ based on their visual characteristics denoted by $h^{(i)}$ and $h^{(j)}$. We use the Hellinger metric [21] to compute the distance between $h^{(i)}$ and $h^{(j)}$. For $L_1$ normalized $h^{(i)}$ and $h^{(j)}$, distance $d(h^{(i)}, h^{(j)})$ is computed by:

$$d(h^{(i)}, h^{(j)}) = \left( \sum_{k=1}^{K} \left( \sqrt{h_k^{(i)}} - \sqrt{h_k^{(j)}} \right)^2 \right)^{1/2}$$

(1)

with $K$ denoting the size of the codebook for BOW (number of clusters found by approximate kmeans).

In constructing this network, first, two nodes $i$ and $j$ are connected by a weighted edge equal to their attribute similarity (defined in equation 2) if node $i/j$ belongs to the set of $T$ most similar nodes to node $j/i$. Second, two spatially adjacent nodes are connected with a weighted edge equal to the average of the weights of all edges connected to the corresponding nodes.

The attribute similarity between two nodes $i$ and $j$ is given by the following:

$$\omega(i, j) = e^{-d(h^{(i)}, h^{(j)})} \times \mathbb{I}(L(i) = L(j))$$

(2)

where $d(h^{(i)}, h^{(j)})$ represents the distance between appearances of two nodes $i$ and $j$, $L(i)$ denotes the label associated with the image that node $i$ belongs to, and $\gamma$ is a constant larger than 1. We set $\gamma > 1$ to give a higher weight to the visual similarity of two nodes that belong to images with the same label. The function $\mathbb{I}(x)$ represents the indicator function and is equal to 1 if $x$ holds true and zero otherwise.

Next, a spectral clustering technique is applied to this network to aggregate related regions. For graph partitioning, we use the normalized cut method [26]. Each partition is referred to as a community. One can think of each community as a bag that contains all parts of an object.

Let $H_i$ denote the set of all nodes in the spatial neighborhood of node $i$, $\phi_c$ be a community with $c \in \{1, \ldots, C\}$, and $T_i$ denote the set of all nodes that are in the top $T'$ nearest neighbors of node $i$. The strength of association of a node $i$ to a community $\phi_c$ is measured by two factors: first by the attribute similarity between node $i$ and community $\phi_c$, second by considering the attribute similarity between neighbors of node $i$ and different communities in the network along with the relation between community $\phi_c$ and each of the communities in the network.

Figure 2: (a) Sample images, their corresponding segmented images and their initial graphical image representations. (b) An example of network of segmented images. Dotted lines represent connections between segmented regions based on spatial adjacency or visual similarity. Each color of an ellipse represents a different community. (c) Resulting images after mapping each of the segmented regions to the detected communities and its corresponding updated graphical image representation. This graph is used for computing the graph matching score. Figure is best viewed in color.
Let \( g(i \in \phi_c) \) denote the attribute similarity between node \( i \) and community \( \phi_c \). The function \( g(i \in \phi_c) \) is defined by the fraction of top \( T' \) nearest neighbors to node \( i \) that belong to community \( \phi_c \):

\[
g(i \in \phi_c) = \frac{\sum_{j \in \hat{T}_i} I\{j \in \phi_c\}}{T'}. \tag{3}
\]

Moreover, \( f(\phi_{c'} , \phi_c ) \) is defined to measure the relation between two communities \( \phi_{c'} \) and \( \phi_c \):

\[
f(\phi_{c'} , \phi_c ) = \frac{\sum_{i \in \phi_{c'}} \sum_{j \in \phi_c} I\{A_{i,j} > 0\}}{\sum_{i \in \phi_{c'}} \sum_{j = 1}^N I\{A_{i,j} > 0\}}, \tag{4}
\]

where \( N = |V| \) denotes the total number of nodes in the network. In particular, \( f(\phi_{c'} , \phi_c ) \) measures the number of links between the two communities \( \phi_{c'} \) and \( \phi_c \) divided by the total number of links between community \( \phi_{c'} \) and all other communities. Thus, the strength of association of a node \( i \) to a community \( \phi_c \) can be determined by \( P^{(i)}_c \):

\[
P^{(i)}_c = \frac{\sum_{c' = 1}^C \sum_{j \in \hat{N}_i} f(\phi_{c'}, \phi_c ) g(j \in \phi'_c) g(i \in \phi_c) \sum_{i' \in \phi_{c'}} f(i' \in \phi_{c'}, \phi_c ) g(j \in \phi'_c) g(i \in \phi_{c'})}{\sum_{c' = 1}^C \sum_{j \in \hat{N}_i} \sum_{i' \in \phi_{c'}} f(i' \in \phi_{c'}, \phi_c ) g(j \in \phi'_c) g(i \in \phi_{c'})}, \tag{5}
\]

where \( C \) denotes the total number of detected communities. Now one can use a maximum likelihood classifier to classify each node by the community with the strongest association:

\[
h^{(i)}_u = \arg \max_e P^{(i)}_c \tag{6}
\]

where \( h^{(i)}_u \) represents the updated representation of each node \( i \). Learning these image part groupings enables one to illustrate the image using much smaller number of pieces. Choosing an appropriate number of detected communities allows us to set the number of pieces equal to the number of objects that are present in an image.

To find images with the same configuration as the query, one can adopt a sub-graph matching approach and retrieve images with the highest matching score between their corresponding graph representations. However, such a sub-graph matching do not generally perform well due to variations in segmentation. This effect can be reduced by an updated graph representation (Figure 3) that enables one to perform a more robust sub-graph matching for retrieval. Two nodes \( i \) and \( j \) in graph \( G^{(I)} \) are merged if their updated representations are the same and also they are adjacent:

\[
j = \begin{cases} 
i & \text{"merging"} \quad \text{if} \quad h^{(i)}_u = h^{(j)}_u \\
j & \text{"not merging"} \quad \text{if} \quad h^{(i)}_u \neq h^{(j)}_u \end{cases}. \tag{7}
\]

In the remainder of this paper, we represent the updated graph of image \( I \) by \( G^{(I)}_u \). One can compute the sub-graph matching score between the updated graph representations for the query and the database images to find images that best match the query of interest while preserving the spatial configuration, as well as objects’ sizes and positions.

### 2.3. Graph matching

Let \( Q \) be the query image. We are interested in retrieving images that match the exact spatial configuration, size, and position of the segmented regions of \( Q \). Let \( G^{(Q)}_u = (V^{(Q)}_u , E^{(Q)}_u ) \) and \( G^{(I)}_u = (V^{(I)}_u , E^{(I)}_u ) \) be the updated attributed graph representations for a test image \( Q \) and a training image \( I \), respectively. Let \( n_Q = |V^{(Q)}_u| \) and \( n_I = |V^{(I)}_u| \) represent the number of nodes in each updated attributed graph. In this section, a mapping is found between \( V^{(Q)}_u \) and \( V^{(I)}_u \) that best preserves the attribute between the two graphs. We seek a set of matches \( M = \{i_Q i_I\} \) to maximize the graph matching score.

To get a matching score between \( G^{(Q)}_u \) and \( G^{(I)}_u \), a modified version of balanced graph matching algorithm [5] is used. Let \( x \in \{0, 1\}^{n_Q n_I} \) be a binary vector, such that \( x_{i_Q i_I} = 1 \) if \( i_Q i_I \in M \). We require to have a one-to-one mapping constraint, this is \( \sum_i x_{i_Q i_I} = 1 \) and \( \sum_i x_{i_i i_Q} = 1 \). The matching score between graphs of \( Q \) and \( I \) is defined by solving the following optimization problem which takes the form of an Integer Quadratic Program:

\[
s(Q, I) = \max_x \frac{\sum x^T W x}{x^T x} \quad \text{s.t.} \quad B x = b \tag{8}
\]

where \( s(Q, I) \) represents the graph matching score between \( Q \) and \( I \), and \( W \) is a \( n_Q n_I \times n_Q n_I \) comparability matrix indicating the similarity between nodes and between edges. An example of matrix \( W \) is illustrated in Figure 4. It is worth noting that \( B x = b \) represents the mapping constraint. For one-to-one matching, we let \( b = 1 \) in (8). This optimization problem is solved using spectral matching and by computing the leading eigenvector \( x \) of \( W \).

The similarity between nodes and between edges is de-
fined by the following:

\[ W_{q_i,j_i} = \exp \left( -\frac{\Delta(A_{q_i,j_i}^{(q)}, A_{i,j}^{(I)})}{\alpha} \right) \]  

(9)

where \( \Delta \) is a function that represents the distance between two nodes or between two edges and will be defined shortly. Here, \( A^{(I)} \) denotes an adjacency matrix associated with the updated graph of image \( I \). Diagonal elements of the adjacency matrix \( A^{(I)} \) are vectors corresponding to the updated node representations of \( G^{(I)}_u \), and their corresponding normalized size and position. The off-diagonal entries contain scalar binary values representing the edges between the nodes of \( G^{(I)}_u \). For different values of \( \alpha \in [0, 1] \), one can emphasize more on the importance of the node similarity versus edge similarity. When matching nodes characteristics, smaller values of \( \alpha \) correspond to a less emphasis on node similarities compared to edge similarities.

The \( \Delta \) function in (9) is defined by the following:

\[
\begin{align*}
\text{if } (i_Q = j_Q \text{ and } i_I = j_I) \rightarrow \text{"comparing nodes"}: \\
\Delta \left( A_{q_i,j_i}^{(q)}, A_{i,j}^{(I)} \right) &= \beta_1 \left[ 1 - \delta \left( h_u^{(q)}, h_u^{(I)} \right) \right] + \\
&\quad \beta_2 \left| S_i^{(q)} - S_i^{(I)} \right| + \beta_3 \left| P_i^{(q)} - P_i^{(I)} \right|
\end{align*}
\]

(10)

\[
\begin{align*}
\text{if } (i_Q \neq j_Q \text{ and } i_I \neq j_I) \rightarrow \text{"comparing edges"}: \\
\Delta \left( A_{q_i,j_i}^{(q)}, A_{i,j}^{(I)} \right) &= 1 - \delta \left( A_{q_i,j_i}^{(q)}, A_{i,j}^{(I)} \right)
\end{align*}
\]

\[
\begin{align*}
\text{Otherwise } \rightarrow \text{"comparing nodes and edges"}: \\
\Delta \left( A_{q_i,j_i}^{(q)}, A_{i,j}^{(I)} \right) &= 0
\end{align*}
\]

3. Evaluation

Database: We have performed experiments using three challenging datasets: PASCAL VOC2007 [8], ImageNet ILSVRC2010 [6] and TREC. PASCAL VOC2007 is a publicly available dataset containing 9,963 images and 20 object classes. A subset of ImageNet ILSVRC2010 [6] which contains roughly about 1 million images is added to the VOC2007 dataset as distractors to test the scalability of our system with respect to the size of the dataset. The resulting combined dataset is referred to as “VOC+ImageNet”. To evaluate the performance of different methods in identifying the object of interest when it occupies only a small portion of the image in a cluttered background, a set of images is collected by extracting frames from TRECVID 2012 instant search (INS) dataset [22] and it is referred to as TREC dataset. Since the groundtruth is only published for a subset of the data, only classes that have sufficient numbers of true positives are considered. TREC dataset contains 10,289 images and 10 object classes.

Qualitative analysis: To show the effectiveness of our approach, we consider a query consisting of multiple
image-patches (objects). Our goal is to retrieve images from the dataset that not only include such objects but also satisfy a set of requirements such as size, position, and spatial configuration provided by the user. Figure 5 (a) illustrates the image-patches of interest that should appear in the target image with the desired spatial configuration, size, and position. Using the proposed approach the top 3 retrieved images from dataset “VOC07” are depicted in Figure 5 (b). As one can see, our method returns images that satisfy the requirements provided in Figure 5 (a).

**Baseline methods:** Since the state of the art image retrieval systems can not be applied to queries as presented in Figure 1, for fair comparison, we consider the case in which the query is a single image with the goal of retrieving an exact spatial configuration of its segmented regions. This can be viewed as a sub-class of problems that can be handled by our approach.

The performance of the proposed system is compared with the retrieval system based on the Spatial Pyramid Matching (SPM) by encoding the global positions of features in the image [13].

We further compare our method with the case in which each image in the dataset is segmented and represented by a graph of its segmented regions. Here, each segmented region is represented solely by a BOW representation [28]. This will be referred to as “Basic Graph Matching”.

**Performance:** Each database is divided into a training set and a testing set. For each class, a model is learned using image features, their corresponding class types, and a SVM classifier. These models are used to estimate the class type of each image as well as their associated confidence score. The estimated class types for each image are then sorted based on their confidence scores. Each image is only associated with a set of class types that have the top $Y$ confidence scores.

Given a query image, the above model is used to associate the query image with a set of class types with top $Y$ confidence scores. In what follows, a “positive set” refers to a set containing test images that share at least one class type with the query. The rest of the database images are referred to as a “negative set”. Images included in the positive set are ranked higher than the ones in the negative set. In addition, images in each set are re-ranked based on their similarity score to the query image. The similarity is defined by $\omega = e^{-d(R(I), R(Q))}$ where $d(R(I), R(Q))$ denotes the Hellinger distance between two image representations indicated by $R(I)$ and $R(Q)$.

In “Basic Graph Matching” and the proposed approach, the retrieved images for each query in the positive and negative sets are re-ranked separately. This ranking is based on combined graph matching scores and visual similarity scores. This process is summarized in Figure 6.

In our experiments, a set of queries are randomly selected from each dataset and the accuracy of the retrieval system is measured using mean-average-precision (MAP). In computing the retrieval accuracy, each image $I$ in the test set is weighted based on how much its characteristics match the query $Q$. These characteristics are class type $\mathbb{C}$, spatial configuration $A$, size $S$ and position $P$. Given a query $Q$, we define a label $L$ for each image $I$ to specify how much its characteristics matches the corresponding ones in the query:

$$L_{Q,I} := [\mathbb{C}_{Q,I}, A_{Q,I}, S_{Q,I}, P_{Q,I}]$$ \hspace{1cm} (11)

where $\mathbb{C}_{Q,I}$ is one iff $Q$ and $I$ both belong to the same class type, otherwise zero. Similarly, $A_{Q,I}, S_{Q,I},$ or $P_{Q,I}$ are either one or zero. In particular, they are set to one if $Q$ and $I$ are both from the same class type and their corresponding spatial configuration, size, or position match, respectively.

Finally, the weight that determines how much each image characteristics match the query is given by:

$$\tau(Q,I) = L_{1}(L_{Q,I}).$$ \hspace{1cm} (12)

Table 1 illustrates the accuracy of the proposed retrieval system compared with the baseline methods. It is shown that the proposed approach achieves a higher retrieval accuracy than the baseline methods by 11 percent in the VOC dataset and 15 percent in the TREC dataset.

In addition, Figure 7 shows a comparison between the retrieval accuracy of different methods at different depths for the “VOC+ImageNet” dataset. These results emphasize the scalability of our approach.
Figure 6: Process of ranking the retrieved images for the methods discussed in Section 3.

<table>
<thead>
<tr>
<th>Database: VOC07</th>
<th>Method / K</th>
<th>200</th>
<th>500</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial Pyramid $\ell_1$</td>
<td>0.39</td>
<td>0.44</td>
<td>0.52</td>
<td></td>
</tr>
<tr>
<td>Spatial Pyramid $\ell_2$</td>
<td>0.41</td>
<td>0.46</td>
<td>0.54</td>
<td></td>
</tr>
<tr>
<td>Basic Graph Matching</td>
<td>0.42</td>
<td>0.48</td>
<td>0.54</td>
<td></td>
</tr>
<tr>
<td>Proposed Approach</td>
<td><strong>0.60</strong></td>
<td><strong>0.63</strong></td>
<td><strong>0.65</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Database: TREC</th>
<th>Method / K</th>
<th>200</th>
<th>500</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial Pyramid $\ell_1$</td>
<td>0.37</td>
<td>0.39</td>
<td>0.42</td>
<td></td>
</tr>
<tr>
<td>Spatial Pyramid $\ell_2$</td>
<td>0.39</td>
<td>0.41</td>
<td>0.43</td>
<td></td>
</tr>
<tr>
<td>Basic Graph Matching</td>
<td>0.40</td>
<td>0.42</td>
<td>0.44</td>
<td></td>
</tr>
<tr>
<td>Proposed Approach</td>
<td><strong>0.52</strong></td>
<td><strong>0.56</strong></td>
<td><strong>0.59</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Comparison of the accuracy of different retrieval systems discussed in Section 3 with different codebook sizes ($K$). The results are reported for VOC2007 and TREC databases.

**Computational cost:** A moderate computational complexity is important when considering scaling to thousands of images and hundreds of categories. The computational cost of learning image parts is considerably reduced by performing the search using a space partitioning data structure (k-d tree) with running time of $O(\log N)$. Also, as the size of the database increases, the network can be constructed using a subset of the images from the training data. Therefore, the cost of network construction would not increase. It is also worth noting that the computational cost related to the graph matching algorithm of the proposed approach is relatively small as the number of nodes in the updated graph representations are considerably fewer than the initial graph representations. The overhead cost for the retrieval of each query compared to the global image representations is roughly equal to 0.1 seconds using a quad core computer with 3.0 GHz processor.

4. Conclusion

We presented an approach to image search using image-patches and pre-specified spatial configurations. In general, such queries can not be handled by global image representations. The updated graphical structures are robust to segmentation variations and are suitable for sub-graph matching. Extensive experiments conducted on challenging datasets demonstrate that the proposed approach compares favorable with current state of the art methods. For future work, we plan to explore the applicability of the proposed method for enhanced object tracking.

966
5. Acknowledgments

This work is supported by ONR grant #N00014-12-1-0503.

References