Efficient Scene Image Clustering for Internet Collections

Heng Yang
School of Computer Science
Northwestern Polytechnical University
Xi’an 710072, P. R. China
yh3191@gmail.com

Qing Wang
School of Computer Science
Northwestern Polytechnical University
Xi’an 710072, P. R. China
qwang@nwpu.edu.cn

Zhoucan He
School of Computer Science
Northwestern Polytechnical University
Xi’an 710072, P. R. China
hezhoucan@163.com

Abstract—This paper proposes an efficient approach to find clusters of spatially related scene images collected from the website. Our method firstly builds a guide table, in which the ranked results are given according to the relevance scores of image pairs obtained by the image retrieval methods. Then the image clusters are generated by repeatedly choosing a seed image and performing query expansion directed by the guide table. In the query process, feature matching is performed by using an affine invariant constraint which is presented to effectively reject outliers of the image feature correspondences. The proposed image clustering approach has been tested on the Bell Tower dataset consisting of more than 1K images which are collected from the photo-sharing website Flickr.com. The experimental results demonstrate the efficiency and effectiveness of our method.

Keywords—image clustering, bag of visual features, query expansion, area ratio constraint.

I. INTRODUCTION

With the sharply explosion of the digital photography over the internet, people can easily obtain large image collections of the famous places that they are interested in using the related keywords searching from the photo-sharing websites, such as Flickr.com [1]. However, the searching results are always not well organized and even heavy contaminated by images with wrongly associated tags, which makes people fail to grasp the highlights of the desired scene. As a result, automatic image clustering or grouping for the web collections has become the hot issue in recent years, and it is also a challenging problem due to the huge size of the dataset.

The aim of this paper is to propose an efficient image clustering algorithm to categorize huge size scene image collections into different groups and each group contains the spatially overlapping images.

A great number of approaches for addressing large scale image clustering and categorization issue are presented in literatures. Snavely et al. [2] developed Photo Tourism system which can browse and structure large scale images by 2D and 3D visualization. One of the key steps of the system is to automatically group the large amount of unordered images by exhaustively calculating feature match between each possible image pair using ANN (Approximate Nearest Neighbor) searching algorithm. Therefore, the computational complexity of their grouping strategy is $O(n^2)$, where $n$ is the number of images of the dataset. In case of dealing with the huge size dataset, the time cost is quite heavy and even unacceptable. The similar grouping scheme is also adopted in [3], which aims at summarizing scene images. Zheng et al. [4] presented a web-scale landmark recognition system, in which the landmark visual models are built by using image matching and unsupervised clustering approach. The clustering algorithm is carried out on the matched local region graph. However, constructing the graph needs to perform image feature matching on all image pairs in the whole dataset, which also suffers the burden of heavy computation. Zeng et al. [5] proposed an annealing based algorithm which optimizes an objective function for grouping images. However, the method also relies on knowing the similarity of most of image pairs in advance.

A more efficient way for grouping images is presented by Li et al. [6]. They first divide the large amount of images into small groups by clustering low-dimensionally global GIST descriptors [7] and choose one representative image for each cluster as the iconic view. Then the iconic views (the subset of the whole image set) are organized by an iconic scene graph using keypoint based methods. Finally, a normalized graph cut algorithm is employed to partition iconic dataset into clusters. Chum et al. [8] proposed a web-scale image clustering method based on the randomized data mining. They use min-Hash algorithm to find out the cluster seeds and then the seed are used as the visual queries to obtain clusters.

The commonness of the above algorithms is that it is necessary to perform either ANN searching or spatially clustering on millions of feature descriptors to obtain the similarity between images. It makes them lack of efficiency or even impractical in application for the huge size database. To be different with them, the proposed approach just performs feature matching between some of the image pairs which are potentially content-related. The content-relevant measure can be scored by the bag-of-visual-words approach which has been proven successful in image retrieval applications [9,10,11]. These methods are inspired by text retrieval methods and support very efficient visual search even in large-scale dataset by using the inverted files scheme [9]. The retrieval results can be used as a kind of guidance to conduct the approach to perform image feature matching in the subset of all image pairs. The most significant contribution is that time complexity of our clustering algorithm is just $O(n)$.

The remainder of the paper is organized as follows. Section 2 presents the image clustering algorithm and in
Section 3 we discuss the experimental results and evaluate the performance. Finally, conclusions are drawn in Section 4.

II. OUR APPROACH

The core idea of our method is the query expansion process by which the image cluster is formed as transitive closures of sets of spatially related images. To improve the efficiency of the approach, other two key steps are introduced. One is to generate a guide table in which the images are ranked by the specific relevance scores by image retrieval methods. The guide table is used to further direct the subsequent feature matching in the query expansion process. Another one is to use affine invariant as the constraint to improve feature matching performance. Algorithm 1 gives the summary of our approach.

Algorithm 1: Scene Images Clustering

Input: n images in the collection C gathered from photo-sharing website
Output: Image clustering result \( G = \{ G_i | i = 1,2,...,m \} \)

1. Extract SIFT [12] features from all the images
2. Generate rank table by using the retrieval algorithm
2-1 Build a vocabulary tree [10] based on the training images.
2-2 Quantize feature vectors to their corresponding visual words by the GNP (Greedy N-Best Paths) [11] algorithm.
2-3 Take each image in turn to retrieve in the dataset. For the \( j \)-th (\( j=1,2,...,n \)) image, the retrieval results are ranked in descending order according to the relevance scores and organized in the \( j \)-th row of a so-called guide table (GT).
3. Generate image clusters using query expansion
3-1 Randomly select an unassigned image as a seed \( s \in C \). If such a seed can be found, create a new group \( G_s \), including \( s \) (i.e. \( s \in G_s \)) and goto step 3-2; otherwise, all the images have already been grouped and goto step 3-3.
3-2 Choose an element \( e \in G_s \) as a query (suppose \( e \) is the \( j \)-th image in the dataset and has not been chosen as a query before). Goto sub-step 3-2-1 to perform image feature matching using the \( j \)-th row of GT as the guidance. Repeat this step 3-2 until all the elements in \( G_s \) have been taken as the queries, and \( G_s \) is thus the one clustering result and goto step 3-1.
3-2-1 Suppose the images in the \( j \)-th row of GT are denoted as \( e_1, e_2,..., e_r \), perform feature matching between \( e \) and \( e_i \) in turn (make sure \( e_i \) is not the element of \( G_s \) ) by the affine invariant constraint. If the number of matched features \( N_m \) exceeds the pre-set threshold \( T \), add \( e_i \) to \( G_s \) and choose next candidate \( e_{i+1} \) to repeat this sub-step; otherwise, stop this sub-step and return back to step 3-2.
3-3 Discard the groups that contain less than \( N \) images and output the grouping results.

A. Image retrieval using bag-of-visual-words approaches

The step 2 in Algorithm 1 briefly lists the high-level description of generating the retrieval results table and more details are further explained in this section. The vocabulary tree is a kind of bag-of-visual-words algorithm and has been proven quite efficient in image retrieval applications [10, 11]. The vocabulary tree nodes are built by the hierarchical k-means clustering algorithm on SIFT feature vectors from the training data. In our experiment, a vocabulary tree with 10 branching factor and 5 levels is built using the whole dataset as the training data. Next, the GNP searching algorithm is employed to improve the quantization accuracy and further to enhance the retrieval performance, since it considers more candidates instead of one at each level of the tree. The high-dimensional feature vectors are thus quantized to visual words by GNP algorithm, and the words are organized by inverted file structure which keeps track of the number of times each visual word appears in each image. In the retrieval process, the relevance scores are calculated using the TF-IDF scheme [9]. The retrieval process runs very fast. In our experiment (the dataset size \( n=1249 \) ), it takes only about 20s to obtain the \( n \times n \) guide table.

Algorithm 2: Feature correspondence of the same visual word \( W \)

Input: \( F_{1i} \) and \( F_{2j} \), which denote the feature subsets in two images \( I_1 \) and \( I_2 \), where the features are quantized to the same visual word \( W \)
Output: \( p \) matched feature pairs

1. \( p_i = \min \{ | F_{1i} |, | F_{2j} | \} \)
2. Find the minimum distance among \( | F_{1i} | \times | F_{2j} | \) candidate distances and mark the corresponding features \( f_i \in F_{1i} \) and \( f_j \in F_{2j} \) as a correspondence feature pair.
3. Remove \( f_i \) from \( F_{1i} \) and \( f_j \) from \( F_{2j} \).
4. If \( F_{1i} \neq \emptyset \) and \( F_{2j} \neq \emptyset \), goto step 2; otherwise output the results.

B. Feature matching using the affine invariant constraint

The retrieval results can be taken as the guidance to indicate the potential image matching pairs that have more likely to be content related. The purpose to perform image feature matching is to obtain enhanced image similarity, since that the local features hold more information than their corresponding visual words, such as their positions in image. The enhanced relevance scores can help to reach the high image clustering performance.

In the feature matching phase, the initial matched features can be obtained by utilizing the inverted files. The features are considered matched to each other when they are quantized to the same visual word. Algorithm 2 shows the method that we choose the feature correspondence for one of the same visual words. It is worth noticing that
Algorithm 2 guarantees the one-to-one mapping between feature sets, which is important to the next step.

Next, the initial correspondence results should be further verified by the specific constraints in order to reach higher matching precision and thus provide the enhanced similarity measure. The most common used constraint is geometric verification, e.g., RANSAC-like algorithms. However, this process is quite time consuming and it is too strong for image clustering. In this paper, we propose a simple yet effective constraint - area ratio of triangles which is an invariant under the affine transformation [13]. As we know, the image transformation within multiple views can be approximately considered as affine transformation in most cases. Therefore, this constraint is theoretically effective. Algorithm 3 gives the details of rejecting outliers by the affine invariant constraint. As a result, the remaining correspondence feature pairs that pass the verification are considered as the matched ones and the number of the matched points is taken as the enhanced similarity measure between images.

Algorithm 3: Verification of the feature matching

**Input:** Initial matched feature pair set IPS of I₁ and I₂

**Output:** The refined matched feature pairs that pass the verification

1. Choose four matched feature pairs in sequence from IPS, which are denoted as A→a, B→b, C→c, D→d. Make sure any three of them are not collinear.
2. Verify the matching by Algorithm 4. If it returns TRUE, leave the four feature pairs in the IPS; otherwise, remove the four pairs from IPS and add them to a candidate pair set CPS.
3. If all the pairs in IPS are checked, goto step 4; otherwise, goto step 1 and choose the next four matched feature pairs.
4. Randomly choose three matched pairs from IPS, denoted as A→a, B→b, C→c, and choose one pair in IPS, denoted as D→d. Make sure any three of them are not collinear. Verify them by Algorithm 4. If it returns TRUE, add the pair D→d to IPS.
5. Repeat step 4 until all the pairs in CPS have been checked. The feature pairs in IPS are the final output results.

C. Complexity analysis

The computational complexity of our approach is composed of two parts, which are retrieval process and feature matching process. The first process is quite efficient due to the use of inverted file structure. As mentioned above, to obtain the whole guide table takes just about 20 seconds on our experimental dataset. Therefore, the time cost of the first part can be ignorable compared to the second one. The second part is the most time consuming since it performs matching between feature sets. The total time complexity (TC) can be calculated only including the second part by (1), where \( n \) is the number of images of \( i \)-th group, \( t \) denotes the average time cost for matching features between an image pair, and \( n \) is the total number of images in the dataset.

\[
TC = t \sum_{i=1}^{n} (1 + 2(n_i - 1)) = t \sum_{i=1}^{n} (2n_i - 1) = t(2n - m) \quad (1)
\]

As a result, the total time complexity of our approach is \( O(n) \) when \( n \leq m \). It is much lower than the previous approaches whose complexities are \( O(n^2) \).

In addition, the parameters \( T \) and \( N \) in Algorithm 1 are the pre-set thresholds, which can influence the clustering performance. In general, with the increase of \( T \), the average clustering precision (ACP) will increase; but the average clustering recall (ACR) will decrease. The threshold \( N \) determines the lower bound of the number of images per group. In our experiments, we find that the high clustering performance can be reached when \( T \) and \( N \) are set 20 and 4 respectively.

Algorithm 4: Area ratio constraint

**Input:** Four matched feature pairs which are A→a, B→b, C→c, D→d and any three of them are not collinear

**Output:** TRUE or FALSE to indicate whether they pass the constraint verification

1. Calculate the triangle areas \( S \) and the area ratio should keep invariant, i.e., \( S_{ABC} S_{BCD} S_{CDA} S_{DAB} = S_{DAB} S_{CDA} S_{BCD} S_{ABC} \).
2. Let \( V_1 = (S_{ABC}, S_{BCD}, S_{CDA}, S_{DAB}) \) and \( V_2 = (S_{ABC}, S_{BCD}, S_{DAB}, S_{CDA}) \). \( V_1 \) and \( V_2 \) are corresponding normalized ones respectively. Calculate the scalar product \( p \) of \( V_1 \) and \( V_2 \).
3. If \( p \geq 0.99 \), the four matched feature pairs satisfy the constraint and return TRUE; otherwise, output FALSE.

III. EXPERIMENTAL RESULTS

We have tested our method on the photograph dataset of Bell Tower, Xi’an city, China. The images are automatically downloaded from the Flickr.com website using keyword searching. The experiments run on a common PC: Intel(R) Pentium duel-core processor 2.0 GHz and 2 GB memory.

A. Clustering result and evaluation

The Bell Tower dataset has 1249 images and totally 867K SIFT features extracted from these images. We use all these features to train a vocabulary tree with 10 branching factor and 5 levels. The seed image of each cluster is shown in Figure 1 and six examples of cluster are shown in Figure 2. The clusters that have less than \( N=4 \) images are rejected by our approach, and the rejected images are considered as the contaminated ones. There are totally 819 images left after clustering, categorized into 22 groups. We judge the correctness of the clustering result manually by checking that whether the images in the same group are spatially overlapping. We also manually pick up the images that should be grouped to the clusters but are rejected falsely by our approach. Figure 3 lists the overall 24 images which are rejected but are indeed visually relevant to the certain clusters.
In order to evaluate the clustering performance, two evaluation indexes, average clustering precision (ACP) and average clustering recall (ACR), are defined in Eq.(2) and Eq.(3), where $m$ is the number of clusters, $tp_i$ and $fp_i$ denote the number of the true and false positives respectively, and $mi$ is the number of the true negatives.

$$ACP = \frac{1}{m} \sum_{i=1}^{m} p_i = \frac{1}{m} \sum_{i=1}^{m} \frac{tp_i}{tp_i + fp_i}$$  \hspace{1cm} (2)

$$ACR = \frac{1}{m} \sum_{i=1}^{m} r_i = \frac{1}{m} \sum_{i=1}^{m} \frac{tp_i}{tp_i + mi}$$  \hspace{1cm} (3)

The higher $ACP$ and $ACR$ are, the higher clustering performance is. Our approach can obtain as high as 98.9% of $ACP$ and 96.8% of $ACR$ on the Bell Tower dataset with $T=20$ and $N=4$, respectively.

### B. Constraint verification

Table I lists the feature correspondence results of the selected image of C1 shown in Figure 2 to verify the effectiveness of the proposed affine invariant constraint. These six images are added to C1 according to the matching of $(a,e)$, $(e,f)$, $(f,b)$, $(f,c)$ and $(c,d)$ respectively. The homography matrix between each image pair is estimated by RANSAC algorithm based on the initial matched features and is taken as the ground truth to verify the correctness of the matched features. From Table I, we can see that the proposed constraint can effectively reject outliers, though it may reject some inliers. The high precision of feature matching obtained by using the proposed constraint can provide more reliable similarity measure between images. Figure 4 shows an example of feature correspondence between the image pair $(e,f)$ in C1 of Figure 2. It can be observed that outliers are rejected by the proposed constraint.

TABLE I. **CORRESPONDENCE RESULTS WITHOUT AND WITH THE CONSTRAINT**

<table>
<thead>
<tr>
<th>Image pairs</th>
<th># Matched features (True num, Precision)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a,e)</td>
<td>27 (16, 59%) Without the constraint</td>
</tr>
<tr>
<td>(c,f)</td>
<td>34 (23, 68%) Without the constraint</td>
</tr>
<tr>
<td>(f,b)</td>
<td>28 (16, 57%) Without the constraint</td>
</tr>
<tr>
<td>(f,c)</td>
<td>49 (35, 71%) Without the constraint</td>
</tr>
<tr>
<td>(c,d)</td>
<td>48 (27, 56%) Without the constraint</td>
</tr>
</tbody>
</table>

IV. CONCLUSION

We present an efficient image clustering approach for web collections. The proposed method firstly generates the guide table that ranked the retrieval results of each image using the bag-of-visual-words methods. Then images are grouped by query expansion techniques under the direction of the guide table. On the feature matching stage, an effective constraint is further proposed based on the affine invariant. The constraint can reject the outliers of feature correspondence without using any RANSAC-like estimation algorithm. The computation complexity of our algorithm is only linear with the number of images of the dataset. Experimental results have demonstrated the high clustering performance of our approach. The output of our system can provide user an efficient way in browsing the large-scale image collections. In addition, the clustering results can be used for the 3D reconstruction of the scene. It can be done by the well-known structure-from-motion methods and this is left in our future work.
V. ACKNOWLEDGEMENT

The work is supported by National Natural Science Fund (60873085), National Hi-Tech Development Program (2007AA01Z314), P. R. China. The first and second authors also thank the funds supported by China Scholarship Council.

REFERENCES


<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2. Selected images from selected clusters. The images within red rectangles denote the false positives which are wrongly categorized.
Figure 3. The true negatives supposedly assigned to the corresponding groups denoted by the text upon images.