Effective Scene Matching with Local Feature Representatives

Shugao Ma[1] Weiqiang Wang[1,2] Qingming Huang[1,2] Shuqiang Jiang[2] Wen Gao[2,3]
[1]*Graduate University of Chinese Academy of Sciences, Beijing, China*
[2]*Key Lab of Intell. Info. Process., Inst. of Comput. Tech., Chin. Acad. of Sci., Beijing, China*
[3]*Institute of Digital Media, Peking University, Beijing, China*
E-mail: {sgma, wqwang, qmhuang, sqjiang, wgao}@jdl.ac.cn

Abstract

Scene matching measures the similarity of scenes in photos and is of central importance in applications where we have to properly organize large amount of digital photos by scene categories. In this paper, we present a novel scene matching method using local features representatives. For a given image, its scene is compactly represented as a set of cluster centers, called local feature representatives, where the clusters are obtained using the affinity propagation (AP) algorithm to aggregate local features according to their spatial closeness and appearance similarity. The similarity of scenes in two images is then measured by a modified Earth Mover Distance (EMD) between their corresponding sets of local feature representatives. Empirical experiments on real world photos shows that our method is comparable to the state-of-the-arts [1][2].

1. Introduction

Scene matching aims to identify whether two images are taken in the same scene. Compared with the issue of image matching, two matched images can involve different foreground objects at different positions in which the viewpoints and lighting changes are also permitted to some extent (see Fig.1). An effective solution to this issue is very useful. For example, thousands of digital photos can be organized by scene categories, and people can easily select their favorite photos from the same scenes for printing. In recent years, the use of local invariant features are very popular in image matching [3, 4, 5, 6], since they are robust to common geometric, photometric changes, and partial occlusions, which are also preferable for scene matching. Some researchers have applied the local invariant features to the task of scene matching or classification. For instance, [3, 7] presented a voting scheme in which each image in databases obtains a certain number of votes depending on the number of matched local features against an query image. The scheme cannot discriminate among images with different distributions of similar local structures. Recently Grauman et al. [1] proposed an image matching approach called the approximate EMD (AEMD) and applied it to the scene recognition task. Their method measures the similarity between images by the L_1 distance of two sets of local features in the embedded Earth Mover Distance (EMD) [8] space. Their embedding map essentially evaluates a histogram of local features, so the potential quantization errors introduced by binning possibly harm the matching performance of the AEMD method.

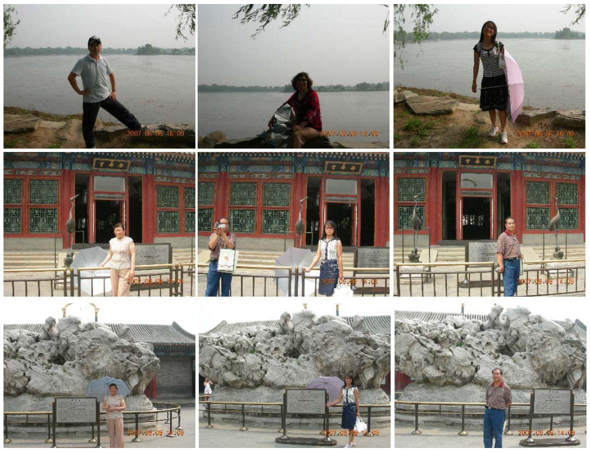


Fig. 1. Sample images of the same scene

In this paper, we present a new scene matching method based on the representatives of local features. First our method utilizes the affinity propagation

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

algorithm [9] to cluster local features according to their spatial closeness and appearance similarity. Then an image is represented by a compact set of cluster centers, called local feature representatives. Each local feature representative is associated with a weight which reflects the size of the cluster it represents. Finally we measure the similarity of scenes in two images by a modified Earth Mover Distance (EMD) between their sets of local feature representatives. Compared with the AEMD method, our quantization of local features is more adaptive to their distributions.

The paper is organized as follows. In section 2, we present the details of our scene matching method. Section 3 reports the results of evaluation experiments and section 4 concludes the paper.

2. Our scene matching method

2.1. Affinity propagation

Frey and Dueck [9] presented a new clustering algorithm called "affinity propagation" (AP). It can effectively find the exemplars of vast data points through recursively transmitting real-valued messages among the data points. After the affinity propagation, each exemplar and the data points it represents form a cluster. We summarize the AP algorithm in Fig. 2.

```
Affinity Propagation
Input: affinity matrix of data points $s(i,k)$
Output: a set of exemplars and the clusters
Step 1. Set $a(i,k)=0$ for all $i,k$
Step 2. Update responsibilities:
    $r(i,k) \leftarrow s(i,k) - \max_{k' \neq k}\{a(i,k') + s(i,k')\}$
Step 3. Update availabilities
    $a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{i' \notin \{i,k\}} \max\{0, r(i',k)\}\}$
    $a(k,k) \leftarrow \sum_{i' \neq k} \max\{0, r(i',k)\}$
Step 4. Decide the exemplars
    $e_i = \arg\max_k \left(a(i,k) + r(i,k)\right)$
Repeat step 2 to 4 for a fixed number of iterations
or when the decisions for exemplars stay constant
for a certain number of iterations
```

Fig. 2. Affinity propagation Algorithm

In the AP algorithm, $s(i,k)$ is a similarity value between data points i and k . $s(k,k)$ is specified by the users and represents a priori preference that data point k should be chosen as an exemplar. Two kinds of messages are passed between data points: responsibility $r(i,k)$ and availability $a(i,k)$. The responsibility $r(i,k)$ sent from data point i to candidate exemplar k represents how well data point k is suitable for being the exemplar of data point i ; the availability $a(i,k)$ sent from candidate exemplar k to data point i represents how appropriate it is for data point i to choose data point k as its exemplar. e_i represents the index of the data point chosen as the exemplar for data point i at the end of each iteration. An advantage of affinity propagation is that instead of requiring a predefined number of clusters and the initial cluster centers like the k-centers algorithm, it automatically identifies the number of exemplars by preferences values (i.e. $s(k,k)$) and the message-passing procedure. The choice of preference values reflects the expectation of the number of clusters. The median of all the similarities is usually a good choice if no a priori knowledge about the data points' distribution is available.

2.2. Generating local feature representatives

We exploit the method presented in [3] to extract local invariant features from images. Then a set of SIFT features are obtained. To capture the distribution properties of local features in the spatial space and the feature space, we apply a two-pass AP algorithm to cluster the SIFT features according to their spatial closeness and appearance similarity. Since the distribution of the SIFT features are unknown beforehand, we have no pre-knowledge about the appropriate number of clusters, so we aim to make the AP algorithm to automatically generate a moderate number of clusters. First, all local features are clustered according to their spatial closeness. The similarity between two local features is measured by the Euclidean distance of their spatial coordinates in an image, and the preference values $s(k,k)$ are set to the mean of all the spatial distances. After this step, local features geometrically close form a cluster. Then, the local features in each spatial cluster are further clustered based on their appearance similarities. We choose the Euclidean distance in feature space to measure the appearance similarity of two local features, and the preference values $s(k,k)$ are set to the mean of the corresponding distances in each spatial cluster. The second-pass clustering runs very fast since there are a small number of data points in each spatial cluster. For each generated cluster, a representative is required to be chosen for it. Although the exemplars identified by the AP algorithm can be used, our method uses the mean of the feature vectors in each cluster instead, since some experiments show this choice leads to better performance. Finally each representative $c_i, i = 1,...,l$ is associated with a weight $w_i = n_i / \sum_{i=1}^{l} n_i$, where n_i denotes the number of local features in the *i*th cluster and l is the total number of clusters.

2.3 Scene dissimilarity

Through the two-pass AP clustering, each image is represented as a set of feature clusters $\{(c_i, w_i)\}$. Rubner *et al.* [8] had demonstrated that the Earth Mover Distance (EMD) is a good choice to measure the difference between two variable-length representations of distributions. The EMD between two images $P = \{(p_i, w_{p_i}) \mid i = 1,2,...,m\}$ and $Q = \{(q_j, w_{q_j}) \mid i = 1,2,...,n\}$ can be expressed as

$$\begin{aligned} & EMD(P,Q) = \min_{f_{ij}} \sum_i \sum_j f_{ij} d(p_i, q_j) \\ s.t.\ & \sum_i \sum_j f_{ij} = \min(\sum_i w_{p_i}, \sum_j w_{q_j}), \\ & \sum_j f_{ij} \le w_{p_i},\ \sum_i f_{ij} \le w_{q_j}, \\ & f_{ij} \ge 0, \qquad i = 1,...,m, j = 1,...n \end{aligned} \quad , \qquad (1)$$

where $d(p_i, q_j)$ called ground distance is a distance measure between two feature vectors p_i and q_j. Apparently it is a linear programming problem and efficient algorithms exist to find the flow $F = [f_{ij}]$ to minimize the overall cost. In the context of our problem, the total flow is 1. We use the following measure

$$d(p_i, q_j) = 1 - \sum_{k=1}^{D} \min(p_i^k, q_j^k) \qquad (2)$$

for the ground distance to characterize the dissimilarity of two local features, where p_i^k and q_j^k denote the *k*th components of the *D*-dimensional feature vectors p_i and q_j. Instead of directly using $d(p_i, q_j)$, we devise a non-linear scaling technique to boost discriminative power of the original EMD measure, called the scaling EMD (SEMD), in which the ground distances between two features are shortened or lengthened according to their original ground distance in Eq.(2), i.e.,

$$\begin{aligned} & d'(p_i, q_j) = \begin{cases} 0, & d(p_i, q_j) \in (0, \alpha) \\ d(p_i, q_j), & d(p_i, q_j) \in (\alpha, \beta) \\ d(p_i, q_j) * a, & d(p_i, q_j) \in (\beta, 1) \end{cases} \\ & s.t.\ \ 0 < \alpha < \beta < 1,\ \ a > 1 \end{aligned} \quad . \qquad (3)$$

If two features p_i and q_j are matched very well, the distance $d(p_i, q_j)$ is a very small value, and then scaling it into zero signifies that a larger flow allocation f_{ij} between them is expected in the linear programming to minimize the total cost. On the contrary, for two features with a large distance, the further magnified distance can better suppress the flow allocation between them in linear optimization. Thus, SEMD can better implement the matching of two sets of features. We empirically set $\alpha = 0.05$, $\beta = 0.3$ and $a = 10$ in our system.

3. Evaluation experiments

In this section, we compare our algorithm with the approximate EMD (AEMD) algorithm [1] and the algorithm presented in [2] in which the K-means algorithm is used to extract local feature signatures and then the EMD is used to measure texture dissimilarity (KEMD). Although Lazebnik *et al.* [2] apply it in the texture classification task, we think it is also applicable in scene matching. The dataset in our evaluation consists of 170 real world photos. We manually group them into 26 groups and each group contains 5~8 photos of a same scene. Since we notice that low resolution images are sufficient for the scene matching task, our system resizes the input photos to about 300 by 200 pixels to speed up the matching process, and the experiments show the matching performances are not degraded in evidence.

For each algorithm, we apply it to calculate the scene dissimilarities of all image pairs in the dataset, and then we evaluate its matching performance by the normalized average rank (NAR) [1], as well as precision-recall. For an image I in the dataset, we sort the other images in ascending order according to their dissimilarities with the image I, and the NAR for the image I is defined as:

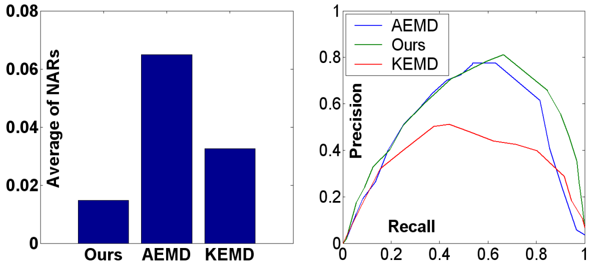
$$\overline{R}_I = \frac{1}{NN_R} \left(\sum_{k=1}^{N_R} R_k - \frac{N_R(N_R - 1)}{2} \right), \qquad (4)$$

where N is the number of the images in the dataset minus 1, N_R is the number of relevant images which belong to the same image group with I, and R_k is the rank of the *k*th relevant image after sorting. If the matching performance is perfect in which case the relevant images are the top N_R images after sorting, the corresponding NAR is 0. Apparently the metric NAR becomes larger as the matching performance becomes worse. For each algorithm, we use the average of the NARs of all images in the dataset (ANAR) to measure its overall performance. The lower value means the better retrieval matching performance. Another evaluation measure we employed is the precision-recall. For a given dissimilarity threshold δ, we define the precision and recall for an image I as:

$$p_I = \frac{N_{RS}}{N_S}, \qquad r_I = \frac{N_{RS}}{N_R} \tag{5}$$

where N_S is the number of images whose dissimilarity values with I are smaller than the threshold δ, N_{RS} is the number of relevant images among the N_S images and N_R is the total number of relevant images. We exploit the average of the precisions and recalls for each image as another matching performance measure of the three algorithms.

In comparison experiments, we replace the PCA-SIFT [10] in the original AEMD method with the SIFT descriptor for the sake of fair comparison, and set the number of clusters in the KEMD method as 20 to make it obtain the best experimental results. As shown in Fig. 3 (a), the ANAR is 0.015 for our algorithm, 0.065 for AEMD and 0.032 for KEMD, so our method has a better performance. The precision-recall curves for the three algorithms in Fig. 3 (b) also show the effectiveness of our method.



(a) (b)

Fig. 3. Experiment results. (a) Comparison of the average of NARs. (b) Precision-recall curves for AEMD, KEMD and our method.

4. Conclusion

We propose a new approach to address the scene matching problem using local feature representatives. In particular, by incorporating both AP to cluster local features by spatial closeness and similar appearance, and the proposed SEMD technique, our approach performs favorably on a real world photos dataset against the state-of-the-arts: the AEMD [1] and the KEMD [2] methods.

Acknowledgements

We are grateful to Kristen Grauman for her provision of the AEMD code. The work is supported by National Key Technologies R&D Program under Grant 2006BAH02A24-2 and the research start-up fund of GUCAS.

References

[1] K.Grauman and T. Darrell. Efficient Image Matching with Distributions of Local Invariant Features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pages 627-634, 2005.

[2] S. Lazebnik, C. Schmid and J. Ponce. A Sparse Texture Representation Using Local Affine Regions. IEEE transactions on Pattern Analysis and Machine Intelligence, 27(8): 1265-1278, Aug. 2005.

[3] D. G. Lowe. Distinctive Image Features from Scale Invariant Keypoints. *International Journal of Computer Vision*, 60(2): 91-110, 2004

[4] K. Mikolajczyk and C. Schmid. Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1): 63-86, 2004

[5] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust Wide-baseline Stereo from Maximally Stable Extremal Regions. In *Proceedings of the British Machine Vision Conference*, pages 384–393, 2002.

[6] T. Tuytelaars and L. Van Gool. Matching Widely Separated Views based on Affine Invariant Regions. *International Journal of Computer Vision*, 59(1):61–85, 2005.

[7] F. Shaffalitzky and A. Zisserman. Automated Scene Matching in Movies. In *Proceedings,Challenge of Image and Video Retrieval*, pages 186-197, July 2002.

[8] Y. Rubner, C. Tomasi, and L. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2): 99–121, 2000.

[9] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, vol. 315, pages 972—976, 2007.

[10] Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *Proceedings of IEEE Conference on Computer vision and Pattern Recognition,* vol. 2, pages 506-513, 2004.