# Spectral Images and Features Co-Clustering with Application to Content-based Image Retrieval

Jian Guan and Guoping Qiu

School of Computer Science and Information Technology The University of Nottingham, United Kingdom

Abstract—In this paper, we present a spectral graph partitioning method for the co-clustering of images and features. We present experimental results, which show that spectral coclustering has computational advantages over traditional kmeans algorithm, especially when the dimensionalities of feature vectors are high. In the context of image clustering, we also show that spectral co-clustering gives better performances. We advocate that the images and features co-clustering framework offers new opportunities for developing advanced image database management technology and illustrate a possible scheme for exploiting the co-clustering results for developing a novel content-based image retrieval method.

Keywords—spectral graph partitioning, co-clustering, image database, content-based image retrieval

#### I. INTRODUCTION

Managing increasingly large image repositories is a technically challenging problem, which in recent years has attracted extensive research interests. Data clustering is standard technique that is also useful for image database management. To cluster images into visually homogeneous groups, images are represented by their visual features such as colour histogram and texture descriptors. However, as has been pointed out in [2] traditional clustering methods cannot identify which feature(s) is more important to a certain image (cluster). For example, when colour histogram is used to represent images for clustering, information about which colour bins are more strongly associated with a particular cluster is not known. Surprisingly, there is little existing research which tries to answer following questions: Why are certain images grouped together? What features do the images fall in the same cluster have in common? To be able to answer these questions will be helpful in gaining deeper understanding of image database management problems and in developing better solutions.

Base on the duality of images and their features, we attempt to answer these questions and to develop advanced content-based image retrieval (CBIR) technology through the idea of images and features co-clustering [2]. We use quantized image patches as image features [1] and use spectral bipartite graph partitioning [3] to perform coclustering. We will first present results of spectral images and features co-clustering and then present a scheme for exploiting the results of image and feature co-clustering for developing advanced content-based image technology.

Xiang-Yang Xue Department of Computer Science and Engineering Fudan University, Shanghai, China

# II. BIPARTITE GRAPH MODEL AND IMAGES AND FEATURES CO-CLUSTERING

Many works in a variety of discipline have used graph model to do clustering, for example, Pothen et al used it for sparse matrix partitioning [4], Hagen and Kahng used it for circuit partitioning [7], Shi and Malik used it for image segmentation [6], Dillon used it for word-document clustering [3] and Qiu used it for CBIR [2]. Here we use a model similar to the one proposed in [2], where images and their representation histograms are modeled as a bipartite graph.

An image descriptor is usually in the form of a feature vector, we call each vector entry a feature prototype. Given an image set I and a set of feature prototypes P, treating each image and each feature prototype as a vertex, we can form a weighted bipartite graph  $G = \langle I, P, W \rangle$ , where  $W = \{w_{ij}\}, w_{ij}$ is the weight connecting the ith image and the ith feature prototype vertices. In this context, images and features coclustering refers to dividing images and their feature prototypes into associated groups where each group of images is associated with certain feature prototypes, which can be translated to a graph partitioning problem [2].

# III. SPECTRAL CO-CLUSTERING

We use a term defined in graph theory: Consider a special case for graph partitioning, Graph Bisection, which divide a

case for graph partitioning, *Graph Bisection*, which divide graph into 2 subsets of vertices, 
$$V_1$$
 and  $V_2$ ,
$$cut(I_1, I_2, P_1, P_2) = \sum_{i \in I_1, j \in P_2} w_{ij} + \sum_{i \in I_2, j \in P_1} w_{ij} \tag{1}$$

where  $I_1 \cup P_1 = V_1$ ,  $I_2 \cup P_2 = V_2$ ,  $I_1 \cap I_2 = \varphi$ ,  $I_1 \cup I_2 = I$ ,  $P_1 \cap I_2 = \emptyset$  $P_2 = \varphi$ ,  $P_1 \cup P_2 = P$ . Our goal is to minimize the association between two different partitions. With this definition, it is clear that the problem is equivalent to the MinCut in graph theory. As many others e.g., [3], have pointed out, MinCut could produce unbalanced partitioning result. Statistically, when the image database is very large, we would prefer that image numbers in different clusters are close and, so are the feature numbers. Taking the size of cluster into consideration, Ratio Cut [7] and Normalized Cut [6] are more suitable. In this paper, we use the criteria of Normalized Cut for its robustness [9].

We further refine the definition for the graph and introduce weight for each vertex.  $G = \langle I, P, W, WI, WP \rangle$ , where  $WI=\{wi\}$  is the weight for image vertices and  $WP=\{wp\}$  is the weight for feature prototype vertices. We choose the weight for each vertex as the sum of the weights of its indent edges. It is important to note that, most image features, e.g. the features used in this paper, color histogram, CPAM histogram and Spatial CPAM, have already

introduced the mechanism of normalization, thus weights for all image vertices are the same, while the values of wp's may vary. This means all images are treated equally, while some features are more important than others, which is exactly what we want.

Based on Normalized Cut, we can write the cost function

$$J = \frac{cut(V_1, V_2)}{weight(V_1)} + \frac{cut(V_1, V_2)}{weight(V_2)} = \frac{cut(I_1, I_2, P_1, P_2)}{\sum_{k \in I_1} wi_k + \sum_{l \in P_1} wp_l} + \frac{cut(I_1, I_2, P_1, P_2)}{\sum_{k \in I_2} wi_k + \sum_{l \in P_2} wp_l} \tag{2}$$

Clearly, by minimize J, we will minimize inter-cluster connection as well as maintaining balance between the 2 clusters. References [3] and [5] have given an approximate solution to Equation (2), which naturally leads to SVD. Let  $D_1$  and  $D_2$  be diagonal matrices where  $D_1(k, k)=wi_k$ ,  $D_2(l, k)=wi_k$ *l*)= $wp_l$ . We have following algorithm for images and features co-clustering:

- Given an image set I and their feature vectors  $\{d_i\}$ , form a graph  $G = \langle I, P, W, WI, WP \rangle$ , where  $W = [d_1; d_2; d_3; ...d_n]$ . Compute  $D_1$  and  $D_2$ , let  $M = D_1^{-1/2} W D_2^{-1/2}$ .
- 2.
- Do SVD on M, let  $u_2$  and  $v_2$  be the left and right singular vectors corresponding to the second largest singular value.
- Form the partition vector  $p = \begin{bmatrix} D_1^{-1/2} u_2 \\ D_2^{-1/2} v_2 \end{bmatrix}$ , each entry in p is

sequentially associated with a vertex, an image or a feature prototype.

Find a cut point c which divides p into 2 parts,  $V_1 = \{i \mid p_i \le c\}$ and  $V_2 = \{i \mid p_i > c\}$ , such that

$$\arg\min(\sum_{i,j\in V_1} (p_i - p_j)^2 + \sum_{i,j\in V_2} (p_i - p_j)^2)$$

whereby the images and feature prototypes are divided into 2 clusters as well.

If necessary, 2 sub-graphs can be formed based on the 2 subsets of images and feature prototypes respectively, run the algorithm recursively to obtain desired partitioning.

# IV. CLUSTERING PERFORMANCE

Above all, spectral co-clustering itself is a clustering technique. We have performed extensive experiments against a classical clustering method, k-means, on 2 large image datasets. One dataset consists of 12710 color photos from the commercially available Corel Photo CDs and the other one is the Columbia Object Image Library (COIL) with 7200 images [11]. Features used in the experiments include simple joint RGB color histogram (64 bins) and CPAM histogram (384 bins, 64 chromatic and 64 achromatic appearance prototypes at 3 resolutions, some examples of the prototypes are shown in Fig. 1). Details on how to build these features can be found in [10] and [2].



**Fig. 1:** Examples of chromatic and achromatic feature prototypes

For a dataset of K images and each represented by L feature prototypes, to perform spectral co-cluster, we only need to calculate a few singular vectors of a  $K \times L$  matrix, hence the method is computationally very efficient. Also, the

algorithm needs only to store a  $K \times L$  matrix [3] instead of (K+L) x (K+L) matrix as in normalized cut [6]. Our implementation is based on Lanczos algorithm with partial reorthogonalization [8]. Fig. 2 shows a comparison of computational times of spectral co-clustering method and the k-means method. It is seen that for low dimensional features, the two methods have similar computational times, however, when the feature dimension is high and the number of cluster is large, the spectral co-cluster method converges much faster. Another advantage of the spectral method is that it is deterministic, unlike k-means, which greatly depends on the initial conditions.

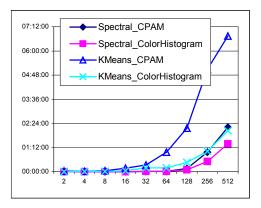


Fig. 2: Comparison of spectral co-clustering and k-means using CPAM Histogram (feature vector length: 384) and color histogram (feature vector length: 64). The x-axis is the number of clusters and the y-axis is time the algorithms took to converge on an Intel Pentium IV 1.8GHz PC with 256M RAM.

How to measure the clustering quality is an open issue. For the Corel colour photo dataset, although the 12710 photos are from 128 hand-label categories with various themes, some images in different categories actually contain very similar contents while some images in the same category looks rather different. A simple (and informal) way to evaluate the clustering quality of this dataset is via subjective judgement. If the visual appearance of the images in each cluster is homogeneous, we can say the method gives good clustering results. Fig. 3 shows two clusters obtained by the spectral coclustering method and k-means method, where it is seen that although both clusters contain homogeneous images, the spectral co-clustering cluster contains more images (which obviously belonging to the same category), this means that the k-means algorithm has split similar images into different groups (an undesirable result). Fig. 4 shows another two clusters obtained by the spectral co-clustering and k-means methods respectively. In this example, the cluster of the coclustering contains 5 images and 4 of them contain the same object; in contrast, the cluster obtained by k-means containing the same object has 18 images, of which there are no less than 6 different objects/scenes. At an informal level (which may be the best one can do to judge the clustering quality of data of this nature), spectral co-clustering seems to give more sensible solutions.



**Fig. 3**: Examples of clustering results. Database: 12710 color photos. Features: 64-bin color histogram. Cluster: 128. Left: one of the clusters from the spectral co-clustering technique. Right: a similar cluster from k-means clustering. Note that images in both groups contain homogeneous colors, the cluster obtained by spectral co-clustering contained more images which should indicate a better performance.



**Fig. 4**: Examples of clustering results. Database: 12710 color photos. Features: 384-bin CPAM histogram. Cluster: 1024. Top: one of the clusters from the spectral co-clustering technique. Bottom: a similar cluster from k-means clustering. Note that images in the cluster obtained by spectral co-clustering contained images that are more homogeneous than those in a similar group obtained by k-means.

In order to evaluate the clustering quality more objectively, we have performed clustering on the COIL dataset [11]. This dataset contains images of 100 different objects and for each object there are 72 images. Ideally, we would like the clustering algorithms to automatically group images of the same object into the same cluster. Therefore, we assume that the number of objects (hence the number of clusters) is known. For each cluster, we count the percentage of images belonging to the same object. Obviously, the higher the percentage, the more homogeneous is the cluster, and the better the performance. Several test subsets are built on COIL-100, each with different numbers of objects, e.g. COIL20 is a test set with 20 objects. Table 1 shows the results of the spectral co-clustering method against k-means method. It is seen that, on average, the spectral co-clustering gives better performances.

**Table 1.** Average clustering results of spectral co-clustering and k-means using joint RGB colour histogram (64 bins) as feature

	COIL20	COIL40	COIL60	COIL80	COIL100
Spectral	91.36%	94.32%	92.42%	90.57%	92.00%
K-Means	90.00%	91.14%	86.21%	88.07%	83.27%

#### V. CO-CLUSTERING FOR CONTENT-BASED IMAGE RETRIEVAL

By co-clustering, we have simultaneously grouped images and features together. We can now identify that certain features are more important for a certain set of images. This is unlike most previous methods where each feature prototype is treated as equally important for every image, and the importance of a given feature prototype to a given image is unknown. It is generally agreed that to associate meaningful semantic meanings to low-level features is difficult. Through co-clustering, we have automatically associated certain

features with certain images. Since the meaning of images is more apparent to humans, we can (to some extent) derive the "semantic" meaning for these features. This is exact the case in linguistic where "semantic" is possible only within certain context. For example, the co-clustering result of Fig. 3 and the joint RGB color histogram bin [R(0-64), G(0-64), B(192-255)] has been grouped together. Without the images, what can be said about this bin is that it is some kind of 'bluish' color. Now, within the context of this image cluster, we can say this bin indicates blue sky and ocean. In addition, the set of feature prototypes will give us clues about the images grouped with them, which could potentially provide a more effective way to index and search image database. Base on above simple reasoning, we believe that the images and co-clustering framework offers many opportunities for developing better and more effective technologies for content-based image retrieval. In this section, we present an example for such a potential.

# A. Spatial CPAM

Note that in previous sections, the features did not contain spatial information. In many cases, image contents may be better characterized by also recording the spatial relations of the feature prototypes. Here we introduce a new image content descriptor termed Spatial CPAM (SCPAM). Given an image, we first zoom it out (reduce it) to a certain small size. Humans can recognize images of objects at very coarse resolutions or small image sizes such as those printed in Figures 3 and 4. Philosophically speaking, if humans can recognize coarse resolution (small size) images, then for computer recognition, it is also only necessary to represent images at a coarse resolution. After reducing the image, we divide it into  $m \times n$  blocks. For each block, we can use the algorithm of [1] to associated it with 2 CPAM feature prototypes (see Fig. 1), 1 for achromatic and 1 for chromatic information respectively. In this way, we define a 3dimensional binary image descriptor S as follow: S(x, y, i) = 1and S(x, y, j+N) = 1 iff the block in x column of y row is associated with the ith achromatic CPAM prototype and the jth chromatic CPAM prototype, where N is the total number of achromatic CPAM prototypes; otherwise  $S(x, y, i) = 0, \forall x$ , v, i.

Another purpose of introducing SCPAM is to facilitate the visualization of the association of features and images in the same cluster. Note that the SCPAM descriptor S contains all necessary information to reconstruct the image with the CPAM feature prototypes. An example image constructed with 64 achromatic and 64 chromatic CPAM feature prototypes is shown in Fig. 5. Note in the Figure, although it is a distorted version of the original, it is still recognizable. This property of SCPAM may be helpful in developing new content-based image retrieval methods.





**Fig. 5:** Left: Original image. Right: Reconstructed image using CPAM appearance prototypes.

# B. Co-Clustering SCPAM Features and Images

To perform co-clustering use the SCPAM descriptor, the 3-dimensional matrix S can be collapsed to a vector, say, in a column first manner. Note that in this scheme, the feature prototype set P consists of  $m \times n$  copies of CPAM feature prototypes. Use the spectral graph partitioning method described in previous sections, we can co-cluster the features and images.

For each cluster of images obtained by co-clustering, it is associated with certain blocks of the images, and of course those blocks are encoded by certain CPAM feature prototypes. Using only those blocks that are associated with a given cluster of images, we can visualize the important image parts for that cluster of images, as illustrated in Fig. 6. In this example, each image is resized to 16 (4 x 4) blocks of 4 x 4 pixels. Each block is then encoded using 64 achromatic and 64 chromatic CPAM prototypes, thus, a 2048 dimensional binary SCPAM vector represents each image. For each cluster of images, we draw those blocks that are associated with the cluster on the image on the left, which we call a cluster visualization image. Note that the same block of the visualization images may be associated with different CPAM feature prototypes; we simply average them for visualization purpose.

From Fig. 6, it is seen that the visualization image reflects the common attributes of the objects in the images within the cluster. This demonstrates that our features and images co-clustering technique has successfully associated important and meaningful features prototypes with images, and from this association, we can infer the predominant object shape of the images. This example demonstrates that with co-clustering, we have more information available to offer more flexibility for developing image database management technology. How to fully exploit the extra information most effectively is a topic that has never been explored.



**Fig. 6**: Left column: cluster visualization image where only image blocks associated with the image cluster (on the right) are displayed. Right column: image clusters obtained from co-clustering.

# C. Use Co-Clustering Results for Image Retrieval

One possible way to exploit the co-clustering results for image database management is schematically illustrated in Fig. 7. From Fig. 6, we see that it is possible to use those feature prototypes associated with a given cluster to construct a visualization image of the cluster. For each cluster, we can construct its associated visualization image. All visualization images then form a cluster object palette, which can be incorporated into the graphical user interface (GUI) of an image database management system. Note that each item in

the palette is associated with a cluster of images that contain objects similar to the visualization image and at a similar spatial position within the image. Such visual cues in the GUI should be helpful to users looking for their desired images.

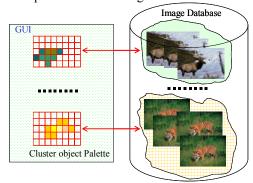


Fig. 7: One possible ways to exploit co-clustering results for developing image database management technology.

#### VI. CONCLUDING REMARKS

In this paper, we have first introduced spectral graph partitioning for the co-clustering of images and features. We have demonstrated that spectral co-clustering has a computational advantage over traditional k-means algorithm, especially when the feature vectors are of high dimensional. We have presented informal subjective and formal objective comparisons of image clustering results of spectral co-clustering and k-means and shown that spectral co-clustering has better performances. We have illustrated that the images and features co-clustering framework may offer new opportunities for developing advanced image database management technology and presented a possible scheme for exploiting co-clustering results for content-based image retrieval.

**Acknowledgment**: This work is partially supported by a University of Nottingham Research in or on Asia Grant.

# REFERENCES

- [1] G. Qiu, "Indexing chromatic and achromatic patterns for content-based image retrieval", Pattern Recognition, vol. 35, pp. 1675 1686, August, 2002
- [2] G. Qiu, "Image and feature co-clustering", ICPR 2004, pp. 991-994
- [3] I. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning", KDD 2001, pp. 269 274
- [4] A. Pothen, H. Simon and K. Liou, "Partitioning Sparse Matrices with Eigenvectors of Graphs", SIAM J. Matrix Anal. Appl., vol.11. no. 3, pp.430-452, July 1990
- [5] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. "Bipartite graph partitioning and data clustering", CIKM 2001, pp 25-32
- [6] J. shi and J.Malik, "Normalized cuts and image segmentation", CVPR 1997, pp. 731-737
- [7] L. Hagen and A. B. Kahng, "New Spectral Methods for Ratio Cut Partitioning and Clustering", IEEE Trans. on CAD 11(9), pp. 1074-1085, September 1992.
- [8] H. Simon, "The Lanczos algorithm with partial reorthogonalization", Math. Comp. 42 (1984), no. 165, pp. 115--142.
- [9] Y. Weiss, "Segmentation using Eigenvectors: A Unifying View", ICCV 1999, pp. 975-982
- [10] M. Swain and D. Ballard, "Colour Indexing", International Journal of Computer Vision, Vol. 7, 1991, pp. 11-3
- [11] S. A. Nene, S. K. Nayar and H. Murase, "Columbia Object Image Library (COIL-100)", Technical Report CUCS-006-96, Feb. 1996