

# WEB IMAGE CLUSTERING

Maha El Choubassi<sup>1</sup>, Ara V. Nefian<sup>2</sup>, Igor Kozintsev<sup>2</sup>, Jean-Yves Bouguet<sup>2</sup> and Yi Wu<sup>2</sup>

<sup>1</sup>University of Illinois at Urbana Champaign  
Image Formation and Processing Group, Urbana, IL

<sup>2</sup> Intel Corporation  
Application Research Labs, Santa Clara, CA

## ABSTRACT

While image clustering has many important applications ranging from personal to web image management, its use is often limited by the difficulty of extracting reliable semantics from low level image features. The image clusters can be improved by using features extracted from image regions rather than the whole image. Region segmentation can be improved in turn, by considering all images within the same cluster rather than segmenting each image independently. This observation leads to the unified Bayesian framework for image clustering and segmentation presented in this paper. The experimental results, reported using several types of visual feature extractors on a database of web documents containing over 6000 images, illustrates a significant improvement over existing techniques.

*Index Terms*— clustering, image segmentation.

## 1. INTRODUCTION

With the rapid increase in rich multimedia documents containing images, the need for clustering these documents becomes increasingly more important. In particular, clustering the results of a web image search can present the user with several semantic categories of the search and remove some of the unrepresentative results. There are several approaches to structure the retrieved web image results. Webseek [1] uses both text and image content to categorize web images based on a manually set taxonomy. In [2], the authors describe a hierarchical clustering system of general web images that uses the visual features to refine the clusters obtained using text features. In [3], the authors use Google's image search to learn object categories. Intra-class variations due to translation and scaling are taken into account by extending the probabilistic Latent Semantic Analysis technique (pLSA) of [4] into Translation and Scale invariant pLSA (TSI-pLSA). In [5], the authors describe a method for learning appearance-based models of the object classes in a supervised manner. Each of the nine objects in the database is represented by a histogram of words in a visual dictionary. The algorithm determines an optimally compact dictionary by merging pairs of visual words obtained from manually segmented images. LOCUS [6] learns object classes using a generative probabilistic model that includes shape and appearance information. Variations due to deformation, translation, and scaling are also accounted for in the model. Gaussian mixture models are popular methods in image clustering [7]. However, the results of these methods for web image clustering are often limited due to the large variations in appearance, position and scale of the objects of interest in web images.

The clustering algorithm described in this paper extends the Gaussian mixture model and increases its flexibility in modelling the web images. First, each of the image clusters is described in turn by a Gaussian mixture model that captures better the variations in appearance of the web images. Secondly, the feature vectors used in clustering are obtained not only from the entire image but also from a set of image regions that determine a complete image segmentation. Often image segmentation depends on a set of parameters that are set beforehand. In our approach the values of the segmentation parameters are determined to maximize the likelihood of the images within each cluster. In turn the improved image segmentation regions determine a better set of features for image clustering leading to an iterative algorithm described in Section 4. The parameters of the image segmentation and the visual feature extraction techniques are described in Sections 2 and 3 respectively. The results of the image clustering algorithm described in this paper are presented in Section 5.

## 2. IMAGE SEGMENTATION

In this paper the image regions are obtained using JSEG segmentation algorithm [8]. JSEG starts by determining a set of "region seeds" in a color quantized space. The region seeds are determined by computing a homogeneity score "J" obtained from image blocks at different scales. Next, the seeds are grown into larger regions and are merged based on color histogram similarity measure. The segmented regions can be obtained by varying three parameters. The influence of these parameters on the segmentation results is discussed below.

1. The quantization parameter  $quant \in [0, 600]$  determines the minimum distance between two quantized colors in the image. The larger is the value of the  $quant$  parameter, the coarser is the quantization.
2. The scaling parameter  $scale$  controls the size of the image windows used in determining the region seeds. The smaller the value of  $scale$  parameter, the coarser the obtained segmentation.
3. The region merging parameter  $merge \in [0, 1]$  determines the merging threshold between two adjacent regions based on the distance between their color histograms. The larger  $m$  is, the more regions are merged together.

In order to choose a set of values for the segmentation parameters, we randomly picked 36 images out of our database, one for each category in our database and we conducted the following experiment. We applied the JSEG algorithm to all the test images using

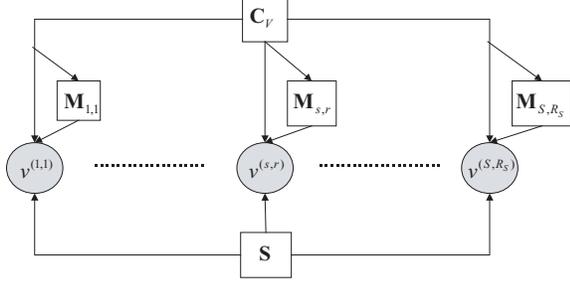


Fig. 1. The image clustering model

a large set of combinations of the segmentation parameters. Based on our observations, we decided to set  $scale = 1$  in order to avoid over segmenting the image. We set  $quant \in \{20, 255, 600\}$ , and  $merge \in \{0.4, 0.6, 0.8\}$  that determine 10 possible segmentations (including the overall image) for each image. Note that the parameter values are spread over the entire range of  $quant$  and  $merge$  parameters. This determines a large variety of segmentations of each image to be used in the image clustering algorithm.

The region based segmentation algorithm described above is compared with a low complexity block segmentation method. In block segmentation the image is partitioned into  $N_x$  and  $N_y$  horizontal and vertical non-overlapping rectangular blocks respectively. In this paper we consider 10 segmentation configurations of the image into  $(N_x, N_y) = (1,1)$  corresponding to the whole image, (1,2), (1,3), (2,1), (2,2) (2,3), (3,1), (3,2), (3,3) and (4,4) uniform non-overlapping blocks.

### 3. THE VISUAL FEATURE EXTRACTION

The visual features used in this paper are extracted from each image block or image region obtained by the previous segmentation algorithms. At first the content of each image region is captured using histograms of salient SIFT keypoints [9], a technique similar to that presented by Csurka *et. al.* in [10]. In our approach the dimensionality of the SIFT points is first reduced using Principal Component Analysis (PCA) to dimension 40. Next, a “soft histogram” with 100 bins is computed for all the vectors in each of the image regions obtained as described in the previous section. In our approach the value of a histogram bin is calculated as:

$$P(b_i) = \sum_j P(b_i|v_j)P(v_j)$$

where  $v_j$  are the set of visual feature vectors uniformly distributed and  $P(v_j|b_i)$  is determined by a Gaussian density functions with parameters learned from all visual vectors through the EM algorithm. The SIFT based features described above are compared in Section 5 to several other image features extraction techniques including color moments [11], edge directions [12], wavelet transform coefficients [13] and the HSV correlograms [14].

### 4. IMAGE CLUSTERING

The clustering method described in this paper is illustrated by the Bayesian network in Figure 1.  $C_V$  is the hidden cluster node that takes the discrete values  $c$ ,  $S$  is a hidden node with discrete values  $s$  associated with each segmentation configuration and  $M_{s,r}$  is a

hidden node with discrete values  $m$  representing the mixture component of each cluster corresponding to region  $r$  in segmentation configuration  $s$ . The continuous observations nodes are denoted as  $\mathbf{v} = [v^{(s)}]$ , where  $v^{(s)} = [v^{(s,r)}]$  is a sequence of observation vectors corresponding to the regions in the  $s$ th segmentation configuration,  $r = 1, \dots, R_s$ , is the index over all regions in an image and  $R_s$  is the number of regions extracted from the image under the  $s^{th}$  segmentation. The observation likelihood is given by

$$P(\mathbf{v}|c) = \sum_s P(\mathbf{v}|s, c)P(s)$$

where  $P(\mathbf{v}|s, c) = \prod_r P(v^{(s,r)}|s, c)$  and  $P(v^{(s,r)}|s, c)$  is given by a Gaussian mixture

$$P(v^{(s,r)}|s, c) = \sum_m P(m|c)\mathcal{N}(v^{(s,r)}, \mu_{c,m}, \sigma_{c,m})$$

with mean  $\mu_{c,m}$  and variance  $\sigma_{c,m}$ . The parameters of the model are learned using the EM algorithm described by the following steps:

1. initialize the parameters of the Gaussian mixtures for each of the image clusters. The initial assignments of images to clusters is determined at random or using the relevant text associated with each web image.
2. compute the likelihood of all observations and the a posteriori probability  $\gamma_{c,m}^{(n,s,r)} = P(c, m, s|v_n^{(s,r)})$ .

$$\gamma_{c,m}^{(n,s,r)} = \frac{P(v_n^{(s,r)}|m, c, s)P(m|c)}{\sum_{m', c', s'} P(v_n^{(s,r)}|m', c', s')P(m'|c')}$$

where  $n = 1, \dots, N$  is the index over all  $N$  images in the database. The computational complexity is decreased by using “hard assignment” and computing  $\gamma_{c,m}^{(n,s,r)}$  as follows:

$$\gamma_{c,m}^{(n,s,r)} = \begin{cases} 1, & \text{if } \{c, m, s\} = \arg \max_{c', m', s'} \{P(m'|c')\} \\ P(v_n^{(s,r)}|m', c', s') & \\ 0, & \text{otherwise.} \end{cases}$$

3. update the parameters of the Gaussian mixture for each cluster from the image blocks and the partition assigned to that cluster.

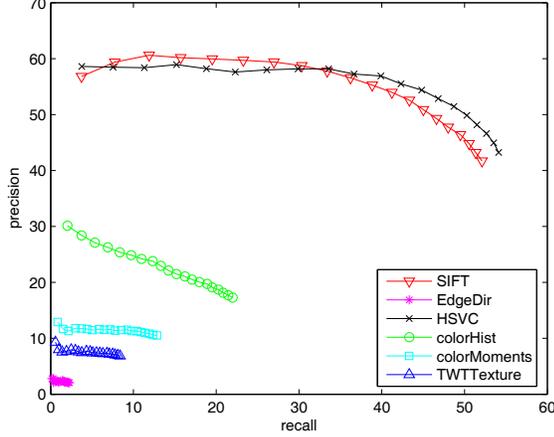
$$P(m|c) = \frac{\sum_{n,s,r} \gamma_{m,c}^{(n,s,r)}}{\sum_{n,s,r} \sum_{m'} \gamma_{c,m'}^{(n,s,r)}}$$

$$\mu_{c,m}(i) = \frac{\sum_{n,s,r} \gamma_{c,m}^{(n,s,r)} v_n^{(s,r)}(i)}{\sum_{r,n,s} \gamma_{c,m}^{(n,s,r)}}$$

$$\sigma_{c,m}^2(i) = \frac{\sum_{n,s,r} \gamma_{c,m}^{(n,s,r)} (v_n^{(s,r)}(i) - \mu_{c,m}(i))^2}{\sum_{n,s,r} \gamma_{c,m}^{(n,s,r)}}$$

where  $i$  is the index over the dimensions of the feature vector. Note that for simplicity the covariance matrix is diagonal.

4. compute the observation likelihood  $\prod_n P(\mathbf{v}^{(n)}|c)$  for all images given the best image partition and best cluster assignment.
5. if the absolute difference between the observation likelihood at consecutive iterations falls below a threshold stop, otherwise go to Step 2.



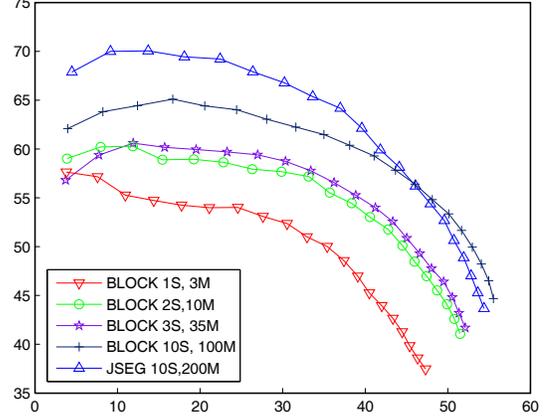
**Fig. 2.** Precision and recall for image clustering using several visual features types.

## 5. EXPERIMENTAL RESULTS

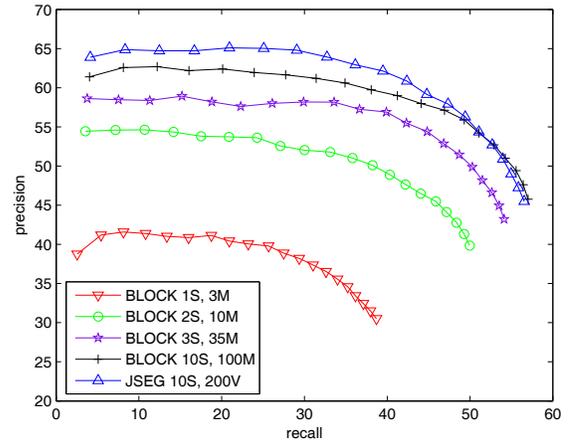
The experimental results of our approach were tested on a database consisting of over 6000 web images corresponding to 36 queries retrieved by the Google image search engine. The queries correspond to nine categories such as sports, flowers, animals, fruits, cities, companies, celebrities, universities and natural disasters and are ‘soccer’, ‘cricket’, ‘tennis’, ‘basketball’, ‘roses’, ‘tulip’, ‘jasmine’, ‘daffodils’, ‘dog’, ‘horse’, ‘elephant’, ‘snake’, ‘grapes’, ‘apple’, ‘mango’, ‘orange’, ‘Beijing’, ‘Paris’, ‘Chicago’, ‘Delhi’, ‘Intel’, ‘IBM’, ‘Walmart’, ‘Citibank’, ‘George Bush’, ‘mother Teresa’, ‘Michael Jordan’, ‘Tom Hanks’, ‘Stanford’, ‘Georgia Tech’, ‘Princeton’, ‘Harvard’, ‘volcano’, ‘earthquake’, ‘hurricane’ and ‘fire’. For each of the above queries we have generated ten extended queries. By example for query ‘roses’ the extended queries are ‘red roses’, ‘yellow roses’, ‘roses bouquet’, ‘roses vase’, ‘roses painting’, ‘roses white’, ‘pink roses’, ‘roses wreath’, ‘blue roses’ and ‘garden roses’. For each extended query we gathered the top 20 images and web pages returned by Google web image search engine. The number of images for each query varies between 150 and 200.

A common performance measure used in clustering is the inter-class and intra-class distances. This criteria measures the quality of clusters in the absence of ground truth data. An alternative criteria, common in data retrieval, and also used in this paper is the computation of the average precision and recall. In our experiments the data obtained for each of the  $N_q = 36$  queries was automatically clustered into  $K = 10$  clusters. Next, in order to match the labels of the ground truth data (determined manually) and the labels computed by our clustering method we used a maximum weight matching algorithm on a weighted bipartite graph  $G(V = O \cup C, E, W)$  where,  $O$  and  $C$  contain vertices for each of the original clusters and computed clusters respectively.  $E$  is the complete set of possible edges and  $W$  is the set of weights where each edge has a weight equal to the number of common elements in the clusters it connects. The quality of the clustering system is determined by the average precision  $P$  and recall  $R$  over all queries computed as

$$P = \frac{1}{N_q} \sum_{q=1}^{N_q} \frac{1}{K} \sum_{k=1}^K \frac{N_c(k, q)}{N_r(k, q)} \quad (1)$$



**Fig. 3.** Precision and recall for image clustering using SIFT features and four types of block segmentation configurations and 10 JSEG-based segmentation configurations.

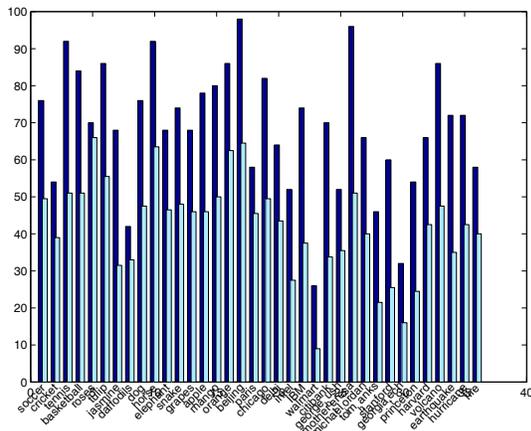


**Fig. 4.** Precision and recall for image clustering using HSVC features and four types of block segmentation configurations and 10 JSEG-based segmentation configurations.

$$R = \frac{1}{N_q} \sum_{q=1}^{N_q} \frac{1}{K} \sum_{k=1}^K \frac{N_c(k, q)}{N_l(k, q)} \quad (2)$$

where  $N_c(k, q)$  is the number of correct matched images for query  $q$  and cluster  $k$ ,  $N_l(k, q)$  is the number of images assigned to cluster  $k$  in query  $q$  in the ground truth data, and  $N_r(k, q)$  is the total number of documents assigned to cluster  $k$  in query  $q$  by our clustering algorithm.

We tested the image clustering performance for the visual features described in Section 3. It can be seen that among all the features tested the SIFT histograms and HSV correlograms perform best. At lower recall values the SIFT histograms achieve higher precision than HSV correlograms features while at higher recall values the roles are changed. In these experiments we used features extracted from the full image as well as features extracted from three block based image segmentations into  $(N_x, N_y) = (1,1), (2,2), (4,4)$  non-overlapping rectangular blocks. In all of the above experiments we used 35 mixtures per cluster.



**Fig. 5.** Precision at top 5 (left column) and top 20 (right column) recall for each of the 36 queries.

Figures 3 and 4 compare the image clustering results for different image partitions using SIFT histograms and HSVC features. The graph compares the performance of the image clustering described in Section 4 using block and data driven segmentations. In block segmentations we set four values for the  $S$  and  $M$  nodes of the clustering model (Figure 1) as follows:  $S = \{1, 2, 3, 10\}$  and  $M = \{3, 7, 35, 100\}$  respectively. The precision-recall curves for each of these parameters are denoted as BLOCK 1S 3M, BLOCK 2S 10M, BLOCK 3S 35M and BLOCK 10S 100M respectively. For  $S = 1$  the entire image is considered, for  $S = 2$  the image is segmented in  $(N_x, N_y) = (1,1), (2,2)$  blocks respectively, for  $S = 3$  the image is segmented in  $(N_x, N_y) = (1,1), (2,2), (4,4)$  blocks respectively and for  $S = 10$  the image was segmented in  $(N_x, N_y) = (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)$  and  $(4,4)$  blocks respectively. For data driven segmentation we used ten segmentation configurations  $S = 10$  obtained using the parameter sets described in Section 2. In this experiment we used 200 mixtures per cluster ( $M = 200$ ). This experiment is denoted as JSEG 10S 200M. Note that when features are extracted from the entire image the algorithm becomes a Gaussian mixture fitting and the performance increases with the number of block segmentations used. However, partitioning the image into rectangular blocks does not always capture the actual semantic regions in the image. As seen in Figures 3 and 4, using JSEG segmentation with 10 segmentation configurations per image leads to a significant increase in performance over block based segmentation with the same number of configurations. Figure 5 illustrates the precision and recall at top five and top 20 for each of the 36 queries for the JSEG 10S 200M experiment.

## 6. CONCLUSIONS

This paper describes a web image clustering algorithm that relies on a unified Bayesian framework that iteratively selects the best segmentation and image clusters from existing data. In our approach, the visual features were obtained from either rectangular image blocks or from image regions obtained using the JSEG segmentation algorithm. The accuracy of image clustering increased with the number of block segmentation configurations and reached the best overall performance for 10 JSEG-based image segmentations. In addition, our experimental results determined that visual features extracted

HSV correlograms and from the histogram of SIFT points within each image region provide the most reliable set of features for image clustering among those presented in this paper. Future work will be directed towards enhancing the image feature extraction and including more complex region based feature extraction techniques to increase the accuracy of the image clustering method.

## 7. REFERENCES

- [1] “Webseek: A content-based image and video search and catalog tool for the web,” <http://persia.ee.columbia.edu:8008/>.
- [2] D. Cai, X. Hi, Z. Li, W. Ma, and J. Wen, “Hierarchical clustering of www image search results using visual, textual and link information,” in *Proceedings of the 12th annual ACM international conference on Multimedia*, 2004, pp. 952–959.
- [3] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, “Learning object categories from google’s image search,” in *Proceedings of the International Conference on Computer Vision*, 2005.
- [4] T Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, 1999.
- [5] J. Winn, A. Criminisi, and T. Minka, “Object categorization by learned universal visual dictionary,” in *Proceedings of the International Conference on Computer Vision*, 2005.
- [6] J. Winn and N. Jovic, “Locus: Learning object classes with unsupervised segmentation,” in *Proceedings of the International Conference on Computer Vision*, 2005.
- [7] J. Goldberg, S. Gordon, and H. Greenspan, “Unsupervised image-set clustering using an information theoretic framework,” *IEEE Transactions on Image Processing*, pp. 449–458, 2006.
- [8] Y. Deng and B. Manjunath, “Unsupervised segmentation of color texture regions in images and video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 800–810, 2001.
- [9] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 2, no. 60, pp. 91–110, 2004.
- [10] G. Csurka and J. Willamowski C. Bray C. R. Dance, L. Fan, “Visual categorization with bags of keypoints,” *Proc. of the 8th European Conference on Computer Vision, Prague, May 2004*.
- [11] Markus A. Stricker and Markus Orengo, “Similarity of color images,” in *Storage and Retrieval for Image and Video Databases (SPIE)*, 1995, pp. 381–392.
- [12] H. Tamura, S. Mori, and T. Yamawaki, “Texture features corresponding to visual perception,” *IEEE Transactions on Systems, Man and Cybernetics*, pp. 460–473, 1978.
- [13] T. Chang and C. Kuo, “Texture analysis and classification with tree-structured wavelet transform,” *IEEE Transactions on Image Processing*, vol. 2, pp. 429–441, 1993.
- [14] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih, “Image indexing using color correlograms,” 1997.