Oriental Journal of Computer Science & Technology Vol. **1**(2), 155-160 (2008)

Content-based image retrival in the World Wide Web

P.N.R.L. CHANDRA SEKHAR[1], D. RAJYA LAKSHMI[2], J.A. CHANDULAL[3] and G.V.S. RAJKUMAR[4]

[1]Department of CSE, [2]Department of IT, [3]Department of CSE, Department of IT, GITAM University, Visakhapatnam (India)

(Received: November 1, 2008; Accepted: December 25, 2008)

ABSTRACT

In general the image search engines on the World Wide Web rely purely on the keywords around the images and the filenames, which produces a lot of irrelevant images in their search results. Alternative to this there are other methods based on content based image retrieval which requires user interaction to submit a query image, and gets images that are similar in content. In this paper we presented a novel approach that combines the two above methods for retrieval of relevant images. In this method we first retrieves the results of a keyword query from an existing image search engine, then clusters the results based on extracted image features such as color and texture and returns the cluster that is inferred to be the most relevant to the search query.

Keywords: keyword search, clustering, image segmentation, color and textures.

INTRODUCTION

The World Wide Web contains a great quantity of image and other visual information [4] such as videos, movies, and comic strips, that may belong to structured collections (e.g., museum collections) or be independent (e.g., images found in Web pages in the form of individuals photographs, logos, and so on). Considering this enormous amount of visual information unorganized, without efficient retrieval tools it would be appreciably reduce its utilization by users and prevent them benefiting from it. Hence, it is vital to have tools for image retrieval from the Web and index the visual content of the Web into a searchable catalog that helps users to navigate across various subjects. This could help them quickly find the images they seek, which can then be used for many purposes.

The development of image retrieval engines for the Web may be useful for many professional applications, such as crime prevention and intellectual property protection as well as for other purposes such as home entertainment, education, and training.

Searching for effective methods to retrieve information such as image retrieval from the Web has been in the center of many research efforts during the last few years and also the relevant technologies evolved rapidly.

By using a standard web-based keyword search engine, one is likely to find some of the irrelevant images even in the first couple of results. Whereas content-based image retrieval (CBIR)[5] information systems use information extracted from the content of images for retrieval, and help the user retrieve images relevant to the contents of the query. Nevertheless, most CBIR systems require a user to provide one or more query images[8]. In[1] various clustering algorithms are used to display results of a CBIR system in visually similar groups. This approach makes no attempt in weeding out groups

of images that are irrelevant to the search query and no re-ranking the search results.

In[2] Fergus *et al.* proposed a method that uses the top results from a web-based image search engine to train a classifier, and then filter the search results. To overcome the problem of irrelevant images from the image search, they use an *ad hoc* approach of getting the top 5 results for the same query in different languages (making the assumption that the top 5 results will most likely be correct), and developing a robust classifier that would overcome possible noise and variability of the training set. Furthermore, their feature selection is based on interest point detection and ignores color and texture, which represent powerful cues for object recognition.

In this paper we present an image retrieval system in which a user need only provide a keyword query, as is the case in standard web-based image search engines. The system, however, infers a representation of the object of interest, and re-ranks the images according to relevance based on their content.

The paper is organized as follows. Chapter 2 describes about the structure and overview of the proposed method, chapter 3 describes about implementation, results and conclusions discussed in chapter 4 and chapter 5.

System Overview

In our system first the user has to submit query for an image as is the case in standard web-based image search engine and it extracts the

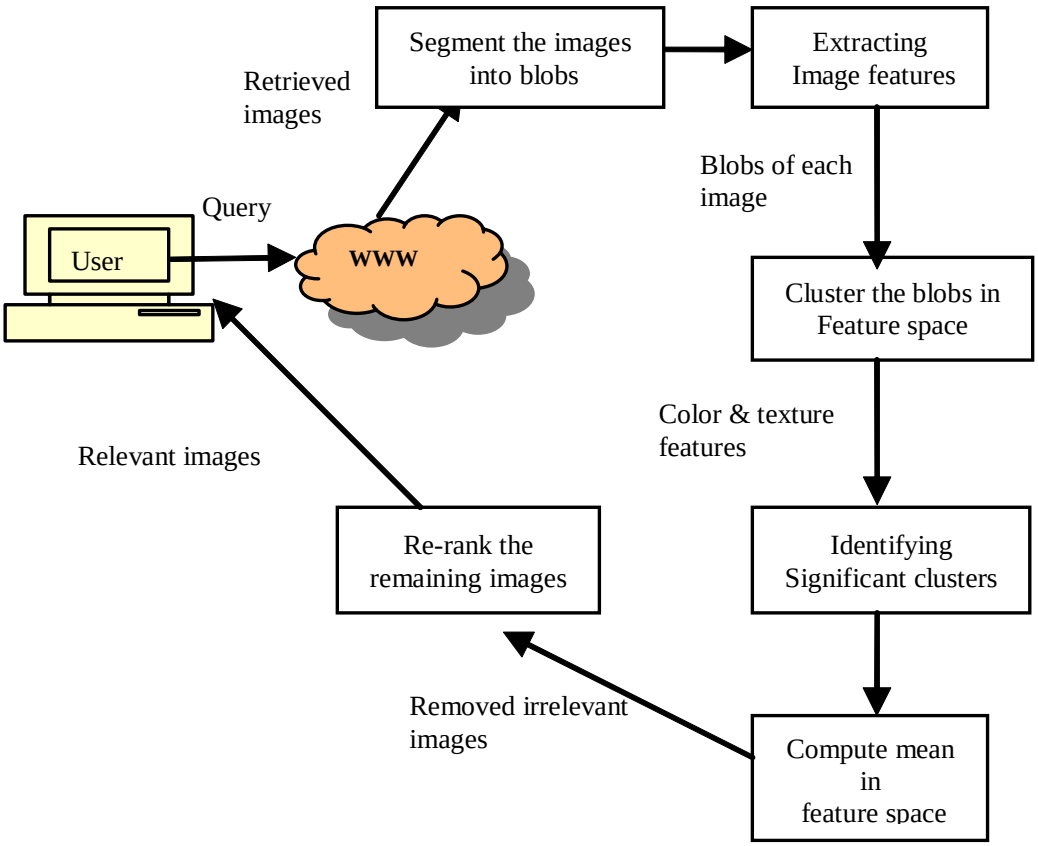

Fig 1: Structure of proposed system

relevant features of each image such as color and texture. As shown in Fig 1 the system has several integral components. First, each image is segmented into similar regions or "blobs." Next, a set of features in terms of color and texture are extracted from each of these blobs. The set of feature vectors retrieved from the top image search results is then clustered using the mean shift clustering algorithm. We posit the cluster that corresponds to the largest number of parent images to be the object of interest, and refer to this as the "significant" cluster. Lastly, a large set of images from the image search is re-ranked based on similarity to this "significant" cluster.

Image Segmentation

Normally every image contains multiple objects, or at least one object and a background. By that, extracting features globally is not appropriate. For this reason we start by segment each image into different regions of similarity, using an image segmentation algorithm, with the intuition that each of these regions is a separate object in the image.

Image segmentation is a well studied problem in Computer Vision, and there are many existing algorithms for this task. We chose to use an algorithm by Felzenszwalb and Huttenlocher, as described in[3], because of its ease of use and speed. This segmentation algorithm partitions an image into similar regions using a graph based approach. Each pixel is a node in the graph with undirected edges connecting its adjacent pixels in the image. Each edge has a weight encoding the similarity of the two connected pixels. The partitioning is done such that two segments are merged only if the dissimilarity between the segments is not as great as the dissimilarity inside either of the segments.

Extracting image features - color and texture

In order to obtain a measure of how similar image blobs are to one another, good features are needed to represent the blobs. So, while segmenting an image into blobs, we extract six features from each pixel. In which three features for color and the other three for texture. We chose color histograms in HSV color space as our color features due to its ability for easy transformation from RGB to HSV and vice versa. These features are denoted as (F1, F2, and F3). To obtain the other three texture features, we apply the Haar wavelet transform to the image. The Haar wavelet is discontinuous and resembles a step function. It represents the energy in high frequency bands of the Haar wavelet transform[9]. After a one-level wavelet transform, a 4 by 4 block is decomposed into four frequency bands, each band containing a 2 by 2 matrix of coefficients. Then, the feature of the block in the HL band is computed. The other two features are computed similarly in the LH and HH bands. These three features of the block are denoted as (F4, F5, and F6). Combining both features we form a feature vector FV (F1... F6).

Mean shift clustering in feature space

From the extracted features we need to cluster the blobs, with the hope that the object of interest will form the largest cluster. Since some of the blobs may represent irrelevant, it is difficult to find the number of clusters that are present. Hence, the standard *k*-means clustering approach is not appropriate. Alternative to this we used the mean shift clustering algorithm. This algorithm is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters. As described in[7] the algorithm begins by placing a window (usually a hyper sphere) around each point in the feature space. On each iteration, each window moves in the direction of the mean shift vector which is computed as follows:

$$y_{t+1} = \frac{1}{|\theta_\lambda|} \sum_{x \in \theta_\lambda} (y_t - x) \quad \ldots(1)$$

In the above eq. (1) y_t is the window center at iteration *t*, and θ_λ is the set of points in the hyper sphere window of radius ë. It is also possible to use a kernel function to weight points according to how far they are from the window center. The windows eventually converge on the points of local density maxima, and these points are returned as the cluster centroids. Thus, the mean shift clustering algorithm eliminates the parameter *k* (number of clusters) at the price of introducing another parameter λ. This parameter, however, is easier to tune to all possible inputs than *k*.

Re-ranking the Images

When the "significant" clusters are obtained

(a) Before - Ranking (Yahoo Image Search results)

(b) After- Ranking (Proposed system results)

Fig. 2: The top results from Yahoo Image Search (a) and our system (b).

in the feature space, next the mean is to be computed. The rest of the images are then resorted based on the distance of their blobs to this mean. Since each image could potentially contain more than one blob, the closest blob in each image is used. Chi-squared distance comparisons are used in the re-sorting because it is known that for histograms, a chi-squared distance measure yields better results than *L*2 distance [6].

As shown in eq. (2) the distance $D_{x^2}(I,J)$ between two histograms *I* and *J* is computed as follows:

$$D_{x^2}(I,J) = \frac{1}{2}\sum_{k=1}^{K}\frac{\left(I_k - J_k\right)^2}{I_k + J_k} \quad ...(2)$$

The result is a re-ranking of the images from the original search engine.

Implementation

Firstly, we used keyword search by the given query, and downloaded the first 200 images from the existing search engine such as Yahoo's search engine using the Yahoo Image API. Since Yahoo Image Search engine provides a free image search API we selected it. Felzenszwalb and Huttenlocher's segmentation algorithm was then run on each image

For the segmentation, we used the default parameters of $k = 200$, $A_{min} = 10$, and $\sigma = .5$, where *k* is how dissimilar the blobs can be, A_{min} is the minimum blob size, and ó is the parameter of the Gaussian which is used to smooth the image before segmentation. After this preprocessing completed, the clustering was run on the first 10 images with the mean shift window size set to 50.

Blobs that were too small (took up less than 5% of the image area) were filtered out of the clustering. The rest of the 200 images were re-ranked as described above.

RESULTS

The above is the key word search Results for the query "Red Roses" (top) and "White horses" (bottom). The yellow circled ones are the results which we felt were completely irrelevant to the given query, although this is very subjective depending on the user. It is very clear that the top results from our proposed method are much more visually consistent than Yahoo's top results. Most CBIR systems use precision and recall to evaluate their performance by means of various queries. As this system is a web-based image search this kind of evaluation is not appropriate as it is subjective to human perception.

As we observed the system works fine for simple queries like what we showed in the results. If a query is vague, or has multiple semantic meanings, the method will favor for more prominent meaning. When ambiguity exists, the system concentrates only on single meanings. For instance, consider the "Red roses" example in Fig. 2. Although the results from the Yahoo Image Search engine demonstrate more variety, they also contain irrelevant images. The results from our system exploit the visual consistencies of red roses, and return only the images with roses as we considered texture features while clustering the blobs.

CONCLUSIONS

In this paper we presented content based image retrieval system that uses the results of a standard web-based image search engine, and re-ranks these results in order of relevance. The only requirement for the system is the user has to give a keyword as query, and infers a representation of this query in feature space, which is used to re-rank the results. In the implementation, the image features consisting of smoothed HSV histogram and texture descriptors. Finally, for simple queries we showed the results and compared with the existing search engine results.

REFERENCES

1. Y. Chen, J. Wang, and R. Krovetz. CLUE: cluster-based retrieval of images by unsupervised learning. *IP*, **14**(8): 1187-1201 (2005).
2. R. Fergus, P. Perona, and A. Zisserman. A visual category filter for Google images. In *ECCV*, pages **1**: 242-256 (2004).
3. P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, **59**(2): 167-181 (2004).
4. M. L. Kherfi and D. Ziou and A. BernardI, Image Retrieval From the World Wide Web: Issues, Techniques, and Systems, *ACM Computing Surveys,* 36(1) (2004).
5. E. Di Sciascio, G. Mingolla, and M. Mongiello**,** Content-Based Image Retrieval over the Web Using Query by Sketch and Relevance Feedback VISUAL'99, LNCS 1614, 123-130 (1999).
6. J. Puzicha, T. Hofmann, and J. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *CVPR*, pages 267-272 (1997).
7. Y. Cheng. Mean shift, mode seeking, and clustering. *PAMI*, **17**(8):790-799 (1995).
8. Special issue on Content Based Image Retrieval, **28**: 9 (1995).
9. Ingrid Daubechies, Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics, Philadelphia, PA (1992).