Improving Web-based Image Search via Content Based Clustering

Nadav Ben-Haim, Boris Babenko and Serge Belongie
Department of Computer Science and Engineering
University of California, San Diego
{nbenhaim,bbabenko,sjb}@ucsd.edu

Abstract

Current image search engines on the web rely purely on the keywords around the images and the filenames, which produces a lot of garbage in the search results. Alternatively, there exist methods for content based image retrieval that require a user to submit a query image, and return images that are similar in content. We propose a novel approach named ReSPEC (Re-ranking Sets of Pictures by Exploiting Consistency), that is a hybrid of the two methods. Our algorithm first retrieves the results of a keyword query from an existing image search engine, clusters the results based on extracted image features, and returns the cluster that is inferred to be the most relevant to the search query. Furthermore, it ranks the remaining results in order of relevance.

1. Introduction

Due to the fact that web-based image search engines are blind to the actual content of images, the result of querying for a specific object is often cluttered with irrelevant data. Alternatively, much research has been done on content based image retrieval (CBIR). Nevertheless, most CBIR systems require a user to provide one or more query images. In [3] the user provides an image, and selects the "blob" in that image that represents the object of interest. Although quite flexible, this system is burdensome on the user. Similarly, in [15] the system requires several images from a user, which it uses as a training set to build a classifier using AdaBoost. This, again, requires the user to to be in possession of example images. Other research has focused on the task of learning probability models linking text to image regions ([2], [1]). This task requires a large set of data labeled by a human. Also, it is not clear how these techniques would perform on noisy data sets, with mislabeling, as is the case with the Internet. Alternatively, in [14] and [4], various clustering algorithms are used to display results of a search engine or CBIR system in visually similar groups. These approaches, however, make no attempt in weeding out groups of images that are irrelevant to the search query, nor re-ranking the search results.

Recently, Fergus *et al*. have 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 in [8], or classify novel images into categories in [7]. To overcome the problem of garbage results 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 ReSPEC, a system in which a user need only provide a keyword query, as is the case with standard web-based image search engines. Our system, however, infers a representation of the object of interest, and re-ranks the images according to relevance based on their content.

Lastly, we discuss how our approach can be applied to the more difficult task of transferring text labels from web based image databases to novel photograph collections for purposes of object recognition.

2. Approach

When using a standard web-based image search engine, one is likely to find garbage output even in the first couple of pages of results. Nevertheless, for simple objects, it can be conjectured that most of the top results will contain the object of interest. Our system exploits this consistency, and attempts to find the object that appears in most of these images.

ReSPEC has several integral components (see Fig. 1). First, each image is segmented into similar regions or "blobs." Next, a set of features is 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 algorithm. We posit the cluster that corresponds to the

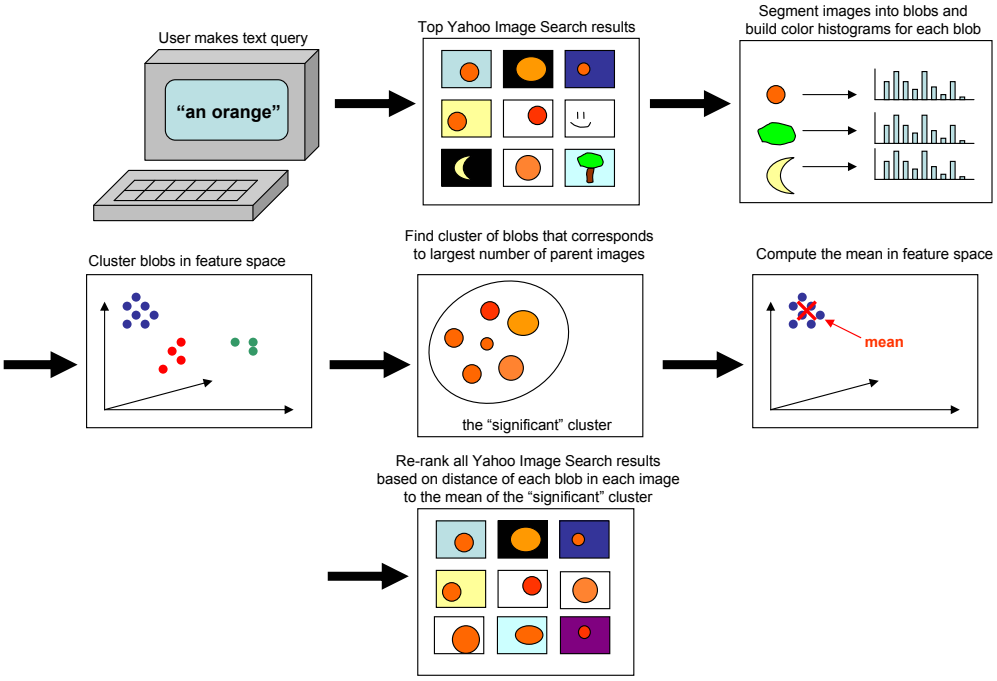

Figure 1. A flowchart depicting the ReSPEC system with the query "an orange."

largest number of parent images to be the object of interest, and we refer to this as the "significant" cluster (see Fig. 2). Lastly, a larger set of images from the image search is re-ranked based on similarity to this "significant" cluster.

2.1. Image Segmentation

Each image most likely contains multiple objects, or an object and a background. Therefore, extracting features globally is not appropriate. For this reason we start by splitting each image into 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 [6], 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.

2.2. Feature Selection

In order to obtain a measure of how similar image blobs are to one another, good features are needed to represent the blobs. We chose to use color histograms in HSV color space as our features. To form a feature vector for each blob, histograms are built for the H, S and V channels, with 15 bins each, and then concatenated together to form a 45 dimensional feature vector. Although histograms have clear advantage over taking the mean color of all the pixels in the blob, there is an inherent problem. For example, consider the following three histograms for hue: $X = (1,0,0,0,0,0,0), Y = (0,1,0,0,0,0,0), Z = (0,0,0,1,0,0,0)$. It is clear that X and Y are more similar in hue. Nevertheless, the distance between X and Y, and X and Z are equal.

To overcome this problem, Hafner *et al.* introduced a method in [9], to measure distance between two histograms I and J with inter-bin similarities encoded with a certain A matrix as follows:

$$D^2_{hist}(I,J) = (I-J)^\top A(I-J) \qquad (1)$$

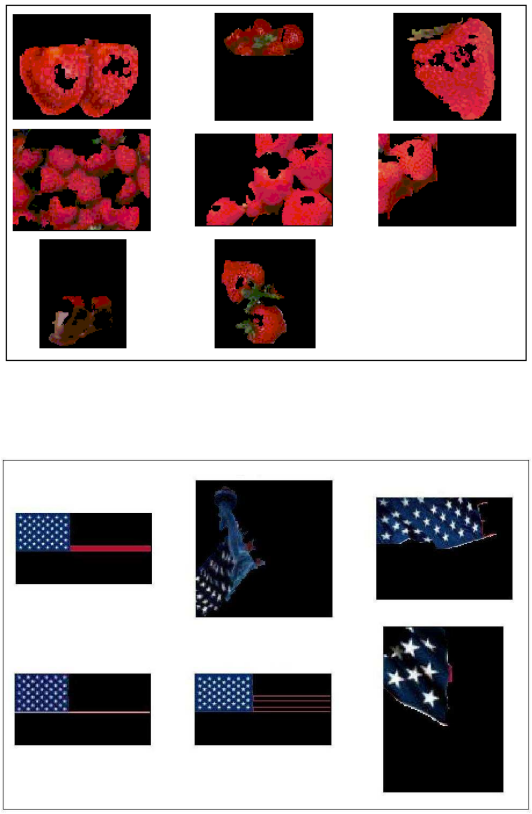

Figure 2. Examples of "significant" clusters for two queries: "strawberries" (top), and "us flag" (bottom).

We take a similar approach, where we choose a matrix A such that $A = B^\top B$ and B is a matrix where each row is a $1-$dimensional Gaussian, shifted by 1 from row to row. We achieve the same effect by convolving each channel with a Gaussian filter ($\sigma = 0.7$). The circularity effect of the Hue channel is taken into account.

2.3. Mean Shift Clustering in Feature Space

The next step in the system is to cluster the blobs, according to their extracted features with the hope that the object of interest will form the largest cluster. Since some of the blobs will represent garbage, it is difficult to predict the number of clusters that are present. Hence, a standard clustering approach such as *k*-means is not appropriate.

The mean shift clustering algorithm, which is an iterative gradient ascent method for finding local density maxima, was used instead. As described in [5][1], the algorithm begins by placing a window (usually a hypersphere) 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:

$$y_{t+1} = \frac{1}{|\Theta_\lambda|} \sum_{x \in \Theta_\lambda} (y_t - x) \tag{2}$$

where y_t is the window center at iteration t, and Θ_λ is the set of points in the hypersphere window of radius λ. 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.

2.4. Re-ranking the Images

After obtaining the "significant" cluster in feature space, the mean is computed. The rest of the images are then re-sorted 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 [12]. The distance $D_{\chi^2}(I, J)$ between two histograms I and J is computed as follows:

$$D_{\chi^2}(I, J) = \frac{1}{2} \sum_{k=1}^{K} \frac{(I_k - J_k)^2}{I_k + J_k} \tag{3}$$

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

3. Implementation

For a given search query, we downloaded the first 500 images from Yahoo's search engine using the Yahoo Image API[2]. We chose to work with the Yahoo Image Search engine because, to our knowledge, this is the only engine that provides a free image search API. Felzenszwalb and Huttenlocher's segmentation algorithm was then run on each image. For the segmentation, we used the default parameters of k = 500, A_{min} = 20, and σ = .5, where k is how dissimilar the blobs can be, A_{min} is the minimum blob size, and σ 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 15 images with the mean shift window size set to 55. Blobs that were too small (took up less than 5% of the image area)

[1]We used Piotr Dollár's implementation of this algorithm, which is available at http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html

[2]The API is available at http://developer.yahoo.net/search/image/V1/imageSearch.html

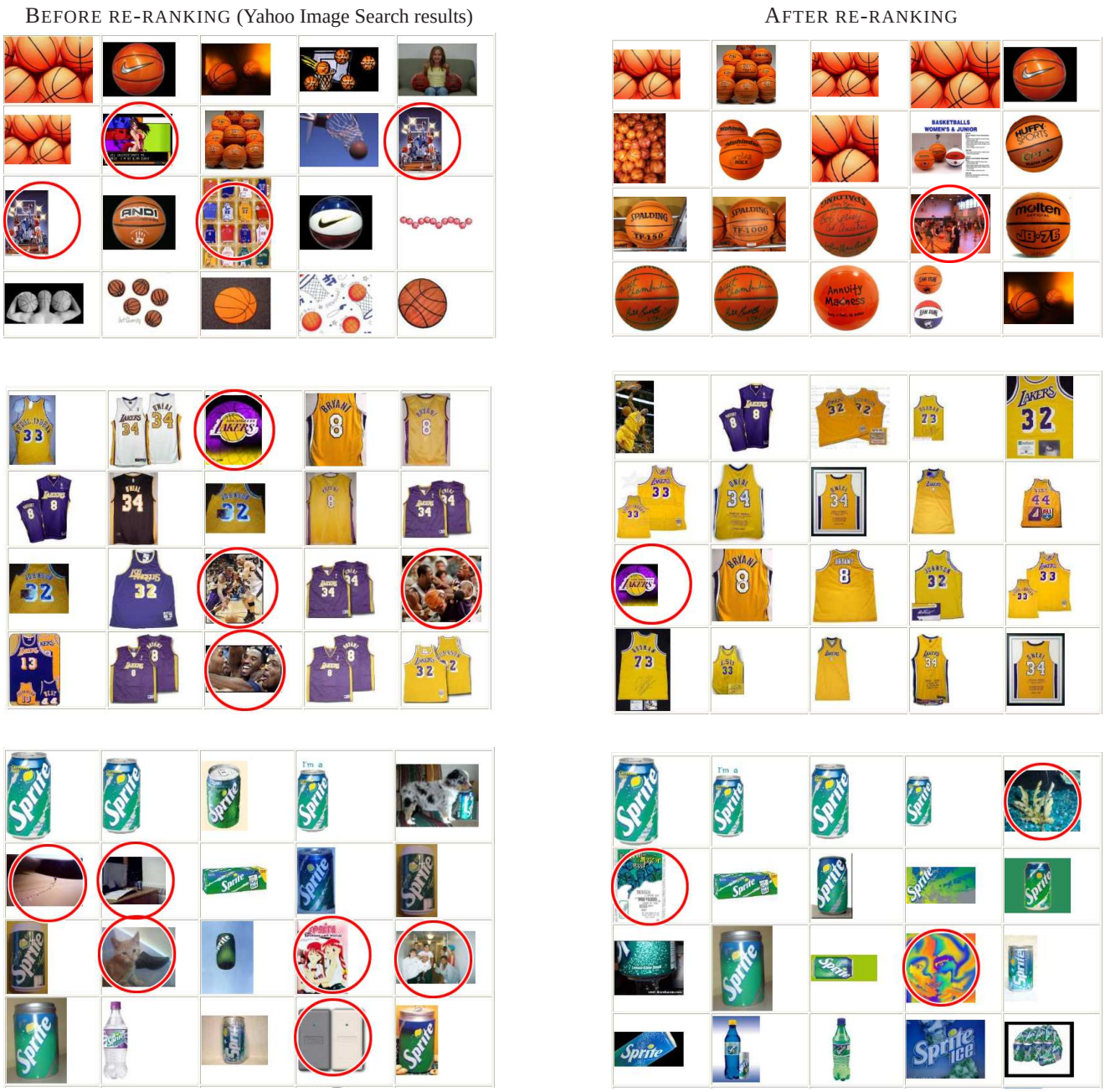

Figure 3. The top results from Yahoo Image Search (left) and ReSPEC (right). Results for the queries "basketballs"(top), "Lakers jersey"(middle), "sprite can"(bottom). We circled results which we felt were completely irrelevant to the query, although this is very subjective depending on the user. Nevertheless, it is clear that the top results from ReSPEC are much more visually consistent than Yahoo's top results.

were filtered out of the clustering. The rest of the 500 images were re-ranked as described above.

4. Results

Quantifying the results of an image retrieval system is not a trivial task. While most CBIR papers use a hand labeled corpus of images to compute precision and recall on a set of queries, this is not appropriate for the task of refining the results of a web-based image search engine. To competently perform evaluation of such a system, a study must be made using human subjects who would evaluate the qualitative performance. In this study we concentrated on the

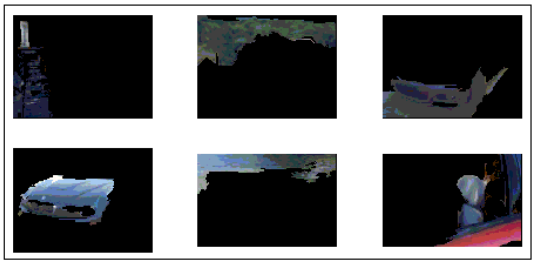

Figure 4. The "significant" cluster detected for the query "car". Since the object is highly variable in terms of color, our system fails on this query.

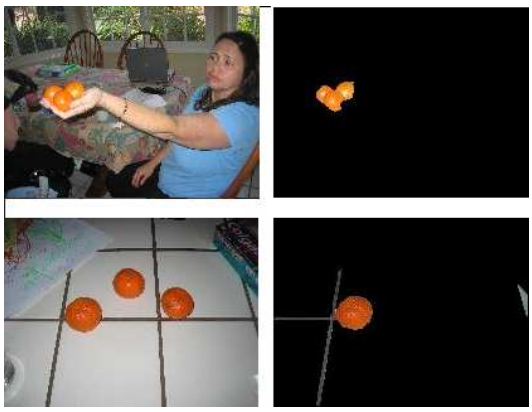

Figure 5. The visual representation for the query "oranges" was extracted from Yahoo Image Search, and then used to re-rank a collection of 10 unlabeled personal images. The two top results are shown. The left column shows the photos, while the right column displays the blobs that were closest to the "significant" cluster.

algorithmic aspects of discovering a visual representation of a given query, and we leave such evaluation for future work. Instead we report example results of our algorithm, and discuss them.

For queries that describe simple objects with a single, visually consistent semantic meaning, our method seems to work quite well. It is important to note that if a query is vague, or has multiple semantic meanings, our method will favor the more prominent meaning. The detection of multiple meanings per query seems plausible, but in this paper we chose to concentrate on single meanings, with the argument that a query could be made more specific if an ambiguity exists. For instance, consider the "Lakers jersey" example in Fig. 3. 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 Lakers jerseys, and return only the images with yellow regions. If a user is indeed interested in a different sort of Lakers jersey, for example, a purple jersey, he/she could narrow down the search query to "purple Lakers jersey."

Another important note about any system similar to what we have described in this paper, is that it must be efficient in order to be used in a real application. In our implementation, the segmentation and feature extraction could be done offline on an entire database (to further speed things up, only the thumbnail images are used). The rest of the processing takes a few seconds, which is fast enough for a real-time application.

5. Conclusions & Future Work

In this paper we presented ReSPEC, a system that uses the results of a standard web-based image search engine, and re-ranks these results in order of relevance. The novelty of our system is that it requires only a keyword query from the user, and infers a representation of this query in feature space, which is used to re-rank the results. In our implementation, the features consisted of a smoothed HSV histogram, but there are many other possible extensions (*e.g.* include textures and shape descriptors, use different colorspace, etc). For instance, in Fig. 4 we demonstrate an example where color information is not enough to capture the visual consistency for the query "car." In particular, it would be interesting to see how our approach would perform if feature selection were changed into interest point detection, or maximally stable extremal region (MSER)[11] detection in combination with SIFT region descriptors [10]. A similar approach was successfully used in [13] to search for visually similar regions in a video.

An exciting application of our approach is robust image annotation transfer from a noisily labeled set of images on the web to an unlabeled set of images in a personal collection. Most photograph collections gathered by people with digital cameras contain completely irrelevant metadata. For example, an image file might be called "DSC00010.JPG." Therefore, a standard keyword search for an object in such a collection would yield no results. In this paper, however, we have described a way to find a visual representation of a text query using the noisily labeled sets of images returned by image search engines. It seems plausible to use such a representation to then search through a collection of personal photographs on a user's machine and sort this collection by relevance to the query. The user would again only need to provide a text query, and in this sense such a system would be largely unsupervised. We attempted an experiment with a very small set of personal photographs, and saw some promising results (See Fig. 5 & Fig. 6). Due to the fact that our feature extraction is limited to color only,

however, we found this to not be discriminative enough to sort large collections of highly variable photos. The limited set of features works for the application that we described in this paper, because the distribution of image features in personal photo collections is in general different than the distribution of image features conditioned on a text query. Specifically, the probability that an object of interest will appear in the image search results is higher than the probability of finding that object in an unconstrained set of photos. Additional discriminative features would be necessary to attack the more challenging problem of generic object recognition in an unconstrained set of images. We leave such ideas to future work.

Figure 6. The visual representation for the query "grass" was extracted from Yahoo Image Search, and then used to re-rank a collection of 10 unlabeled personal images. These pictures are shown in their new order. Notice that the two pictures of grass are at the top.

Acknowledgements

We would like to thank the students of CSE 190A for helpful discussions. S.B. acknowledges support from NSF CAREER Grant #0448615 and the Alfred P. Sloan Research Fellowship.

References

[1] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, *3*, pages 1107–1135, 2003. 1

[2] G. Carneiro and N. Vasconcelos. Formulating semantic image annotation as a supervised learning problem. In *CVPR*, pages II: 163–168, 2005. 1

[3] C. Carson, M. Thomas, S. Belongie, J. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *VISUAL*, pages 509–516, 1999. 1

[4] Y. Chen, J. Wang, and R. Krovetz. CLUE: cluster-based retrieval of images by unsupervised learning. *IP*, 14(8):1187–1201, August 2005. 1

[5] Y. Cheng. Mean shift, mode seeking, and clustering. *PAMI*, 17(8):790–799, August 1995. 3

[6] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, September 2004. 2

[7] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google's image search. In *ICCV*, pages II: 1816–1823, 2005. 1

[8] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for Google images. In *ECCV*, pages Vol I: 242–256, 2004. 1

[9] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *PAMI*, 17(7):729–736, July 1995. 2

[10] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999. 5

[11] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, page 3D and Video, 2002. 5

[12] J. Puzicha, T. Hofmann, and J. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *CVPR*, pages 267–272, 1997. 3

[13] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, 2003. 5

[14] D. K. T. Deselaers and H. Ney. Clustering visually similar images to improve image search engines. *In Informatiktage der Gesellschaft für Informatik, Bad Schussenried, Germany*, 2003. 1

[15] K. Tieu and P. Viola. Boosting image retrieval. In *CVPR*, pages I: 228–235, 2000. 1