Defining Image Content with Multiple Regions-of-Interest

Baback Moghaddam[1], Henning Biermann[2] and Dimitris Margaritis[3]

[1] MERL - A Mitsubishi Electric Research Laboratory, Cambridge MA
[2] Courant Institute for Mathematical Sciences, New York University
[3] Department of Computer Science, Carnegie Mellon University
baback@merl.com, biermann@cs.nyu.edu, dmarg@cs.cmu.edu

Abstract

With the proliferation of multimedia, the web and digital imaging, there now exists a high demand for intelligent tools for image management, most importantly indexing, search and retrieval, commonly referred to as "query-by-image-content". Existing systems often make use of *global* attributes such as overall color distributions which ignore the actual composition of the image in terms of internal structures. In this paper we present an image retrieval system predicated on the principle that it is the *user* who is most qualified to specify the "content" in an image and not the computer. Therefore, the user is asked to provide salient "regions-of-interest" or ROIs and specify the importance of their spatial relationships in the query image. This technique leads to acceptable retrievals (equal if not better than global-based searches) and provides an intuitive interface that is more in tune with the user's notion of "content", thus providing a more powerful image retrieval tool.

Keywords. Content-Based Retrieval, Region-of-Interest, Histogram Matching, Spatial Constraints

1 Introduction

Most of the current content-based image retrieval systems rely on *global* image characteristics such as color and texture histograms (*e.g.,* Altavista's "Photofinder"). While these simple global descriptors are fast and often do succeed in partially capturing the essence of the user's query, they more often fail due to the lack of higher-level knowledge about what exactly was of interest to the user in the query image - *ie.,* the user-defined content. The goal of this research was to develop and test a new technique for image retrieval using *local* image representations in a bottom-up fashion. Our localized representations can be easily grouped into multiple user-specificed "regions-of-interest" and constrained to preserve their relative spatial configuration during retrieval. We posit that this leads to a more *user-centric* and thus a more powerful search engine.

The observation that spatial information is a critical component of image description and subsequent matching has not gone unnoticed by researchers in the field. Recently, the community has witnessed a gradual shift towards spatially-encoded image representations. These techniques range widely from fixed image partitioning in the "ImageRover" system of Sclaroff *et al.*[10], to highly local characterizations like the "color correlograms" of Huang *et al.*[5]. Somewhere in between these two extremes, one can find various techniques which deal with "regions" or "blobs". For example, the "configural templates" of Lipson *et al.*[7] specify a class of images (*e.g.,* snow-capped mountain scenes) by means of photometric and geometric constraints on pre-defined image regions. Other techniques use automatic blob segmentation and description, as in Howe's simple but effective "percentile blob" technique [4] or the more sophisticated "Blobworld" segmentation system of Carson *et al.*[1].

Our system differs from the above in one key aspect: there are no pre-segmented regions. Rather, the user defines "blobs" or ROIs directly on a query image (and implicitly their relative spatial configuration) in order to better communicate to the search engine the intended “content” (which could possibly represent only a subset or partial aspect of the query image selected). The disadvantage of this scheme, however, is that region-matching and subsequent database indexing must be done in an online fashion and moreover in “interactive-time” to be tolerated by the user. Aggressive search pruning and database re-organization does, however, alleviate this problem to some extent. The advantage, on the other hand, is that the user is not limited to working with the available set of pre-defined blobs, as in “Blobworld” [1].

2 Representation and Similarity

Image retrieval in general is based on two key components: a set of image features (like color or texture features) and a similarity metric (used to compare such features). To date most systems use global color histograms to represent the color composition of an image, thus ignoring the spatial layout of color in the query image. Likewise, a single global vector (or histogram) of texture measures (usually computed as the output of a set of linear filters at multiple scales) is used to represent textural attributes (such as granularity, periodicity, directionality, *etc.*) The similarity metric used to compute the degree-of-match between two images is often a Euclidean norm on the difference between two such global color/texture representations.

While global feature-based similarity matching has certain desirable properties (*eg.,* invariance to rotation) it fails to capture the spatial layout and structure of the image. Moreover, what the user typically thinks of as the "content" is seldom captured by the whole image or its global properties. Therefore, it is better to let the user identify the regions in the image which he/she is interested in (the "content"), with the possibility of specifying the spatial layout as a search constraint. This demands a *local* representation at the finest resolution possible, which can be easily grouped into larger regions and perhaps even integrated into to a single global description.

2.1 Local Feature Distributions

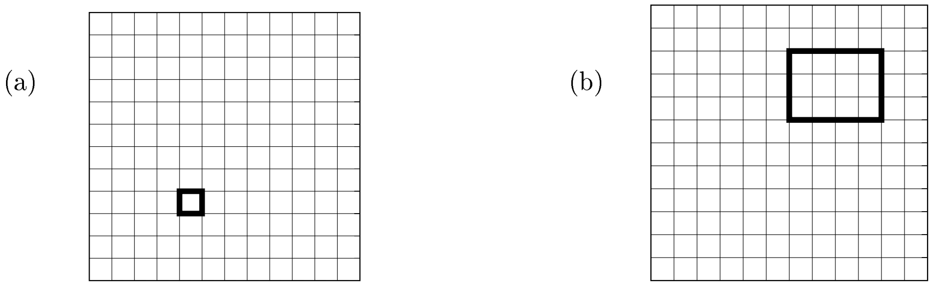
Our system divides the image into an array of 16-by-16 pixel blocks wherein each pixel yields a LUV color coordinate and three texture measurements; edge strength: $\log(G_x^2 + G_y^2)$, Laplacian: $G_{xx} + G_{yy}$ and edge orientation: $\arg(G_x, G_y)$, where G_x and G_{xx} are the 1st and 2nd derivatives of a Gaussian filter with specified scale σ. In our experiments, two separate scale parameters were used: $\sigma = 1$ and $\sigma = 2$, yielding two sets of (“independent” or at least uncorrelated) texture measurements.[1]

Estimates of the *joint distribution* of the features for color and texture were obtained non-parametrically by means of a joint 3D histogram in LUV color space (implemented with 5-by-5-by-5 bins) and a joint 3D histogram of edge magnitude, Laplacian and orientation (implemented with 4-by-7-by-4 bins), computed at two octave scales. The edge strength was quantized (classified) into only 4 values: {no edge, weak edge, average edge, strong edge}. Similarily, the edge orientation was classified into 4 values corresponding to {horizontal, vertical, diagonal left, diagonal right}. Note that in both histograms, the total number of bins is about 120 and given the 256 pixels in a 16-by-16 block, we average out to roughly 2 observations per bin. To aid the estimation process, we also used Bayesian *m-estimates* [3] in counting hits, using database-derived prior distributions in order to balance the tradeoff between prior belief and the observed data.

2.2 Histogram Similarity Measures

We implemented and tested 3 different histogram similarity measures for our data representation: Histogram Intersection [11], Chi-squared statistic [8] and Bhattacharyya distance [2], each of which has a well-defined probabilistic interpretation (in contrast to Euclidean distance norms on histograms, which in our opinion are hard to justify, despite their prevalent use). We validated and compared the performance of these 3 measures on the **VisTex** database [6] with a 58-class texture classification task and found that simple nearest-neighbor classification (using the above similarity measures) yielded acceptable performance (88-90% accuracy). We found no statistically significant difference between the 3 measures to justify selecting one over the other and all 3 were made available to the user in the browser interface.

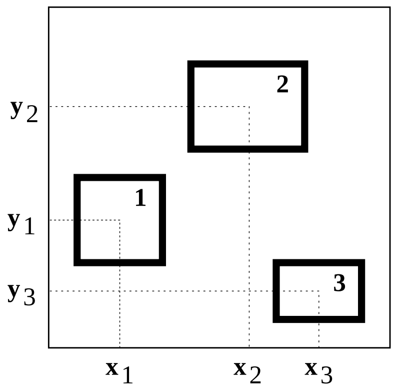
[1] This particular texture representation scheme was selected based on the encouraging image matching results obtained by Schiele & Crowley [9], but other texture features (*e.g.,* wavelet pyramid coefficients [10]) could also have been used.



H_{B_i}

$$H_R = \frac{1}{|R|} \sum_{B_j \in R} H_{B_j}$$

Figure 1: (a) An image block B_i with corresponding histogram H_{B_i} (b) A region R composed of individual blocks B_j and its "pooled" histogram H_R



$$x_1 < x_2 \ , \ x_1 < x_3 \ , \ x_2 < x_3$$

$$y_1 < y_2 \ , \ \overline{y_1 < y_3} \ , \ \overline{y_2 < y_3}$$

Figure 2: Three regions and the complete set of binary relationships corresponding to their spatial configuration.

2.3 Region Matching and Spatial Constraints

These non-parametric densities represent *local* color and texture and due to the additive property of histograms, can be easily combined (summed) to form densities for larger image blocks, including the entire image at which point they become identical to global histograms. When the user specifies a region of interest, its underlying block histograms are "pooled" to represent a "meta-block" histogram as illustrated in Figure 1. A region is then used to index into the database, where an online search for the best matching region (of the same size) is conducted using the aforementioned similarity metrics. Multiple region queries are processed in parallel and the best region match scores are then combined (usualy by summation) to determine the final visual similarity ranking. To speed up the online search, the entire database is first pruned to obtain a small subset (typically 5-10%) of "compatible" images using fast *global* histogram indexing.

In addition to querying by visual similarity, the user also has the option of specifying whether the selected regions should maintain their respective spatial configuration in the retrieved matches. We considered and briefly investigated various techniques for spatial representation and matching, including elastic spring models and graph matching. But in the end we opted for a much simpler formulation based on the consistency of binary relations on the centroid coordinates of the regions, as illustrated in Figure 2. Given the user-defined query Q, consisting of n regions, its spatial configuration similarity to a candidate configuration T (with

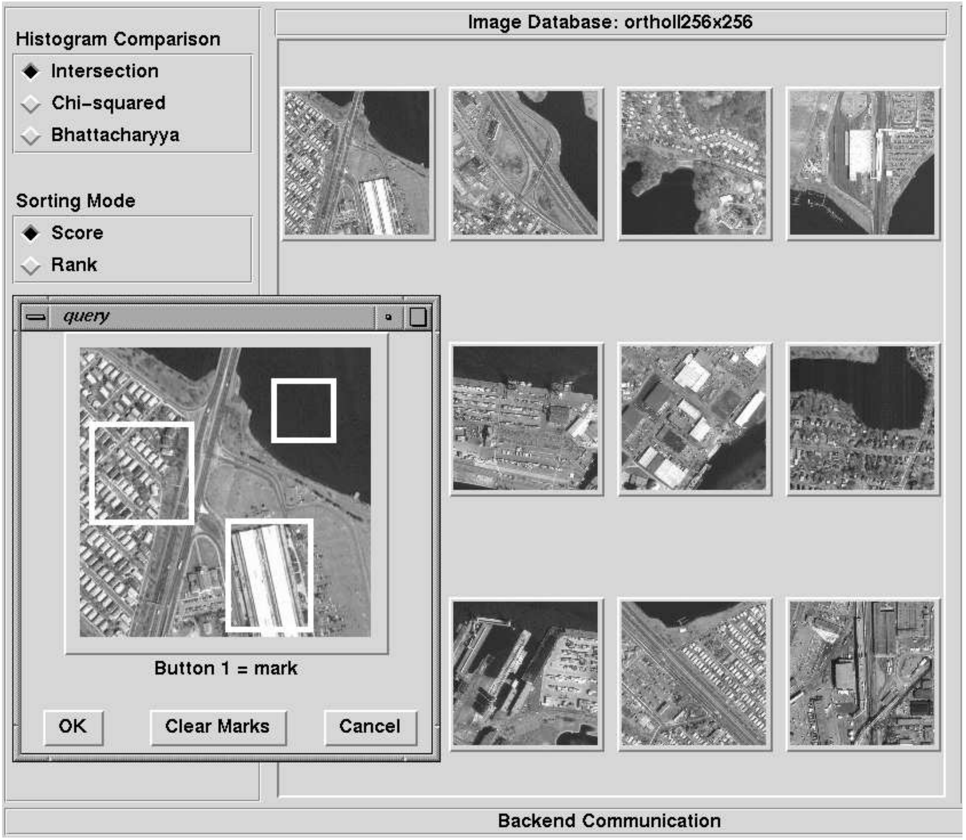


Figure 3: An example of a multiple ROI query with a database of B&W aerial imagery.

corresponding best matching regions) is given by

$$S(Q,T) \;=\; \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f(x_i^t - x_j^t)\ \mathrm{sign}(x_i^q - x_j^q) \;+\; f(y_i^t - y_j^t)\ \mathrm{sign}(y_i^q - y_j^q) \tag{1}$$

where x_i^q and x_i^t are the region centroid coordinates of the query Q and candidate T, respectively. The function f is a bipolar sigmoid (hyperbolic tangent) and its product with the sign function will essentially result in a "fuzzy" or "soft" count of the total number of satisfied constraints (in the set of binary relations) between Q and T. The scale parameter of the sigmoid function can be adjusted to specify how strictly a binary constraint is imposed (in the limit f can be made into a sign function as well). We note that this formulation is an approximate similarity measure as it assumes that the x and y coordinates of a region can be treated independently in determining the correct spatial relativity of two regions. Nevertheless, we have found it to be quick and easy to compute and quite adequate in measuring similarity of spatial configurations. Finally we note that the spatial similarity score is combined (typically by weighted summing) with the visual similarity score of all the regions to obtain a single final score by which the candidate entries in the database are ranked.

3 Results

One of the unfortunate aspects of our user-defined multiple ROI query method is that no automatic image self-matching is possible in order to perform large classification and retrieval experiments to quantify performance. Our technique is inherently *interactive* and *user-based*, thus requiring a human in the testing loop. In other words, "content" is no longer defined by the unique and fixed global attributes of database images, but rather by a myriad of user-defined queries all of which can exist within a single image.

Therefore, the only sensible performance measure is one that quantifies the user's overall "satisfaction" with the retrieved matches. Our experimental design was simple: 5 naive users were instructed in the basic operations of the multiple ROI query interface and asked to perform a minimum of 20 random queries on various databases.[2] Each user-defined region-based retrieval was immediately followed by a *global* search with the same query image afterwhich the user had to decide (forced choice) which set of retrievals (local or global) captured the "essence" of their intended content. The average percentage of acceptable local first-rank matches — which was found to be 73% — indicated that the local searches were indeed favored over global searches (50% would indicate no discernible difference or preference for local *vs.* global).

Figure 3 shows an example query in our browser, running on a database of GIS Orthophoto Imagery of the state of Massachusettes (available at http://ortho.mit.edu). The smaller window in the lower left allows the user to graphically define and edit (in this case) three regions corresponding roughly to "dense urban row housing", "water" and "factory" region types (note that these "classes" are entirely user-defined). The user can either retrieve images which respect the spatial configuration of the query, or alternatively, disable spatial scoring to simply retrieve images containing similar types of regions.

4 Discussion

Currently the online search for individual regions is computationally intensive and more sophisticated pruning strategies should be implemented in order to avoid searching every region of every image in the database. Global histogram indexing is partially effective in pruning the database size down to a reasonably small candidate set. Furthermore, search schemes exploiting hierarchical database organization (based on global and/or local features) should significantly decrease the size of the candidate set and hence the search time. Another speed-up possibility is to immediately reject candidate images based on partial spatial configurations (*e.g.,* if the best match for region 1 is already on the wrong side of region 2, reject the current image). While it may not be possible to rival the speeds of retrieval engines with pre-segmentation — like "Blobworld" [1] — we believe our system offers the flexibility of online user-designed queries, thus leading to more accurate representations of "content". Finally, our system should be useful not only for general image retrieval, but also for domain-specific databases such as the GIS aerial imagery example shown in Figure 3. Defining image content with multiple ROIs can be particularly useful in domain-specific retrieval: for example in medical applications, where both appearance and spatial factors play a significant diagnostic role.

References

[1] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Region-based image querying. In *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, June 1997.

[2] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, 1971.

[3] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis.* Chapman & Hall, 1995.

[4] N. Howe. Percentile blobs for image similarity. In *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, June 1998.

[5] J. Huang, S. K. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1997.

[6] MIT Media Laboratory. Vistex vision texture database, 1995.

[7] P. Lipson, E. Grimson, and P. Sinha. Configuration based scene classification and image indexing. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1997.

[8] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw Hill, 1991.

[9] B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. In *European Conference on Computer Vision*, volume 1, pages 610–619. ECCV, April 1996.

[10] S. Sclaroff, L. Taycher, and M. La Cascia. Imagerover: A content-based image browser for the world wide web. In *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, June 1997.

[11] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[2] Our collection of $O(10^4)$ images consists of separate databases of Corel stock photos, CD covers (from Amazon.com), GIS aerial imagery, 2D MRI medical images and a large and varied assortment of images gathered by crawling the web.