FOCUS : Searching for Multi-colored Objects in a Diverse Image Database*

Madirakshi Das   Edward M. Riseman
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
e-mail : mdas,riseman@cs.umass.edu

Bruce A. Draper
Department of Computer Science
Colorado State University
Fort Collins, CO 80523
draper@cs.colostate.edu

Abstract

We describe a new multi-phase, color-based image retrieval system, FOCUS (Fast Object Color-based qUery System), with an online user interface which is capable of identifying multi-colored query objects in an image in the presence of significant, interfering backgrounds. The query object may occur in arbitrary sizes, orientations and locations in the database images. The color features used to describe an image have been developed based on the need for speed in matching and ease of computation on complex images while maintaining the scale and rotation invariance properties. The first phase matches the color content of an image computed as the peaks in the color histogram of the image, with the query object colors using an efficient indexing mechanism. The second phase matches the spatial relationships between color regions in the image with the query using a spatial proximity graph (SPG) structure designed for the purpose. The method is fast and has low storage overhead. Test results with multi-colored query objects from artificial and natural domains show that FOCUS is quite effective in handling interfering backgrounds and large variations in scale. The experimental results on a database of diverse images highlights the capabilities of the system.

1 Introduction

With the advent of large image databases with complex images, efficient content-based retrieval of images has become an important issue. The aim is to find images in a database which contain the object represented in a query image. When the database has images of multi-colored objects which can be recognized on the basis of their distinctive color signatures alone, the color of the object is an obvious choice for indexing. The simpler problem of finding global similarity between a query image and candidate images based on color has been addressed in [3, 6, 9, 14]. However, multi-colored objects often will not occur by themselves in an image. There may be significant, interfering background clutter in the images or the object may occupy a very small portion of the overall image, making global similarity measures unsuitable for retrieval. In a general database, no assumptions may be made about background objects or the prominence of the query object in the database images in which it is present. This is the case studied here; the queried object may be embedded in very dissimilar images and may occupy a small portion of the image.

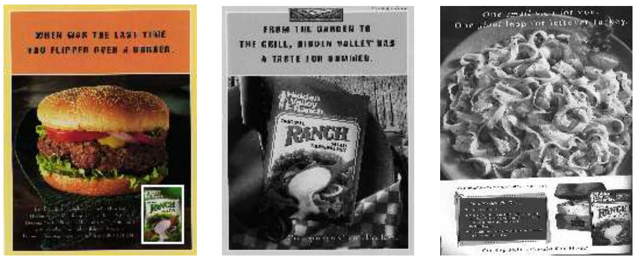


Figure 1: Example of database images showing the presence of the queried object (marked by a white box) at different scales and with different backgrounds

There are many objects where color can be a basis for retrieval – examples include flags, logos, consumer products, textile patterns and postal stamps among artificial objects and flowers, birds, fish and butterflies in the natural domain. The database on which FOCUS has been tested consists of 1200 diverse color images. There are 400 advertisements from magazines and 800 color images from nature including birds, fish, flowers, animals and vegetables. Advertisements are the most challenging component of the database where the goal is to retrieve all advertisements of a product shown in a query image. This is a particularly complex task since the queried object may appear in candidate images in various sizes and orientations with a wide variety of background colors and forms as shown in Fig.1. Unlike other databases on which color-based retrieval has

*This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by the United States Patent and Trademark Office and Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235, in part by the National Science Foundation, Central Intelligence Agency, Department of Defense (DARPA) and National Security Agency under grant number IRI-9619117, in part by ARPA (via USAF Rome Laboratory) under contract F30602-94-C-0042 and in part by NSF Multimedia CDA-9502639. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsors.

been tried (flags, logos, products), in most of the advertisement images, the products do not spatially dominate the image, nor are they necessarily in the center. There is no concept of foreground and background – what is background clutter for this application may actually be the foreground of the image. So, no focus-of-attention pre-segmentation is possible. In addition, most advertisements include natural backgrounds that do not provide sharp color boundaries. In the rest of the paper, background refers to all objects in the image which are not a part of the queried object.

There has been some work in color-based retrieval in the last six years. The earliest significant work in color-based retrieval is by Swain and Ballard who used *color histograms* for indexing [14]. More efficient histogram indexing strategies have been developed in [3, 4] and a more descriptive histogram-based retrieval has been described in [10]. The main drawbacks of histogram matching when applied to a complex database such as this arises from its sensitivity to scale variations and the presence of similar colored objects in the background. Most of the existing retrieval systems which use color [9, 1] are also sensitive to scale and background because they assume that the images to be retrieved are those in which the query object occupies the most prominent part. The histogram cluster-based matching described in [6] is scale invariant but does not handle the presence of interfering background colors in an image. The method proposed in [12] uses automatic region detection which will also be affected by background colors and miss objects when they are small.

Thus, even though color has been recognized as an important tool in content-based retrieval, fast color-based retrieval strategies which can handle interfering backgrounds and large variations in scale are not yet available. FOCUS (Fast Object Color-based qUery System) is motivated by the need to develop such a retrieval strategy.

2 Overview of the system

The speed and accuracy of a retrieval system will depend on the features used to describe the images and the matching strategy used. The feature used needs to provide discrimination between images which contain objects similar to the query and those which do not, while being invariant to differences in the scale, location and orientation of the query object in the candidate image, and the presence of background colors. It is also desirable for the feature to be indexable and allow fast matching. Color histograms [3, 14], color clusters [6] and color adjacency graphs [8] are some examples of color features which have been used. None of these color features satisfy *all* the requirements for a good matching strategy for a retrieval engine. It is difficult, if not impossible, to find a single color feature which will satisfy all the requirements listed above.

We have used two scale and orientation invariant color features, combining them in a two-phase matching strategy to achieve fast but accurate retrieval. The emphasis in the first phase of matching is on speed, and the second phase aims at removal of false matches from the image list produced by the first phase.

During the first phase, database images which have

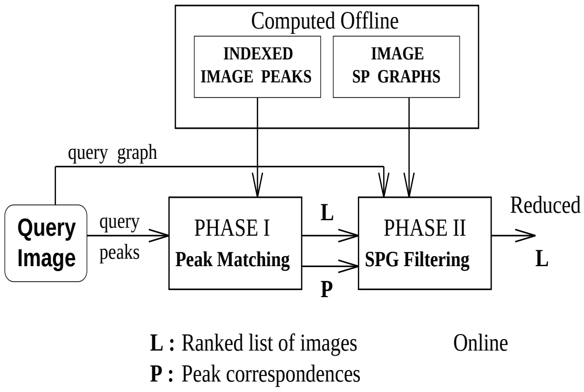


Figure 2: System Overview

all the colors of the query image are extracted using an index structure computed offline. During the second phase, evidence supporting the hypothesis that a candidate image contains the query object is generated by detecting the spatial color relationships of the query object in the image. This is achieved without any pixel level processing of the image, using a graph description of the image described in section 4. The second phase acts as a filter on the list obtained by the first phase, deleting the images where no such evidence is obtained. The retrieval results can be examined by the user after the first phase and a decision made on whether the second phase of processing is needed depending on the number of false matches produced.

3 Phase I: Peak Matching

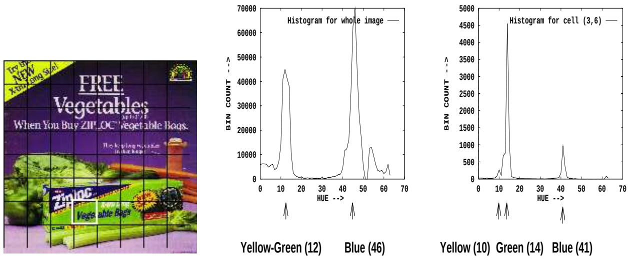
The first phase of matching is intended to produce a candidate image list as quickly as possible. The simplest requirement for an image to contain a multi-colored query object is the presence of all the colors of the query. The color content of an image is described by listing all the distinct colors in the image computed as the *peaks* in the 3-D color histogram of the image. Since describing an image by its histogram peaks is equivalent to the case where very few bins in a histogram have non-zero values, this gives us more discrimination power as shown in [13]. The storage space required is reduced and fast matching is possible, when compared to using the full histogram. Most importantly, the locations of peaks in a histogram are stable under viewpoint change and scale transformation, unlike histogram bin counts.

3.1 Detection of histogram peaks

There is a sharp peak in the color histogram of an image corresponding to distinct, chromatically pure color regions e.g. flags, commercial products etc. However, images containing natural scenes and people, produce wide peaks in the histogram. So even when the query object has distinct colors, in the database image, the peaks corresponding to the query colors may be masked or shifted by the wide peaks from background colors. For example, Fig.3(b) shows the global hue histogram of the image (Fig.3(a)) of a "Ziploc" bag along with different shades of green and yellow in the background. The histogram in Fig.3(c) of an area which covers the "Ziploc" package only, shows the actual peak locations of the colors present on the package. These peaks are lost in the global histogram, being subsumed by the colors in the background. This example also suggests a solution to the problem. The color peaks present in an image can be determined more accurately when the histogram covers a small area of the image, reducing the effect of the presence of interfering colors.

We use a *split and merge* strategy for peak detection. Since we do not know the size or the location of the object of interest in the image a priori, the image is divided uniformly into m x n cells. Local histograms are constructed in the image cells and peaks are detected in the local histograms. A combined list of peaks is produced by merging multiple copies of the same peak, and a *peak color label* is assigned to each peak which is unique for that image. For this database, a cell size of 100x100 pixels is used. Splitting the image also localizes the color peaks in a cell of the image and this information is used during the next phase of matching.

During offline processing, peaks are detected for all the images in the database. Peak detection requires a number of practical choices to be made. For this work, we use the HSV (hue, saturation, value) color space, since it is more stable than RGB under variations in illumination. In addition, since the hue component is the most stable, the HSV histograms are finely discretized along the hue axis and coarsely divided along saturation and value axes (64x10x10). When constructing histograms, all "grey" (ranging from black to white) pixels in the image are ignored, since these pixels map to arbitrary hue locations in HSV space. Very little discrimination power is lost because grey pixels are present in almost all images, usually as background, text or shadows.



(a) (b) (c)

Figure 3: Effect of interfering background on histogram peak location: (a) Database image (b) Global hue histogram of the image (c) hue histogram of cell marked in (a)

The histogram peaks are detected by finding *local maxima* in a 3-D neighborhood window. Though the histogram clustering approach [6] produces fewer clusters, cluster means may not be representative of the whole cluster, as smaller peaks close to larger ones e.g. yellow and green in Fig.3(c), may be merged into a single cluster. Since the query object colors are not known a priori, it is necessary to have peaks representing every distinct color.

3.2 Indexing and Retrieval

The aim of indexing is to narrow the search to the images which could match the given query peaks. When looking for color similarity, approximate matching is necessary so that the matching process is robust to small variations in color. For a given query peak P_q (h_q, s_q, v_q), we need to find all images which contain a peak in the *neighborhood* of P_q. This requires an order-preserving indexing structure which supports range queries. We have used a $B+$ tree [7] to store the peaks in the database sorted with hue as the primary key, followed by saturation and value. A *frequency table* is also constructed which gives the number of images which will be retrieved for each point in the discretized HSV space. For each peak in the query, a range query of $(h_q \pm 3, s_q \pm 4, v_q \pm 5)$ is executed starting with the peak which retrieves the minimum number of images onwards i.e. the rarest among the query peaks is used as the first query. A *join* of the lists of image identifiers is taken to find the images which have peaks matching *all* query peaks. The time complexity of the retrieval process is given by $O(qlog(kN))$, where q is the number of query peaks, N is the total number of images in the database and k is the average number of peaks per image (12 for this database). The join process is linear in the size of the lists retrieved.

The list of images obtained is ordered by increasing mismatch score. The mismatch score is given by the sum of the *city block* distances between each query peak and the candidate peak which it matched. For each image in the list, the correspondence between the peak color label and the query color label which it matched is noted for use in the second phase.

4 Phase II matching strategy

A ranked list of candidate images is obtained at the end of the first phase of matching. A more computationally intensive matching strategy can be applied at this stage since these operations need to be carried out only on the candidate images from the first phase. Even with this reduction, pixel level processing of the images would make this phase of matching too slow for online user interfaces. In response to this problem, we have developed a new graph description of the spatial relationship between color regions which is easy to compute and match.

4.1 Construction of SPG

There will be a number of false matches in the image list retrieved by the first phase of matching in which all the colors of the query object are present but not in the same spatial configuration as in the query object. In the extreme case, the matched colors are scattered across the image and do not form any connected cluster. In other cases, some color adjacency relationship present in the query object may be violated in candidate images. For example, in the query in Fig.6(a), the red (peak color label 0) and blue (peak color label 3) regions are adjacent whereas in the false match (b), they are not adjacent. These false matches could be eliminated if information on spatial distribution of colors in the image was available.

The color adjacency graph (CAG) formulation described in [8] is a good descriptor of the color relationships in a multi-colored object where the color regions are nodes in a graph with edges connecting color regions which share an edge at the pixel level. However, a CAG description of the database images is not feasible for retrieval due to the complexity of the images. Most of the images contain natural objects and color regions in which there are no distinct boundaries between colors. For example, the images shown in Fig.1 are quite typical of the images in the database. An attempt to construct a CAG for these images would produce very large, complex graphs making the matching phase intractable. The spatial description in [5] using color partitions is also very computationally intensive when invariance to orientation and position is required. Therefore, we need a simpler representation for the spatial distribution of colors that allows efficient generation and matching for all types of images.

It should be noted that during the peak detection process, we have already localized color peaks in image cells, giving us the color content in each cell. We now use this information to construct a graph describing the *approximate* spatial relationships between colors in the image *without any additional pixel level processing.* The aim is to capture all possible adjacencies between color regions.

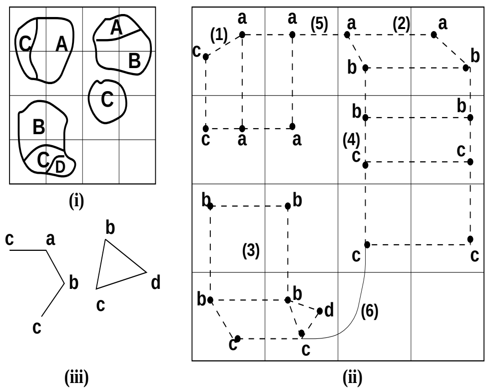


Figure 4: Example of SPG construction (i) Synthetic image divided into cells (ii) Cells marked with nodes (peaks) contained in them. The intermediate graph is shown in broken lines (iii) SPG constructed from (ii)

We start by constructing an intermediate graph representation directly from the peak description of the image based on whether pixel level adjacency is *possible* between two color regions. Fig.4 is used to explain how the spatial relationships between color regions can be inferred from the color peak description and condensed into a compact graph – the *spatial proximity graph* (SPG). Fig.4(i) shows a synthetic image of objects with four color regions (A,B,C,D) producing peaks which are labelled (a,b,c,d) respectively. The peaks detected in each cell are shown in Fig.4(ii) by including the peak color label within the cell. These peaks form nodes in the intermediate SPG. Edges in the intermediate SPG indicate that the two peaks could be from adjacent color regions in the original image and are generated from the following observations.

• When two nodes occur in the same cell, they could be from adjacent color regions in the original image, so they are connected by an edge e.g. edge labelled (1). Some nodes may be connected which are not actually adjacent e.g. edge (4), but we cannot determine the exact adjacency relation within a cell without pixel level processing which is to be avoided here.

• Identically labelled nodes in neighboring cells could be a part of the same color region in the image and therefore are connected by an edge e.g. edge (2). The edge labelled (5) shows a case where two nodes are connected but are actually not from the same color region. If two regions of different colors are adjacent, there will be at least one cell where peaks from both the regions will be present *together* and therefore will be connected by an edge in *that* cell. So nodes of different color labels are not connected across cell boundaries.

• Diagonal edges are not considered because they would be redundant e.g. (3), and may add some edges where no adjacency is possible e.g. line (6). If two color regions are adjacent, they cannot have peaks only along a diagonal since there is just a single pixel of contact between two cells along the diagonal.

Putting the above discussion concisely, let nodes of the intermediate SPG be of the form c_m^i, where m is the peak color label of the node and i is the cell in which it is located. There is an edge E between two nodes of the graph if the following condition is met.

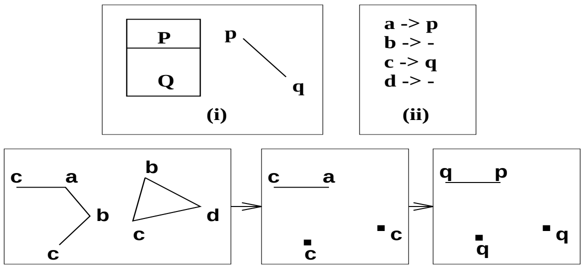
• $E(c_m^i, c_n^j)$ if $(i = j)$ OR $(m = n$ and (i, j) are 4-neighbors).

The intermediate graph obtained is not scale invariant, since a larger region would produce more nodes in the graph. The smaller, scale invariant SPG which still captures the spatial relationships between colors is obtained by *collapsing* connected nodes of the same color label into a single node of that color label. The graph may still have multiple nodes of the same color label, but only if these peaks were spatially disconnected in the image. Fig.4(iii) shows the SPG obtained by collapsing the intermediate graph in Fig.4(ii). The SPG is computed offline for all database images and stored using an adjacency matrix representation.

The spatial proximity graph (SPG) description has a number of very useful properties. Apart from being scale and orientation invariant, it can be computed easily for all types of images, with or without prominent color boundaries. The SPG shows all possible pixel-level adjacencies that could occur in an image, without going through pixel-level processing, but adds some false adjacencies as well. This is handled in the matching phase by detecting subgraph isomorphism instead of exact graph matches.

4.2 Matching SPGs

The problem tackled during the online second phase is to detect if the query color graph occurs as a subgraph of the candidate image SPG. However, the whole image SPG need not be used. At the end of the first phase of retrieval, the correspondence between color labels in the image and the query peaks are available for each image in the retrieved list. The color label of the nodes in the image SPG are replaced by the query peak number they matched using the available correspondence. Any node from the image SPG which



(iii)

Figure 5: SPG filtering on the synthetic example in Fig.4 (i) Query image and graph (ii) Correspondence with query peaks obtained from first phase of matching (iii) Construction of reduced SPG by deleting unmatched peaks and relabelling nodes

does not match a query peak are removed. Fig.5 shows the process of constructing the reduced SPG from the SPG computed offline.

The query graph needs to be found in this reduced image SPG. This is an instance of the subgraph isomorphism problem which is known to be NP-complete. However, due to the restricted nature of this problem where the reduced SPG nodes are labelled with the same labels as the query graph, the matching computation is feasible. The running time is of the order of $O(n^m)$ where n is the size of the query adjacency matrix and m is the maximum number of instances of a color label in the reduced SPG, typically 3 or less.

The search time is further reduced by starting the matching process with the query peak which has the minimum number of instances in the reduced SPG of the image. Fig.6 shows an example query graph and the reduced image SPG where a false match is detected. The four query peaks are labelled 0-3, and the reduced SPG nodes are labelled with the same numbers. For clarity, the labels have been placed in the region of the image where the peaks were detected.

(a) (b)

Figure 6: Example of SPG filtering (a) "Blueberry Morning" query image with SPG superimposed (b) A false match with SPG superimposed

5 Query construction and processing

A query can be selected by marking a sub-image [11] which covers the object of interest from a database image. The query image should not contain any background colors and include the salient colors of the object.

Since there is no background included in the query, the query color peaks are computed from the global histogram of the query image. However, it is not sufficient to describe the color adjacencies in the query using the SPG. For a fixed cell size, a small query image may have two peaks in the same cell and thus an edge between them in its SPG. However, when a bigger copy of the query object appears in a database image, these two peaks could now be in separated cells with no edge between them. This would create a mismatch with the query graph. On the other hand, if there was adjacency at the *pixel level* between two colors in the query image, this would be reflected in the SPG of the database image even if the query object was of a much larger size in the database image. So pixel level adjacencies are computed for the query colors to maintain the scale invariance property. More details of the query processing phase are available in [2].

6 Results

The database has 1200 images of various sizes up to 2.5Mb. The overhead incurred in storing the required color information extracted offline is about 5% of the size of the database. The retrieval results obtained by this system can be judged by the criteria used in text retrieval, *precision* and *recall*. Precision is the proportion of correct retrievals in the images retrieved up to the last correct image. Recall is the proportion of images retrieved out of all the images in the database that should have been retrieved for the given query. On a query set of 25, the recall was 90%. The average precision after phase 1 was 64% and after phase 2 it improved to 75%.

The time taken for a complete cycle of retrieval consists of the query processing time, phase 1 matching and phase 2 matching. FOCUS runs on a 133 MHz Pentium processor and all times mentioned are averaged over many trials. Query processing takes about 0.1 sec on a query of size 100x200, which is the average size of queries tried. Phase 1 matching takes 0.1-0.2 sec and increases only logarithmically with the size of the database. Phase 2 matching takes about 0.01 sec for each image in the list produced by phase 1 which currently has 30 images on an average. The retrieval process is fast enough to be scalable to large databases.

A few sample retrieval results are shown in Fig.7 with the query marked by a white box. These results show good retrieval performance even when the query object is present in different sizes and with different backgrounds in the candidate images. A demonstration of FOCUS is available over the world wide web at `http://vis-www.cs.umass.edu/~mdas/color_proj.html`.

7 Conclusion

We have presented a fast image retrieval system which produces good results with multi-colored query objects and is robust to wide variations in scale and the presence of interfering backgrounds. The speed of the system and the small storage overhead make it suitable for use in large databases with online user interfaces. In the future, we plan to test FOCUS on a larger database and add more phases to utilize other types of image information.

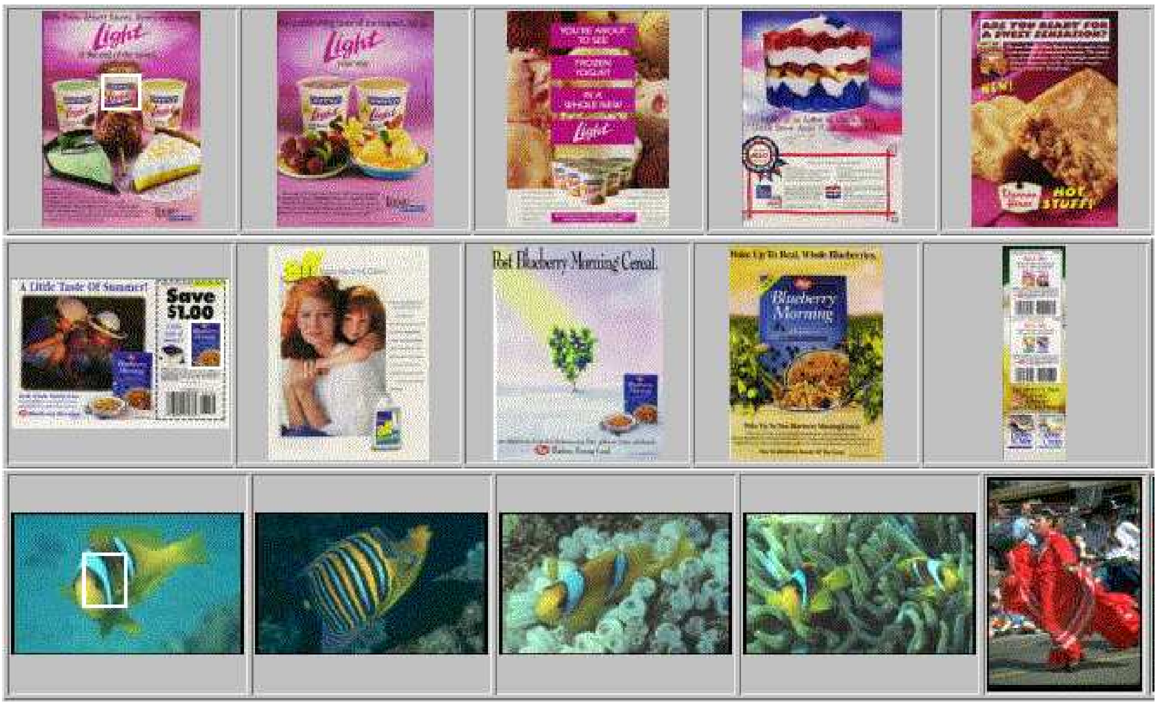


Figure 7: First five retrieved images for three different queries in order of rank with the query marked by a white box.

Acknowledgements

We would like to thank R.Manmatha for testing the retrieval results and providing valuable suggestions and Jonathan Lim for building the web-page for the online demonstration of the system. This work is a part of the Multimedia Indexing and Retrieval group, a joint effort of the Center for Intelligent Information Retrieval and the Computer Vision Laboratory at UMass.

References

[1] J.R.Bach, C.Fuller, A.Gupta et al, "The Virage Image Search Engine: An open framework for image management", *Proceedings of SPIE, Vol. 2670*, pp 76-87, Feb 1996.

[2] M.Das, B.Draper, W.Lim et al, "A Fast Background-Independent Retrieval Strategy for Color Image Databases", *Technical Report TR96-79*, Computer Science, UMass, 1996.

[3] Y.Gong, C.H.Chuan and G.Xiaoyi, "Image Indexing and Retrieval Based on Color Histograms", *Multimedia Tools and Applications, Vol 2, No. 2*, pp 133-156, Mar 1996.

[4] J.Hafner, H.Sawhney, W.Niblack et al, "Efficient Color Histogram Indexing for Quadratic Form Distance Functions", *IEEE Transactions on PAMI, Vol. 17, No. 7*, pp 729-736, July 1995.

[5] W.Hsu, T.S.Chua and H.K.Pung,"An Integrated Color-spatial Approach to Content-based Image Retrieval", *ACM Multimedia Conference*, pp 305-313, 1995.

[6] M.S.Kankanhalli, B.M.Mehtre and J.K.Wu, "Cluster-Based Color Matching for Image Retrieval", *Pattern Recognition, Vol. 29, No. 4*, pp 701-708, Apr 1996.

[7] H.F.Korth and A.Silberschatz, "Database System Concepts", *McGraw-Hill*, 1991.

[8] J.Matas, R.Marik and J.Kittler, "On Representation and Matching of Multi-Coloured Objects", *Fifth International Conference on Computer Vision*, pp 726-732, June 1995.

[9] W.Niblack, R.Barber et al, "The QBIC project: Querying Images by Content using Color, Texture and Shape", *Proceedings of SPIE, Vol. 1908*, pp 173-87, Feb 1993.

[10] G.Pass and R.Zabih, "Histogram Refinement for Content-Based Image Retrieval", *Workshop on Applications of Computer Vision*, pp 96-102, 1996.

[11] S.Ravela, R.Manmatha and E.M.Riseman, "Image Retrieval Using Scale Space Matching", *European Conference on Computer Vision*, pp 273-282, 1996.

[12] J.Smith and S.F.Chang,"Tools and Techniques for Color Image Retrieval", *Proceedings of SPIE, Vol. 2670*, pp 426-437, Feb 1996.

[13] M.A.Stricker, "Bounds for the Discrimination Power of Color Indexing Techniques", *Proceedings of SPIE, Vol. 2185*, pp 15-24, Feb 1994.

[14] M.J.Swain and D.H.Ballard, "Color Indexing", *International Journal of Computer Vision, Vol. 7:1*, pp 11-32, 1991.