VisualSEEk: a fully automated content-based image query system

John R. Smith and Shih-Fu Chang
Department of Electrical Engineering and
Center for Image Technology for New Media
Columbia University
New York, N.Y. 10027 USA
{jrsmith, sfchang}@itnm.columbia.edu

ABSTRACT

[1]We describe a highly functional prototype system for searching by visual features in an image database. The VisualSEEk system is novel in that the user forms the queries by diagramming spatial arrangements of color regions. The system finds the images that contain the most similar arrangements of similar regions. Prior to the queries, the system automatically extracts and indexes salient color regions from the images. By utilizing efficient indexing techniques for color information, region sizes and absolute and relative spatial locations, a wide variety of complex joint color/spatial queries may be computed.

KEYWORDS: image databases, content-based retrieval, image indexing, similarity retrieval, spatial query.

1 INTRODUCTION

In this paper we investigate a new content-based image query system that enables querying by image regions and spatial layout. VisualSEEk is a hybrid system in that it integrates feature-based image indexing with spatial query methods. The integration relies on the representation of color regions by color sets. For one, color sets provide for a convenient system of automated region extraction through color set back-projection. Second, the color sets are easily indexed for retrieval of similar color sets. As a result, unconstrained images are decomposed into near-symbolic images which lend to efficient spatial query.

We present the strategies for computing queries that specify the colors, sizes and arbitrary spatial layouts of color regions, which include both absolute and relative spatial locations. We describe the extraction process and present the compact color set representation of regional color features. We also address several special case spatial queries involving adjacency, overlap and encapsulation of regions. Finally, we evaluate the VisualSEEk color/spatial queries and demonstrate the improved image query capabilities over non-spatial content-based approaches.

[1]ACM Multimedia 96, Boston, MA, November 20, 1996

1.1 Content-based Image Query

The objective of content-based image query is to efficiently find and retrieve images from the database that satisfy the criteria of similarity to the user's query image. When the database is large and the image features are complex the exhaustive search of the database and computation of the image similarities is not expedient. Therefore, recent techniques have been proposed to speed-up image retrieval by utilizing simple image features, such as color histograms [1], devising compact representations [2][3], efficient indexing structures [4], and utilizing effective pre-filtering techniques [5]. However, recent approaches for content-based image query have neglected one important criteria for similarity: spatial information and spatial relationships.

1.2 Spatial Image Query

A significant aspect of discriminating among images depends on the sizes, spatial locations and relationships of objects or regions. However, introducing multiple image regions and spatial information into the query process greatly complicates the content-based image query approaches. This results from the combinatorial explosion in exhaustively comparing multiple regions or objects.

On the other hand, by representing images symbolically, spatial query methods utilize the locations and spatial relationships of symbols. For example, comparisons based upon 2-D strings [6], spatial quad-trees [7] and graphs [8] provide for flexible and efficient spatial querying. However, these approaches cannot easily accommodate measures of similarity of the "symbols" such as those based upon visual features of objects or regions. The problem has only recently been addressed in limited applications, for maps in [9] and for medical images that contain both labeled and unlabeled regions [4].

1.3 Joint Content-based/Spatial Image Query

In this work, we propose a new system that provides for both feature comparison and spatial query for unconstrained color images. To illustrate, as depicted in Figure 1(b) and (c), each image is decomposed into regions which have feature properties, such as color, and spatial properties such as size, location and relationships to other regions. In this way, images are compared by comparing their regions. Furthermore, the system gives

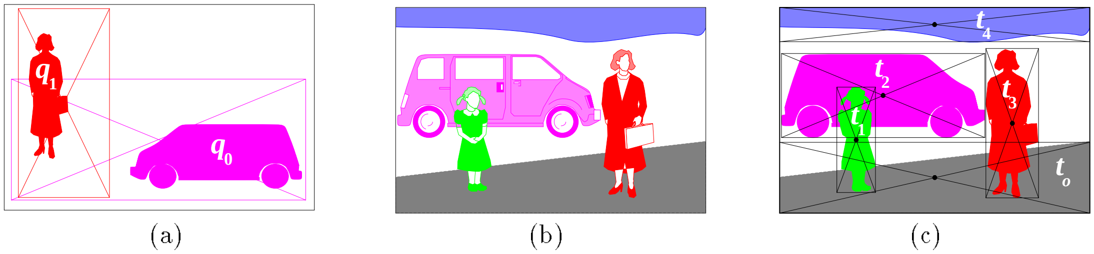


Figure 1: Image decomposition (a) query image, $Q = \{q_0, q_1\}$, (b) example target image, (c) target image decomposed into regions that have both feature and spatial properties, $T = \{t_0, t_1, \ldots, t_4\}$.

the user control in selecting the regions, see Figure 1(a), and parameters that are most important in determining similarity in a given query. As such, the system accommodates partial matching of images as determined by the user.

The most powerful type of image search system allows users to flexibly query for images by specifying both visual features and spatial properties of the desired images. Several recent content-based image query systems do not provide for both types of querying. The QBIC system [3] provides querying of whole images and manually extracted regions by color, texture and shape but not spatial relationships. The Virage system [10] allows querying of only image global features such as color, composition, texture and structure. Several new techniques have been proposed for injecting small amounts of spatial information into the image feature sets.

Related Work

A recent approach by Stricker and Dimai [2] divides each image into five fuzzy regions which contribute to the color moment representation of the image's color. The authors obtain compact feature sets and also allow the user to assign weights to the five spatial regions. In this way, the technique provides for querying by the five absolute region locations. However, it is not possible to query for images by specifying either arbitrary regions or the spatial relationships of regions.

Pass, Zabih and Miller [11] devised a technique which splits a global image histogram into coherent and scattered components. The measure of color coherence identifies the existence of connected colored regions. Although the technique improves on color histogram indexing, it does not support querying by the spatial locations of the color regions. Jacobs, Finkelstein and Salesin [12] devised an image match criteria and system which uses spatial information and visual features represented by dominant wavelet coefficients. Their system allows the user to sketch example images and provides for improved matching over image distance norms. However, their technique provides for little flexibility in specifying approximate and relative spatial information.

Our Approach

In order to fully integrate content-based and spatial image query capabilities we need to devise an image similarity function which contains both color feature and spatial components. We first note that perceived image similarity consists of both intrinsic and derived parameters. For example, the intrinsic part of a match refers to the similarity between query and target colors and/or region sizes and spatial locations. The derived part refers to the inferences that can be made from the intrinsic parameters, such as relative spatial locations and the overall assessment of image matches consisting of multiple regions.

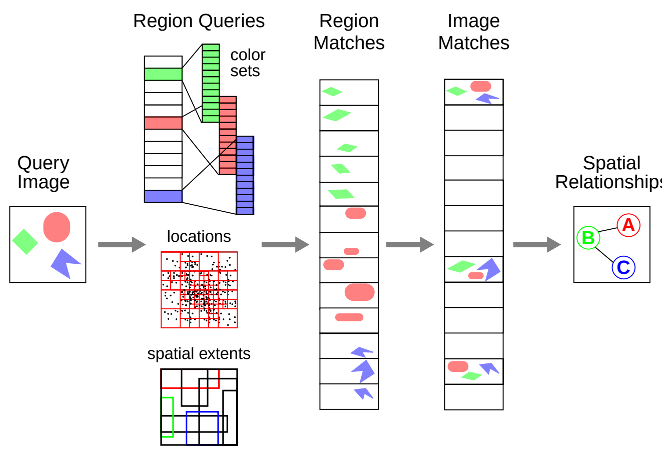


Figure 2: Image query process. Indexing of intrinsic region features: color sets, spatial locations, sizes and spatial extents represented by minimum bounding rectangles. Computation of image matches and evaluation of spatial relationships.

The joint color/spatial images query strategy is summarized in Figure 2. In order to quickly process queries, we design the representations for the intrinsic parameters such as region color, spatial location and size to require minimal computation in matching. For example, color matching is achieved efficiently through color sets. The intrinsic parameters are indexed directly to allow for maximal efficiency in queries. This query process identifies candidate regions which are combined to determine image matches.

In this way, a query specified by the user is translated directly into pruning operations on intrinsic parameters. The derived parameters, such as region relative locations and special spatial relations are resolved only in the final stage of the query. This is because these evaluations have the highest complexity. The pruning performed by the queries on the intrinsic parameters reduce the

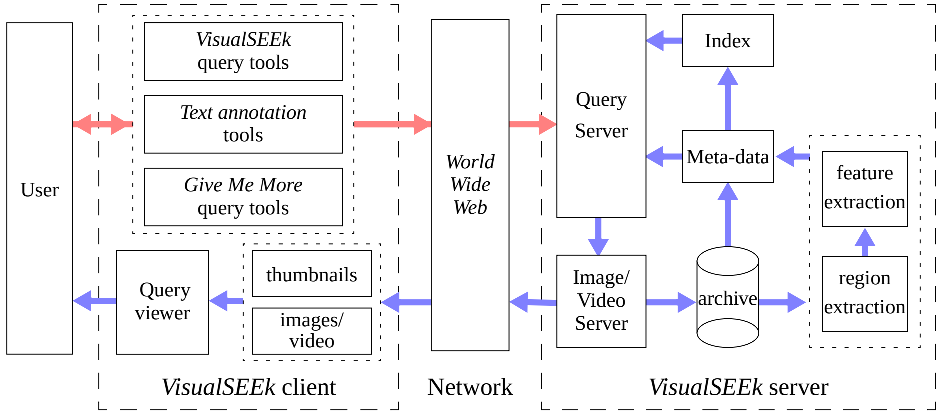


Figure 3: VisualSEEk system overview.

number of candidates images that need to be evaluated at the final stages.

1.4 Unique Features of VisualSEEk

- joint content-based/spatial querying
- automated region extraction
- direct indexing of color features

The VisualSEEk project[2] has emphasized several unique objectives in order to improve image retrieval, such as: (1) automated extraction of localized regions and features [13], (2) querying by both feature and spatial information [14], (3) feature extraction from compressed data [15], (4) development of techniques for fast indexing and retrieval and (5) development of highly functional user tools.

The VisualSEEk client application was developed in the *Java* language to allow for client platform independence and accessibility on the *World Wide Web*. As illustrated in Figure 3, the VisualSEEk system consists of several components: the set of user tools, the query server, the image and video server, the image and video archive, the meta-data database and the index files. Currently, the VisualSEEk system allows searching for images in a test-bed of 12,000 color images. The VisualSEEk tools are also being ported to an application for searching in a collection of over one million images and videos from the World-Wide Web [16].

In the next section, we present the color set representation and back-projection region extraction process. In section 3, we examine the process of querying by color similarity. In section 4, we extend the color query process to include the matching of regions that have properties of absolute location, color and size. In section 5, we consider the case of matching images with multiple regions that have properties of color, size, absolute and relative locations, which includes cases of special relations between regions. Finally in section 6, we provide an evaluation and present some examples of querying by joint color/spatial features.

[2] http://www.itnm.columbia.edu/VisualSEEk

2 COLOR SETS AND BACK-PROJECTION

Color sets provide a compact alternative to color histograms for representing color information. Their utilization stems from the conjecture that salient regions have not more than a few, equally prominent colors. The following paragraphs define color sets and explain their relationship to color histograms.

Color Sets

A key issue in defining the color representation is the choice of color space and how it is partitioned. Three dimensions of color can be defined and measured. For example, each image point can be represented as a 3-D vector $\hat{v}_c = (r, g, b)$ in the RGB color space. The transformation T_c and quantization Q_c^M of the RGB color space reorganizes and groups the vectors $\hat{v}_c$. Perceptually distinct colors correspond to the sets of vectors that are mapped to different indices m as diagrammed in Figure 4. Color sets are defined as follows: let $\mathcal{B}_c^M$ be the M dimensional binary space such that each axis in $\mathcal{B}_c^M$ corresponds to one unique index value m. A *color set* is a binary vector in $\mathcal{B}_c^M$ which corresponds to a selection of colors $\{m\}$.

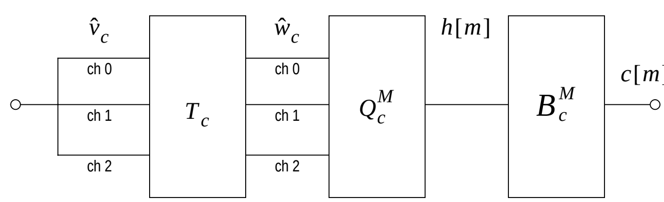


Figure 4: Generation of the binary color space from color channels, color space transformation T_c, quantization Q_c^M, histograms $h[m]$, and color sets $c[m]$.

Color Set Example

For example, let T_c transform RGB to HSV and let Q_c^M where $M = 8$ quantize the HSV color space to 2 hues, 2 saturations and 2 values. The quantizer Q_c^M assigns a unique index m to each quantized HSV color. Then, $\mathcal{B}_c^8$ is the eight dimensional binary space whereby each element in $\mathcal{B}_c^8$ corresponds to one of the quantized HSV colors. A color set $\mathbf{c}$ contains a selection from the eight colors. If the color set corresponds to a unit length binary vector, then one color is selected. If a color set has more than one non-zero value, then several colors are selected. For example, the color set $\mathbf{c} = [10010100]$ corresponds to the selection of three colors, $m = 0$, $m = 3$ and $m = 5$, from the quantized HSV color space.

For implementation in VisualSEEk, we choose T_c to transform RGB to the HSV color space because HSV provides a breakdown of color into its most natural components: hue, saturation and intensity. We choose Q_c^M, as illustrated in Figure 5, to quantize HSV into $M = 166$ colors [13].

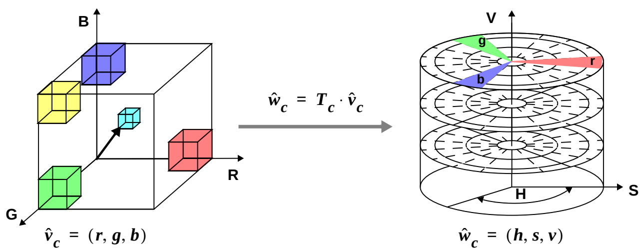


Figure 5: Transformation T_c from RGB to HSV and quantization gives 18 *hues*, 3 *saturations*, 3 *values* and 4 *grays* = 166 colors.

Color Set Back-Projection

We use a color set back-projection technique in order to extract color regions. We briefly describe the technique here and note that a more detailed presentation appears in [13]. The back-projection process requires several stages: color set selection, back-projection onto the image, thresholding and labeling. Candidate color sets are selected first with one color, then with two colors, etc., until the salient regions extracted.

The back-projection of a color set is accomplished as follows: given image $I[x, y]$ and color set $\mathbf{c}$, let k be the index of the color at image point $I[x, y]$, then generate image $B[x, y]$ by

$$B[x, y] = c[k], \quad \text{binary back-projection.} \tag{1}$$

That is, $B[x, y]$ depicts the back-projection of color set $\mathbf{c}$. In order to accommodate color similarity in the back-projection process, a correlated back-projection image can be generated by

$$B[x, y] = \max_{j=0\ldots M-1}(A_{j,k} c[j]), \quad \text{correl. back-projection,} \tag{2}$$

where $A_{j,k}$ measures the similarity of colors j and k, which will be discussed in the next section. After back-projecting the model color set, image $B[x, y]$ is filtered and analyzed to reveal spatially localized color regions. The process and back-projection results are illustrated for an example image in Figure 6.

Information about the regions such as the color set used for back-projection, the spatial location and size are added to the `REGION` relation (see Table 1) and are subsequently used for queries as explained in the next sections. While color set selection and back-projection provides one automated technique for extracting salient color regions from the images, other methods can easily be incorporated into our system. For example, manually extracted regions and their features can also be added to the `REGION` relation.

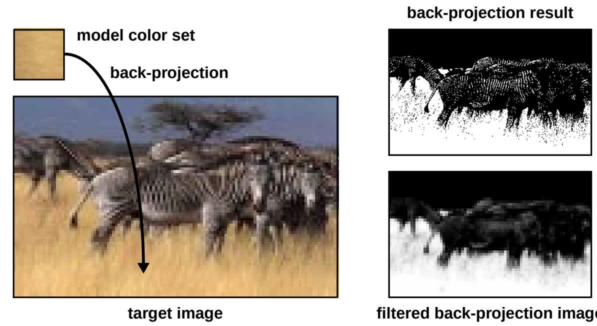


Figure 6: Example back-projection using model color set at top left to extract color region from a target image.

$\overline{\texttt{IMID}}$	$\overline{\texttt{REGID}}$	c	x	y	area	w	h
0001	0001	$[01\ldots0]$	18	63	430	30	15
0001	0002	$[11\ldots1]$	34	45	968	65	32
0002	0001	$[00\ldots1]$	76	54	780	53	42
0003	0001	$[11\ldots0]$	55	12	654	43	55

Table 1: The `REGION` relation with attributes for color set $\mathbf{c}$, region centroid (x, y), region size `area` and width and height (w, h) of the minimum bounding rectangle (MBR).

3 COLOR QUERY

We first describe the procedure for computing the similarity of colors and color distributions. Then we describe the process of indexing and querying by color.

3.1 Color Similarity

In order to match color regions, we need a measure for the similarity of colors, i.e., pink is more similar to red than blue. We base the measurement of color similarity on the closeness in the HSV color space as follows: the similarity between any two colors, indexed by $m_i = (h_i, s_i, v_i)$ and $m_j = (h_j, s_j, v_j)$, is given by

$$a_{i,j} = 1 - 1/\sqrt{5}\Big[\begin{array}{l} (v_i - v_j)^2 + \\ (s_i \cos h_i - s_j \cos h_j)^2 + \\ (s_i \sin h_i - s_j \sin h_j)^2\Big]^{\frac{1}{2}} \end{array} \tag{3}$$

which corresponds to the proximity in the cylindrical HSV color space depicted in Figure 5. The measure of color similarity, $a_{i,j}$, is used within the computation of the distance between color distributions as described next.

Color Histograms

A distribution of colors is defined by a color histogram. By transforming the three color channels of image $I[x, y]$ using transformation T_c and quantization Q_c^M as defined in Section 2, where $(\hat{v}_c)_{xy} = (I_A[x, y], I_B[x, y], I_C[x, y])$, the single variable color histogram is given by, where X

and Y are the width and height of the image, respectively, which are used for normalization,

$$h[m] = \frac{1}{XY} \sum_{x \in X} \sum_{y \in Y} \begin{cases} 1 & \text{if } Q_c^M T_c(\hat{v}_c)_{xy} = m \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Histogram Distance

The most common dissimilarity measures for feature vectors are based upon the Minkowski metric, which has the following form, where $\mathbf{h}_q$ and $\mathbf{h}_t$ are the query and target feature vectors, respectively,

$$d^r_{q,t} = (\sum_{m=0}^{M-1} |h_q[m] - h_t[m]|^r)^{1/r}. \quad (5)$$

For example, both the L_1, $(r = 1)$ [1], and L_2, $(r = 2)$, metrics have been used for measuring dissimilarity of histograms. However, histogram dissimilarity measures based upon the Minkowski metric neglect to compare similar colors in the computation of dissimilarity. For example, using a Minkowski metric, a dark red image is equally dissimilar to a red image as to a blue image. By using color similarity measures within the distance computation, a quadratic metric improves histogram matching.

Histogram Quadratic Distance

The QBIC project uses the histogram quadratic distance metric for matching images [3]. It measures the weighted similarity between histograms which provides more desirable results than "like-bin" only comparisons. The quadratic distance between histograms $\mathbf{h}_q$ and $\mathbf{h}_t$ is given by

$$d^{hist}_{q,t} = (\mathbf{h}_q - \mathbf{h}_t)^t \mathbf{A} (\mathbf{h}_q - \mathbf{h}_t), \quad (6)$$

where $\mathbf{A} = [a_{i,j}]$ and $a_{i,j}$ denotes the similarity between colors with indices i and j. By defining color similarity in HSV color space, $a_{i,j}$ is given by Eq. 3. Since the histogram quadratic distance computes the cross similarity between colors, it is computationally expensive. Therefore, in large database applications, histogram indexing strategies, such as pre-filtering [5], are required to avoid exhaustive search.

Color Sets

Alternatively, we utilize color sets to represent color information. The distinction is that color sets give only a selection of colors, whereas, color histograms denote the relative amounts of colors. Although we use the above system for color set selection in order to extract regions, we note here that color sets can also be obtained by thresholding color histograms. For example, given threshold τ_m for color m, color sets are related to color histograms by

$$c[m] = \begin{cases} 1 & \text{if } h[m] \geq \tau_m \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Color sets work well to represent regional color since (1) T_c and Q_c^M have been derived to give a complete set of distinct colors and (2) salient regions possess only a few, equally dominant colors [13].

Color Set Distance

We use a modification of the color histogram quadratic distance equation (Eq. 6) to measure the distance between color sets. The quadratic distance between two color sets $\mathbf{c}_q$ and $\mathbf{c}_t$ is given by

$$d^{set}_{q,t} = (\mathbf{c}_q - \mathbf{c}_t)^t \mathbf{A} (\mathbf{c}_q - \mathbf{c}_t), \quad (8)$$

Considering the binary nature of the color sets, the computational complexity of the quadratic distance function can be reduced. We decompose the color set quadratic formula to provide for a more efficient computation and indexing. By defining $\mu_q = \mathbf{c}_q^t \mathbf{A} \mathbf{c}_q$, $\mu_t = \mathbf{c}_t^t \mathbf{A} \mathbf{c}_t$ and $\mathbf{r}_t = \mathbf{A}\mathbf{c}_t$, the color set quadratic distance is given as

$$d^{set}_{q,t} = \mu_q + \mu_t - 2\mathbf{c}_q^t \mathbf{r}_t. \quad (9)$$

Since $\mathbf{c}_q$ is a binary vector,

$$d^{set}_{q,t} - \mu_q = \mu_t - 2 \sum_{\forall m \text{ where } \mathbf{c}_q[m]=1} r_t[m]. \quad (10)$$

That is, any query for the most similar color set to $\mathbf{c}_q$ may be easily processed by accessing individually μ_t and $r_t[m]$'s, where $m \in 0 \ldots M-1$, see Table 2. As such, μ_t and $\mathbf{r}_t[m]$'s are precomputed, stored and indexed individually. Notice also that μ_q is a constant of the query. The closest color set, $\mathbf{c}_t$, to $\mathbf{c}_q$ is the one that minimizes $\mu_t - 2\sum_{\forall m \text{ where } \mathbf{c}_q[m]=1} r_t[m]$.

IMID	REGID	μ^*	$r[0]^*$	$r[1]^*$	...	$r[M-1]^*$

Table 2: The COLORSET relation with attributes for the decomposed quadratic distance equation parameters μ_t and $r_t[m]$'s. Denotation by * indicates that a secondary index is built on the attribute in order to allow range queries to be performed on that attribute.

3.2 Color Set Query Strategy

We now define the strategy for processing the color set queries, which is an important building block in the overall image query process. The color set query compares only the color content of regions or images. Spatial queries are considered in the next section. Given query $Q = \{\mathbf{c}_q\}$, the best match to Q is target $T_j = \{\mathbf{c}_{t_j}\}$, where,

$$\begin{aligned} j &= \text{argmin}_j (d^{set}_{q,t_j}) \\ &= \text{argmin}_j (\mu_{t_j} - 2\textstyle\sum_{\forall m \text{ where } \mathbf{c}_q[m]=1} r_{t_j}[m]). \end{aligned}$$

A straightforward way to process the query is to compute d^{set}_{q,t_j} exhaustively for all j using Eq. 10. In this case, the computation of d^{set}_{q,t_j} requires $M'+1$ additions per target and no multiplications, where M' is the number of non-zero colors in $\mathbf{c}_q$. This provides for drastic improvement over the original histogram quadratic form.

The complexity of this approach, where M = size of color set (histogram) and N = number of images in the database, is $O((M'+1)N)$, where $M' << M$, gives the number of non-zero colors in the query color set. Compare this to a naive computation of quadratic histogram distance, which is $O(M^2 N)$. In QBIC, the computation is reduced by the technique in [5] to $O(M^2 C + M'' N)$, where typically, $M'' = 3$, and $C << N$. We note that the technique in [5] can be combined with the color set technique to further reduce complexity.

Given the query color set: *find the best match to color set* $\mathbf{c}_q$, where $d^{set}_{q,t} \leq \tau^c$ defines the maximum tolerance for color set distance, the following range queries produce the best match:

MU	$\leftarrow$	Select μ_t from COLORSET where $\mu_t \leq \tau^c - \mu_q$
COL$_m$	$\leftarrow$	Select $r_t[m]$ from COLORSET where $r_t[m] \geq \frac{\mu_q + \mu_t - \tau^c}{2}$, $\forall m$ where $c_q[m] = 1$
CAND	$\leftarrow$	MU $\cap_{\mathrm{IMID}}$ COL$_i$ $\cap_{\mathrm{IMID}}$ COL$_j$ $\cap_{\mathrm{IMID}} \cdots$ $\cap_{\mathrm{IMID}}$COL$_k$
MATCH	$\leftarrow$	$\min_{d^{set}_{q,t}}$(CAND), $(Eq.\ 10)$.

As this query strategy illustrates, color region matching is accomplished by performing several range queries on the query color set's colors, taking the intersection of these lists and minimizing the sum of attributes in the intersection list. The best match minimizes the color set distance. Since this strategy merely provides efficient indexing of the terms in the decomposed quadratic color set distance equation, the query strategy provides for no false dismissals. The final candidate image list has false alarms which are removed through the final minimization which requires only additions in the computation of $d^{set}_{q,t}$. Using a similar indexing strategy for retrieving images using color histograms in a database of 750,000 images, the computation of the 60 best matches takes less than two seconds [16].

4 SINGLE REGION QUERY

The color set query strategy provides for comparison between the colors of images or regions. Now we consider in addition the spatial distances between the regions.

4.1 Region Absolute Location

In specifying spatial properties of the individual regions in the query, we provide for indexing of region centroids and minimum bounding rectangles.

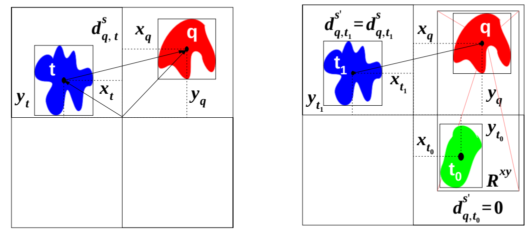
Fixed Query Location

The spatial distance between regions is given by the euclidean distance of centroids as illustrated in Figure 7(a).

$$d^s_{q,t} = [(x_q - x_t)^2 + (y_q - y_t)^2]^{\frac{1}{2}}. \tag{11}$$

Bounded Query Location

We also give the user flexibility in designating the spatial bounds for each region in the query within which

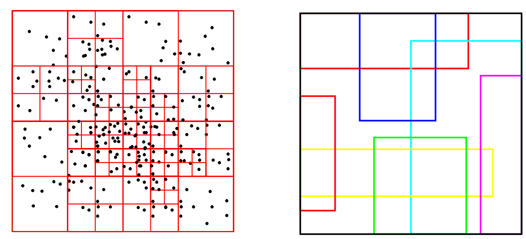


(a) (b)

Figure 7: Region spatial distance (a) fixed spatial distance $d^s_{q,t}$, (b) bounded spatial distance $d^{s'}_{q,t}$, where R^{xy} determines the valid spatial bounds.

a target region is assigned a spatial distance of zero. When a target region falls outside of the spatial bounds the spatial distance is given by the euclidean distance as illustrated in Figure 7(b). This is useful in many situations when the user does not care about the exact position of matched regions as long as they fall within a designated area.

$$d^{s'}_{q,t} = \begin{cases} 0 & \text{if } (x_t, y_t) \in R^{xy} \\ d^s_{q,t} & \text{otherwise.} \end{cases} \tag{12}$$



(a) (b)

Figure 8: (a) spatial quad-tree indexes region centroids, (b) r-tree indexes region MBRs (rectangles).

Centroid Location Spatial Access – Spatial Quad-trees

The centroids of the image regions are indexed using a spatial quad-tree on their x and y values as illustrated in Figure 8(a). The quad-tree provides quick access to 2-D data points [7]. A query for region at location (x_t, y_t) is processed by first traversing the spatial quad-tree to the containing node, then exhaustively searching the block for the points that minimize $d^s_{q,t}$. In the case that the user specifies a bounded spatial query, a range of blocks are evaluated such that points within the spatial bounds are all assigned $d^{s'}_{q,t} = 0$.

Rectangle Location Spatial Access – R-trees

Since the spatial locations of extracted image regions are not sufficiently represented by centroid locations alone, we also index the region spatial locations by their minimum bounding rectangles (MBRs). The MBR is the smallest vertically aligned rectangle that completely encloses the region. For example, Figure 1(c) illustrates the MBRs for some image regions. In this way, a spatial query may specify a search rectangle and the target regions are found that overlap with it spatially. The MBRs of the regions are indexed using an r-tree as illustrated in Figure 8(b). The r-tree provides a dynamic structure for indexing $k-D$ rectangles [17], here $k = 2$. The r-tree, which consists of a hierarchy of overlapping spatial nodes, is designed to visit only a small number of nodes in a spatial search.

4.2 Size

Another important perceptual dimension of the regions is their size in terms of area and spatial extent.

Area

The distance in area between two regions is given by the absolute distance

$$d^a_{q,t} = |\texttt{area}_q - \texttt{area}_t|. \tag{13}$$

Spatial Extent

The width and the height of the MBRs provide for useful comparison of region spatial extents. It is much simpler than shape information, which we do not utilize in the system, and provides excellent grounds for discriminating regions. The distance in MBR width (w) and height (h) between two regions is given by

$$d^m_{q,t} = [(w_q - w_t)^2 + (h_q - h_t)^2]^{\frac{1}{2}}. \tag{14}$$

4.3 Single Region Query Strategy

Integrating these approaches, the overall region query strategy consists of computing individual queries on color set, region location, area and spatial extent, as specified by the user. The process is summarized in Figure 9.

The single region distance is given by the weighted sum of the color set (Eq. 10), location (Eq. 11), area (Eq. 13) and spatial extent (Eq. 14) distances. The user may also assign a relative weighting α_n to each of these attributes of each region. For example, the user may weight the size parameter more heavily than color and location in the query. The overall single region query distance is given by

$$d_{q,t} = \alpha_c d^{set}_{q,t} + \alpha_s d^s_{q,t} + \alpha_a d^a_{q,t} + \alpha_m d^m_{q,t}. \tag{15}$$

We now outline the strategy for processing the joint color, absolute location and size queries for a single region. For example, given the single region query: *find the region that best matches* $Q = \{\mathbf{c}_q,\ (x_q, y_q),\ \texttt{area}_q,\ (w_q, h_q)\}$, the query is processed by first computing the individual queries for color, location, size and spatial extent. The intersection of the region match lists is then computed to obtain the set of common images; the best match minimizes the total distance. Here, B^{xy}, A^{xy} and B^{wh} are the search thresholds for location, area and spatial extent, respectively, which are set by the system or by the user.

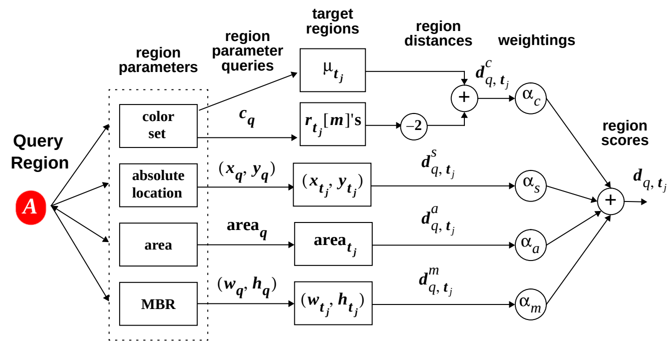


Figure 9: Overall strategy for a single region query with parameters of color set, location, area and spatial extent.

COLOR $\leftarrow$ Color Set Query($\mathbf{c}_q$)
LOC $\leftarrow$ Select (x_t, y_t) from REGION where $(x_t, y_t) \in B^{xy}$
SIZE $\leftarrow$ Select area from REGION where $\texttt{area}_t \in A^{xy}$
MBR $\leftarrow$ Select (w_t, h_t) from REGION where $(w_t, h_t) \in B^{wh}$
CAND $\leftarrow$ COLOR $\cap_{\text{IMID}}$ LOC $\cap_{\text{IMID}}$ SIZE $\cap_{\text{IMID}}$ MBR
MATCH $\leftarrow$ $\min_{d_{q,t}}$(CAND), ($Eq.$ 15).

5 MULTIPLE REGIONS QUERY

The overall image query strategy consists of joining the queries on the individual regions in the query image. The join, which consists of intersecting the results of region matches, identifies the candidate target images which contain matches to all the query regions. For these images, the image match score is computed by adding the weighted scores from the best regions matches. In the final stage, the relative spatial locations, that may have been specified in the query are evaluated to determine the best match image that satisfies the constraints of relative region placement. The image match process is illustrated in Figure 10. By specifying multiple regions in the query without any spatial relations between regions the complexity increases only linearly with the number of query regions.

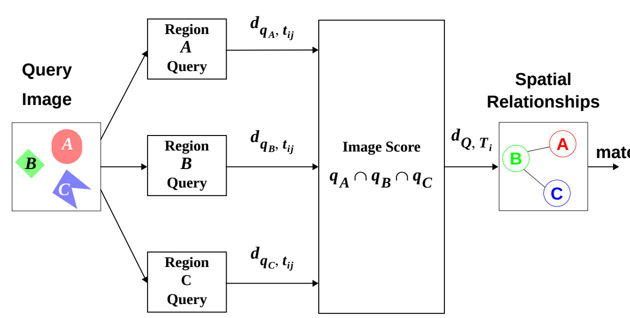


Figure 10: Overall strategy for computing image matches to a query image with several regions by combining individual regions queries.

5.1 Multiple Regions Query Strategy – Absolute Locations

For each region in the query positioned by absolute location, the query strategy outlined for single region query (Section 4.3) is carried out, without computing the final minimization. These lists are intersected and the best image match minimizes the combined region distance. For example, given the multiple region query: *find the image having three regions that best matches* $Q = \{Q^A, Q^B, Q^C\}$, where $Q^i = \{\mathbf{c}^i_q, (x^i_q, y^i_q), \mathtt{area}^i_q, (w^i_q, h^i_q)\}$ gives the query parameters for query region i, the matches are found as follows:

$$\begin{array}{lll} \mathtt{REG}_A & \longleftarrow & \mathtt{Single\ Region\ Query}(Q^A) \\ \mathtt{REG}_B & \longleftarrow & \mathtt{Single\ Region\ Query}(Q^B) \\ \mathtt{REG}_C & \longleftarrow & \mathtt{Single\ Region\ Query}(Q^C) \\ \mathtt{CAND} & \longleftarrow & \mathtt{REG}_A \cap_{\mathrm{IMID}} \mathtt{REG}_B \cap_{\mathrm{IMID}} \mathtt{REG}_C \\ \mathtt{MATCH} & \longleftarrow & \min_{\beta_0 d_{Q_A,T} + \beta_1 d_{Q_B,T} + \beta_2 d_{Q_C,T}}(\mathtt{CAND}). \end{array}$$

The query is processed by intersecting the query region lists to obtain the list of candidate images. The best match minimizes the weighted sum of the region distances between the query and target image.

5.2 Region Relative Location

Querying by absolute locations cannot be easily extended to include relative locations of regions. For example, for a target with L regions and a query image with K regions, there are $L!/((L-K)!K!)$ possible matches, each of which requires a distance computation. For example, $L = 20$ and $K = 3$ requires $1,140$ comparisons. To circumvent this exhaustive search, we utilize a convenient representation of image spatial relationships and their comparisons based upon 2-D strings [6].

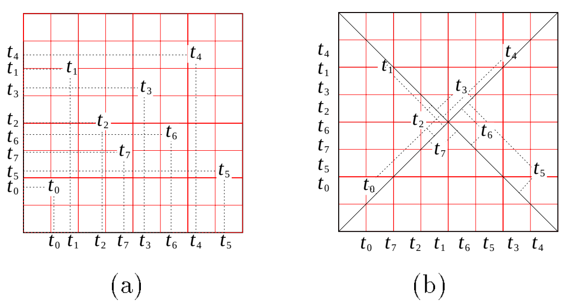


Figure 11: 2-D string representation of spatial relationships (a) un-rotated 2-D string, (b) rotated projection.

The 2-D string represents spatial relationships as illustrated in Figure 11(a). The 2-D string is a symbolic projection of the image along the x and y directions [18]. For example, the 2-D string for the image in Figure 11(a) is given as $(t_0 t_1 < t_2 < t_7 < t_3 < t_6 < t_4 < t_5, t_0 < t_5 t_7 < t_6 < t_2 < t_3 t_1 < t_4)$, where the symbol '$<$' denotes the left-right or bottom-top relation. Using 2-D strings, the match complexity is reduced to approximately $O(L^3/K)$ per target image. Since we evaluate spatial relationships only at the final stage of the query, the complexity is further reduced to $O(L'^3/K)$, where L' is the number of candidate regions in a target image, and $L' < L$ and $L' \simeq K$.

5.3 Special Spatial Relations

We are currently implementing several spatial relations that are of particular interest such as region adjacency, nearness, overlap and surround. These examples of spatial reasoning can be inferred from the 2-D string [18]. Scale invariance is also provided in the 2-D string, but rotation invariance is not.

Adjacency, Nearness, Overlap and Surround

The adjacency of regions is resolved from the 2-D string by determining whether or not other regions exist between two candidate regions. The nearness of regions is resolved using adjacency in the 2-D string, but since the 2-D string provides no metric information, it requires an additional computation of the region distance. Since the 2-D string captures the orthogonal relations of the regions in the image, conditions of overlap and surround can also be determined [18]. Overlapping regions require not only the determination of nearness but also require the evaluation of the intersection of the MBRs of the regions. The case of surround is determined by unifying sub-goal queries that evaluate regions to the north, south, east and west of the surrounded-by region.

Spatial Invariance

Under certain conditions the user may desire invariance in scaling and/or rotation. The 2-D string provides scale invariance since a change in scale does not change the order of symbols in the string. However, rotation invariance is not provided in the 2-D string. Therefore, we approximate rotation invariance around the image center point by providing two projections of each image, one normal projection, and another based upon a rotation by 45 degrees.

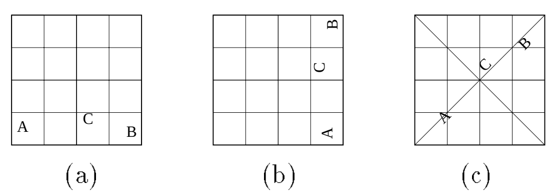


Figure 12: 2-D strings rotation invariance (a) 0° $(A < C < B, ABC)$, (b) 90° $(ABC, A < C < B)$, (c) 45° $(A < C < B, ABC)$ with projection onto the rotated axis.

In this case, the rotated 2-D string is extracted by the projection onto the diagonals of the image as illustrated in Figure 11(b). Image rotation by 90° is provided by swapping the x and y projections as illustrated in Figure 12(a) and (b). Rotation by 45° is detected in the rotated 2-D string as illustrated in Figure 12(a) and (c). Other arbitrary rotations are approximated by either the 2-D string or the rotated 2-D string.

5.4 Multiple Regions Query Strategy – Relative Locations

The resolution of the relative spatial locations is performed in the final stage. For the regions positioned in the query by absolute locations, the location queries from Section 5.1 are performed. For the regions positioned in the query by relative locations, queries on all attributes except location are performed. Then the intersection of the region lists is performed to identify and score candidate target images.

For each candidate image, the 2-D string is generated from the identified regions and is compared to the 2-D string of the query image. This final operation either validates the target image or rejects it. Therefore, after the final stage, all false alarms from the earlier stages are removed.

6 QUERY EVALUATION

We now present some example color/spatial queries and provide some initial evaluations.

Query Formulation

The joint color/spatial queries are formulated graphically by using the VisualSEEk user tools as illustrated in Figure 13. The user sketches regions, positions them on the query grid and assigns them properties of color, size and absolute location. The user may also assign boundaries for location and size.

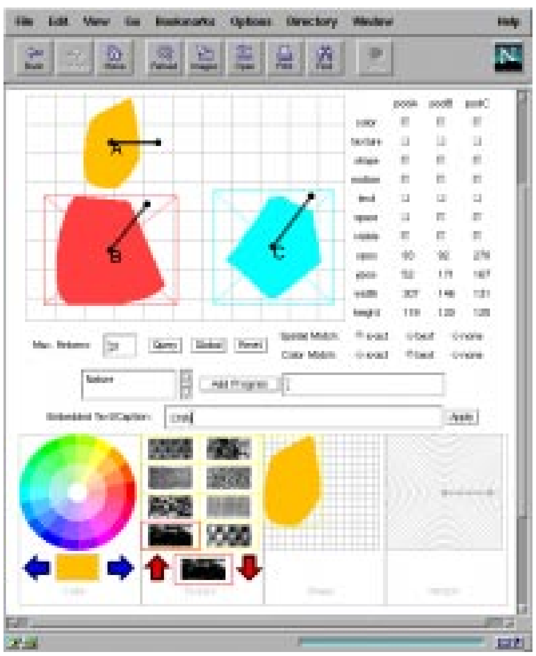
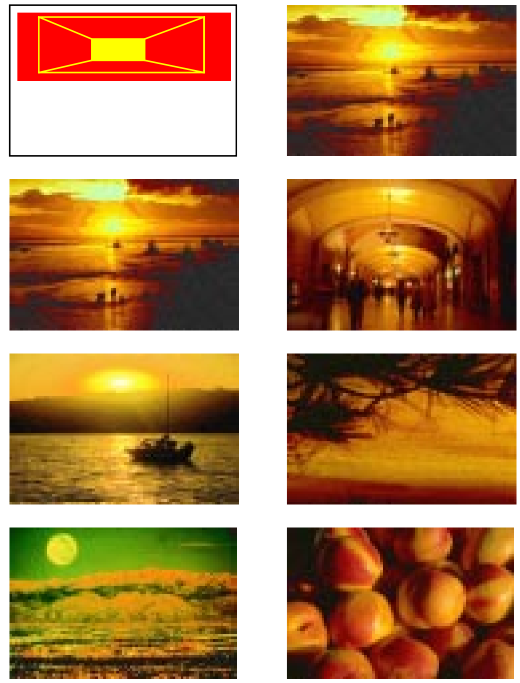


Figure 13: VisualSEEk user interface

Unconstrained Image Queries We illustrate the power and flexibility of the VisualSEEk query system over non-spatial techniques in Figure 14. In Figure 14(a) (top left), a VisualSEEk query is diagrammed that specifies two regions (outer is orange and inner is yellow) and their spatial layout with the goal of retrieving images of sunsets. The best matches to the color/spatial query (left) have a similar arrangement of similarly colored regions. In Figure 14(b), a typical sunset image is used (top right), and the best matches (right) are found that have the most similar color histograms to the query image. We see that the global color histogram query process gives the user little control in specifying the query and more readily returns images that are not desired.



(a) (b)

Figure 14: Sample "sunset" queries (a) VisualSEEk query using diagrammed query at top left, (b) color histogram query using query image at top right. Best matches are listed from top to bottom.

In Figure 15(a) (top left), a VisualSEEk query is diagrammed that specifies three regions (from top to bottom: light blue, tan and green) and their spatial layout. The best matches to the query have a similar arrangement of the three colored regions. In Figure 15(b), the first VisualSEEk image match is used (top right = second left). The best matches are found that have the most similar color histograms to the query image. We see that the color histogram matches have little similarity to the regions in the query image. We note that for the color/spatial queries in Figure 14(a) and Figure 15(a), the query response time is approximately 1-2 seconds.

Synthetic Image Queries To test the color/spatial query system, we generated 500 synthetic images by selecting, manipulating and compositing single color regions into the synthetic images. Some examples are illustrated in Figure 16. While the synthetic images appear quite different from real images, the composition of regions is still difficult to decipher. For example, in the images in Figure 16 (bottom), the original composited regions cannot be extracted exactly because of occlusion and aggregation of the regions.

The region library consists of twelve elementary shapes, illustrated at the top of Figure 17. To construct each synthetic image, shapes were selected at random from the library, and were randomly rotated, scaled, colored and composited, as illustrated in Figure 17. The control factors, such as color, size, and location of regions, were

(a) (b)

Figure 15: Sample "nature" queries (a) VisualSEEk query using diagrammed query at top left, (b) Color histogram query using query image at top right. Best matches are listed from top to bottom.

used to establish a ground truth database.

The synthetic image query experiment was conducted as follows: 100 synthetic query images were generated, which have from one to three randomly selected regions in each, i.e., see Figure 16 (top); 500 target images were generated as described above, i.e., see Figure 16 (bottom). For each query image, all of the target images were assigned a similarity score using the ground truth database and using an exhaustive comparison and distance minimization over all query and target regions (Query GT). The target images from each Query GT were sorted by closest distance to the query image. The target images were assigned a relevance to each query based upon rank in the sorted list as follows: if rank = 1 to 5, relevance = 1; if rank = 6 to 10, relevance = 0.5; if rank = 11 to 500, relevance = 0.

Using the target image relevances obtained from Query GT, the queries were next evaluated using several methods and compared to the Query GT. In Query Q1, the region indexing and distance computation strategy outlined in the paper was carried out on the ground truth database. In Query Q2, the same query strategy was carried out on a region database that was generated automatically from the target images using color set back-projection. Finally, In Query Q3, the combined color histogram of the query regions was matched to the com-

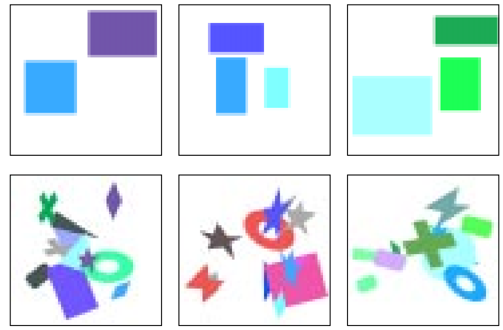


Figure 16: Examples of the synthetic images, query images (top), target images (bottom), used for evaluating the joint color/spatial query system.

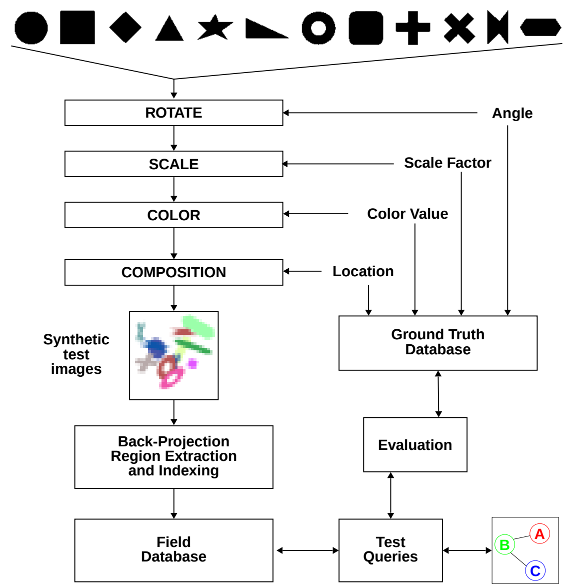


Figure 17: Generation of synthetic images and ground truth database and process for evaluating effectiveness of joint color/spatial queries.

bined region color histogram of each target image as the basis for image retrieval.

We see in Figure 18 that the color/spatial query strategy (Query Q2) performs much better than color histograms (Query Q3) in retrieving image matches. We also see that Query Q1 performs better than Query Q2. The difference represents the loss of information in the process of automated region extraction through color set back-projection. Furthermore, the drop in retrieval effectiveness in Queries Q1 and Q2 from the ground truth Query GT results from the color/spatial indexing strategies.

In computing the similarity between the query and target images in Query GT, all regions in the target are considered. In this way, some target images are identified as matches even when only two out of three regions are close matches. These configurations are considered only because Query GT conducts an exhaustive search on all target regions. In the indexing strategy outlined

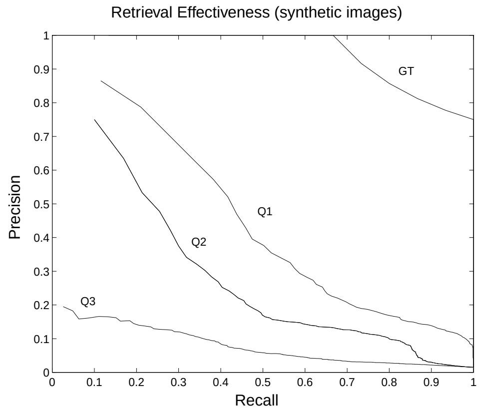


Figure 18: Average retrieval effectiveness of 100 randomly generated queries on database of synthetic images, with the four query methods defined as above.

in the paper, candidate target images are required to possess regions which are all sufficiently close to the query regions. This restriction, which allows the gain in retrieval efficiency over exhaustive search, explains the retrieval effectiveness drop compared to Query GT.

Evaluation of Color Sets

In order to evaluate the impact of the loss of information in using color sets instead of color histograms, we compared their performance in retrieving images by global color content. This experiment does not evaluate the color/spatial query system, rather, it compares color sets directly to color histograms.

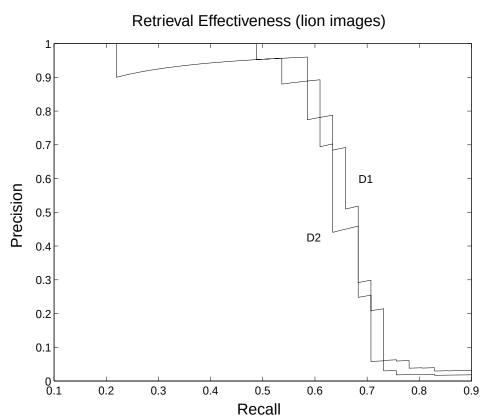


Figure 19: Retrieval of 83 lion images from a database of 3,100 images: $D1$ = color histogram quadratic distance, $D2$ = color set quadratic distance.

In an image database of 3,100 images, we measured the ability of color sets and color histograms to retrieve the 83 images of lions using an example lion image. Figure 19 depicts the retrieval effectivenesses in the retrieval of images of lions. The experiment shows that retrieval effectiveness degrades only slightly using color sets and the quadratic distance measure (Eq. 10) compared to color histograms using the quadratic distance measure (Eq. 6). This indicates that the perceptually significant color information is retained in the color sets.

Example VisualSEEk Queries

We now illustrate the range of color/spatial queries that are possible in VisualSEEk. In the first example, see Figure 20(a), the query (top) specifies the absolute location of a single region. The retrieved image (bottom) has the best match in color and size to the query region and falls within the "zero-distance" bound diagrammed in the query. In Figure 20(b), the query specifies two regions. The retrieved image has two color match regions located at the positions in the query image. In Figure 20(c), the query specifies the spatial relationships of three regions. The retrieved image has three regions that best match the colors of the query regions and their spatial relationship satisfies that specified in the query. In Figure 20(d), the query specifies both absolute and relative locations of regions. In this query, the match to the region positioned by absolute location (top left region in query image) considers both the color and location of this region. The match to the other regions (bottom two regions in query image) at first considers only the colors of these regions. In the last stage of the query, the spatial relationships of the regions are evaluated to determine the match.

7 SUMMARY AND FUTURE WORK

We presented a new image database system which provides for color/spatial querying. Since, the discrimination of images is only partly provided by global features such as color histograms, the VisualSEEk system instead utilizes salient image regions and their colors, sizes, spatial locations, and relationships, in order to compare images. The integration of content-based and spatial querying provides for a highly functional query system which allows for wide variety of color/spatial queries. We presented the strategies utilized by the VisualSEEk system for computing these complex queries and presented some preliminary results that indicate the system's efficiency and power. We will next extend the VisualSEEk system to extract and index regions of texture, and color and texture jointly [14]. We will also investigate and include methods for shape comparison in order to further enhance the image region query system.

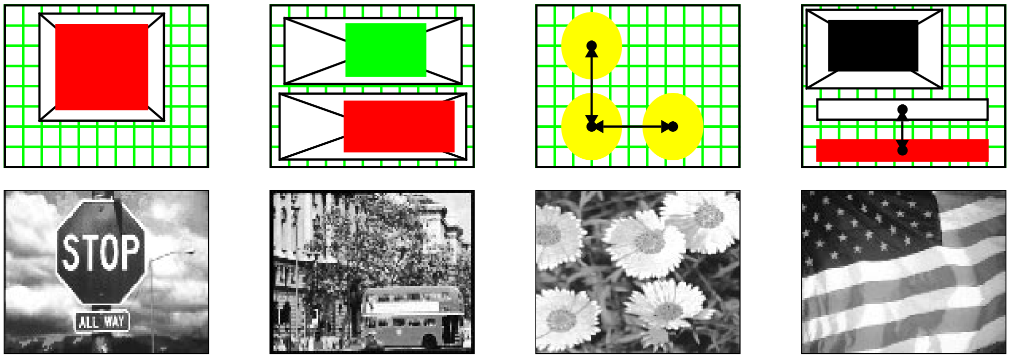
8 ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under a CAREER award (IRI-9501266), IBM under a 1995 Research Partnership (Faculty Development) Award, and sponsors of the ADVENT project of Columbia University. The authors wish to thank Prof. Al Aho for his initial involvement in this project.

REFERENCES

1. M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:1 1991.

2. M. Stricker and A. Dimai. Color indexing with weak spatial constraints. In *Symposium on Electronic Imaging: Science and Technology – Storage & Retrieval for Image and Video Databases IV*, pages 29 – 41. IS&T/SPIE, 1996.



(a) (b) (c) (d)

Figure 20: Example VisualSEEk queries (a) region with absolute location, (b) two regions with absolute locations, (c) multiple regions with relative locations, (d) multiple regions with both absolute and relative locations.

3. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, and C. Faloutsos. The QBIC project: Querying images by content using color, texture, and shape. In *Storage and Retrieval for Image and Video Databases*, volume SPIE Vol. 1908, February 1993.

4. E. G. M. Petrakis and C. Faloutsos. Similarity searching in large image databases. Technical Report 3388, Department of Computer Science, University of Maryland, 1995.

5. J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Trans. Pattern Anal. Machine Intell.*, July 1995.

6. S.-K. Chang, Q. Y. Shi, and C. Y. Yan. Iconic indexing by 2-D strings. *IEEE Trans. Pattern Anal. Machine Intell.*, 9(3):413 – 428, May 1987.

7. H. Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys*, 16(2):187 – 260, 1984.

8. V. N. Gudivada and V. V. Raghavan. Design and evaluation of algorithms for image retrieval by spatial similarity. *ACM Trans. on Information Systems*, 13(2), April 1995.

9. A. Soffer and H. Samet. Retrieveal by content in symbolic-image databases. In *Symposium on Electronic Imaging: Science and Technology – Storage & Retrieval for Image and Video Databases IV*, pages 144 – 155. IS&T/SPIE, 1996.

10. J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. C. Jain, and C. Shu. Virage image search engine: an open framework for image management. In *Symposium on Electronic Imaging: Science and Technology – Storage & Retrieval for Image and Video Databases IV*, pages 76 – 87. IS&T/SPIE, 1996.

11. G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *Proc. ACM Intern. Conf. Multimedia*, Boston, MA, 1996.

12. C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series*, pages 277 – 286, Los Angeles, CA, 1995.

13. J. R. Smith and S.-F. Chang. Tools and techniques for color image retrieval. In *Symposium on Electronic Imaging: Science and Technology – Storage & Retrieval for Image and Video Databases IV*, volume 2670, San Jose, CA, February 1996. IS&T/SPIE.

14. J. R. Smith and S.-F. Chang. Local color and texture extraction and spatial query. In *Proc. Int. Conf. Image Processing*, Lausanne, Switzerland, September 1996. IEEE.

15. S.-F. Chang. Compressed-domain techniques for image/video indexing and manipulation. In *Proceedings, I.E.E.E. International Conference on Image Processing*, Washington, DC, Oct. 1995. invited paper to the special session on Digital Library and Video on Demand.

16. J. R. Smith and S.-F. Chang. Searching for images and videos on the World-Wide Web. Technical Report CU/CTR 459-96-25, Columbia University, August 1996.

17. A. Guttman. R-trees: A dynamic index structure for spatial searching. In *ACM Proc. Int. Conf. Manag. Data (SIGMOD)*, pages 47 – 57, June 1984.

18. S.-K. Chang. *Principles of Pictorial Information Systems Design*. Prentice Hall, 1989.