Content-Based Image Retrieval Based on a Fuzzy Approach

Raghu Krishnapuram, *Senior Member*, *IEEE*, Swarup Medasani, *Member*, *IEEE*, Sung-Hwan Jung, *Member*, *IEEE*, Young-Sik Choi, *Member*, *IEEE*, and Rajesh Balasubramaniam

Abstract—A typical content-based image retrieval (CBIR) system would need to handle the vagueness in the user queries as well as the inherent uncertainty in image representation, similarity measure, and relevance feedback. In this paper, we discuss how fuzzy set theory can be effectively used for this purpose and describe an image retrieval system called FIRST (Fuzzy Image Retrieval SysTem) which incorporates many of these ideas. FIRST can handle exemplar-based, graphical-sketch-based, as well as linguistic queries involving region labels, attributes, and spatial relations. FIRST uses Fuzzy Attributed Relational Graphs (FARGs) to represent images, where each node in the graph represents an image region and each edge represents a relation between two regions. The given query is converted to a FARG, and a low-complexity fuzzy graph matching algorithm is used to compare the query graph with the FARGs in the database. The use of an indexing scheme based on a leader clustering algorithm avoids an exhaustive search of the FARG database. We quantify the retrieval performance of the system in terms of several standard measures.

Index Terms—Content-based image retrieval, fuzzy graph models, graph matching, graph clustering, indexing.

1 Introduction

The rapid growth in the number of large-scale image repositories in many domains such as medical image management, multimedia libraries, document archives, art collections, geographical information systems, law enforcement agencies, and journalism has brought about the need for efficient content-based image retrieval (CBIR) mechanisms. There are several popular CBIR systems such as QBIC, Virage, RetrievalWare, Photobook, Chabot, VisualSeek, WebSeek, MARS system, SurfImage, Netra, and CANDID. We do not mention the specific merits of each of the systems as they are covered in detail in [1]. The problems involved in image retrieval are widely known and special issues of many leading journals have addressed this topic [2], [3], [4], [5]. In CBIR systems, the queries that are used to retrieve images can be broadly classified as primitive, logical, and abstract. A query is said to be a primitive query if it is based on features (such as color, shape, and texture) extracted from the images. A query is said to be logical if it employs the identities of the objects in the image. Sketch-based and linguistic queries in which the user depicts or describes objects/regions in the desired spatial positions and ascribes attributes (such as class label, size, color, and shape properties) to them can also be considered logical queries. Abstract queries are typically based on a notion of similarity which is a concept that cannot be easily captured in a mathematical model [6]. Logical and abstract queries are sometimes known as semantic queries. There have been some attempts at using linguistic queries or semantic attributes [7], [8], [9], [10], [11], [12], [13]. However, this area has not received the attention it deserves.

In this paper, we describe how concepts from fuzzy set theory can be useful in building a more versatile CBIR system that can handle the usual exemplar-based queries as well as graphical sketch-based queries and linguistic queries [14]. We show that fuzzy sets can be used to model the vagueness that is usually present in the image content, user query, similarity measure, and relevance feedback. It is well-known that object labels are not crisp and attribute values such as *small* and *somewhat*, as well as spatial relations such as *left of* and *below*, are handled much better by fuzzy techniques [15], [16]. Therefore, a fuzzy approach allows us to retrieve relevant images that might be missed by traditional approaches. We present FIRST, a Fuzzy Image Retrieval SysTem that incorporates many of these ideas. FIRST uses a fuzzy attributed relational graph (FARG) (originally called fuzzy attributed graph or FAG [17]) to represent each image in the database in which each object/region in the image is represented by a node with attributes (e.g., *blueness*, *size*, and *texturedness*), and the relations between regions are represented by edges with attributes (e.g., *spatial relation*, *adjacency*). We also recast the user query as a FARG, which converts the image matching problem to a subgraph matching problem. It is well-known that subgraph matching is nontrivial and, in fact, NP-complete [18]. To overcome this problem, we use a new low-complexity algorithm to match FARGs. We further reduce the retrieval time by organizing (indexing) the

- *R. Krishnapuram is with IBM India Research Lab, Block I, Indian Institute of Technology, Hauz Khas, New Delhi 110016, India. E-mail: kraghura@in.ibm.com.*
- *S. Medasani is with HRL Laboratories, LLC, 3011 Malibu Canyon Road, Malibu, CA 90265. E-mail: smedasani@hrl.com.*
- *S.-H. Jung is with the Department of Computer Engineering, Changwon National University, 9 Sarim-dong, Changwon-shi, Gyeongnam 641-774, South Korea. E-mail: sjung@sarim.changwon.ac.kr.*
- *Y.-S. Choi is with the Department of Computer Engineering, HanKuk Aviation University, 200-1 Hwajon-Dong, Dukyang-Gu, Koyang-Shi, Kyonggi-Do, South Korea 412-791. E-mail: choimail@hau.ac.kr.*
- *R. Balasubramaniam is with Sun Microsystems Inc., BRM-06, Broomfield, CO 80021. E-mail: RajeshBalasubramaniam@sun.com.*

Manuscript received 11 Feb. 2003; revised 18 July 2003; accepted 22 July 2003.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 118287.

database in terms of groups (clusters) of similar FARGs by using a leader clustering algorithm. The incremental nature of this algorithm lets us add new images to the database easily.

The rest of the paper is organized as follows: In Section 2, we describe how fuzzy set theory can be used in image representation, spatial relation models, similarity measures, indexing methods, and relevance feedback. In Section 3, we describe Fuzzy Attributed Relational Graphs (FARGs) in more detail and describe a new extension of a fuzzy graph matching algorithm for matching FARGs. In Section 4, we present our Fuzzy Information Retrieval SysTem (FIRST). In Section 5, we present the details of a new indexing scheme based on clustering of FARGs. In Section 6, we present retrieval results on a synthetic image database as well as an outdoor scene image [12] database. Finally, in Section 7, we present the conclusions.

2 Background

2.1 A Fuzzy Approach to Feature-Based Image Representation

CBIR systems commonly use a set of features for image representation in addition to some metainformation that is stored as keywords. Most systems use color features in the form of color histograms to compare images [19], [20], [21], [22]. The ability to retrieve images when color features are similar across the database is achieved by using texture features [23], [24], [25], [26], [27]. Shape is also an important attribute that is employed in comparing similarity of regions in images [4], [28], [29], [30], [31], [32]. Since the user's perception of features such as color, texture, and shape is imprecise, a fuzzy approach is much better suited for expressing queries involving concepts such as *a somewhat round tree that is dark green and has fine texture*. With a simple n-dimensional feature vector representation where each element of the vector corresponds to the value of a feature or attribute of the image, it is not easy to handle such queries. An alternative is to use a representation in which each element of the vector stores a fuzzy value of the attribute. We now briefly describe how this can be accomplished.

Let a_i, $i = 1, \ldots, n_A$, denote the ith attribute (e.g., "blueness"). We define a set of linguistic labels (with corresponding membership functions) for attribute a_i. Let the set of linguistic labels be $\{L_{i1}, L_{i2}, \ldots, L_{in_{a_i}}\}$, where n_{a_i} is the number linguistic labels for attribute a_i. We follow the common practice in fuzzy set theory and use L_{ik} also to denote the membership function associated with the linguistic label. For a given attribute vector $x_j = (x_{j1}, x_{j2}, \ldots, x_{jn_A})^T$ representing image j, we compute

$$\{m_{ji1} = L_{i1}(x_{ji}), m_{ji2} = L_{i2}(x_{ji}), \ldots, m_{jin_{a_i}} = L_{in_{a_i}}(x_{ji})\},$$

i.e., the membership degrees in all labels using the membership functions. The value of attribute i is now $(m_{ji1}, m_{ji2}, \ldots, m_{jin_{a_i}})$, which is a fuzzy set defined over the domain of linguistic labels. This process can be repeated for all attributes. Using this approach, it is fairly simple to represent a linguistic query in the same format. As an example, we can consider a database consisting of images of cars. One of the attributes might be *size*, with linguistic labels *small, medium*, and *large*, and another attribute might be *blueness*, with linguistic labels *light blue, medium blue* and *deep blue*. If the query is: "Find all images with small deep blue cars," then the size attribute of the query can be represented by (1, 0, 0) and the *blueness* attribute can be represented by (0, 0, 1). In other words, the membership in the label specified in the query is 1, and memberships in other labels of the attribute are zero. In Section 2.3, we discuss how similarities between images can be computed in this representation.

Feature-based representation is limited in the sense that it does not capture the spatial relations between regions. To remedy the situation, we use a Fuzzy Attributed Relational Graph [17], [33] to model image content. We will present this approach in more detail in Section 3.

2.2 Spatial Relations in Images

Spatial relations between objects in an image can contribute significantly to the description of its content. For example, an image might have a house to the *left of* a road and *below* a tree. Freeman [15] defined 11 primitive spatial relations between two objects (*left of, right of, above, below, behind, in front of, near, far, inside, outside*, and *surround*) and recognized that they are best described in an approximate (fuzzy) framework. Very few systems exist that can handle queries that include spatial relations between objects. VisualSEEk [10] can handle spatial data in terms of the centroids and minimum bounding rectangles of objects in the image. Del Bimbo and Vicario [11] propose the idea of "weighted walkthroughs" to represent the spatial relationships between objects. However, they do not address the issue of handling linguistic descriptions of spatial relations. NETRA [12] uses two bounding rectangles to define the spatial area of interest. These descriptions do not capture the full expressive power of spatial relations.

There have been several fuzzy approaches to computing the degrees of spatial relations between image regions. The earlier methods use angle measurements between pairs (a, b) of points where point a is in Region A and point b is in region B [16]. Other methods use projections of regions on the coordinate axes and try to reason about spatial relations either using dominance relations [34] or fuzzy logic [35]. More recent methods have included approaches based on neural networks [36], mathematical morphology [37], and gravitational force models [38]. We use the morphological approach, which provides a good compromise between performance and computational complexity.

2.3 Similarity Measures for Ranking Images

The traditional approach to ranking images based on the similarity with respect to the query image is to compare the feature vector x_q of the query image with the feature vectors x_j of the images in the database based on a suitable distance (dissimilarity) measure, e.g., the Minkowski norm. This is similar to what is used in the information retrieval community [39], [40], [41]. More generally, we can also consider combining the similarities with respect to individual features in a more complex way. Let $f_i(x_{qi}, x_{ji})$ denote the similarity between the query image and image j in the

database with respect to attribute a_i. Then, the overall similarity between two images can be computed as

$$c_{qj} = g\big(f_1(x_{q1}, x_{j1}), f_2(x_{q2}, x_{j2}), \ldots, f_{n_A}(x_{qn_A}, x_{jn_A})\big). \quad (1)$$

The above model allows us to choose a different function $f_i()$ for each attribute, but no interactions between features. We can assume that the range of $f_i()$ is [0, 1]. The function $g()$ needs to be chosen depending on how we would like to combine the similarities with respect to individual features. In the case of the fuzzy representation described in Section 2.1, the similarity between the query and a database image with respect to feature i can be computed as

$$c_{qji} = h\Big(f(m_{qi1}, m_{ji1}), f(m_{qi2}, m_{ji2}), \ldots, f(m_{qin_{a_i}}, m_{jin_{a_i}})\Big).$$

A different function f can be used for different attributes. Then, the overall similarity between the query and the database image is computed as

$$c_{qj} = g\big(c_{qj1}, c_{qj2}, \ldots, c_{qjn_A}\big)$$

The choice of the connectives $f()$, $h()$, and $g()$ depends on the nature of the query. In general, $f()$ might be the min operator, and $h()$ might be the max operator. If the attributes in the query are connected by *and*, then $g()$ might be a t-norm (intersection operator). If the attributes are connected by an *or*, then $g()$ could be a t-conorm (union operator) and so on. A variety of aggregation operators are available in the fuzzy set literature [42], [43], [44], [45] for this purpose.

The above model, as well as the weighted average method (which is a commonly used aggregation method in CBIR systems [46], [47]), ignore correlations/interactions between features and are not flexible enough to model complex queries effectively. The Choquet Integral (CI) [48], [49] has been proposed [6] as a similarity measure in order to overcome this problem. It allows the user to weight different combinations of features differently. For example, if feature i is highly correlated with feature j and images are highly similar with respect to feature i, then similarity with respect to feature j should not be given much attention. We now briefly discuss the Choquet integral.

Let $Y = \{y_1, y_2, \ldots, y_n\}$ denote a finite universal set and let $\mathcal{P}(Y)$ denote the power set of Y. A function $g : \mathcal{P}(Y) \rightarrow [0, 1]$ that satisfies the following properties is called a fuzzy measure: 1) $g(\emptyset) = 0$, $g(Y) = 1$ and 2) if $A, B \subset Y$ and $A \subset B$, then $g(A) \leq g(B)$. Note that a fuzzy measure $g()$ requires 2^n coefficients, i.e., the $g()$ values of the 2^n subsets of Y. The value of $g(A)$ denotes the worth or importance of subset A. If a fuzzy measure g is additive, then $g(\{y_1, y_2\}) = g(\{y_1\}) + g(\{y_2\})$, i.e., the measure of the whole is the sum of the measures of the parts. If we allow interactions between y_is, then the measure of the whole may be less than or greater than the sum of the measures of the parts.

Let $f : Y \rightarrow [0, 1]$ be a function and let g be a fuzzy measure on Y. The Choquet Integral (CI) of f with respect to g is defined by

$$\mathcal{C}_g(f) = \sum_{i=1}^{n} f(y_{(i)})[g(B_{(i)}) - g(B_{(i-1)})],$$

where $y_{(i)}$ denotes the ith element when the elements are arranged so that $1 \geq f(y_{(1)}) \geq f(y_{(2)}) \geq f(y_{(n)}) \geq 0$, $B_{(i)} = \{y_{(1)}, \ldots, y_{(i)}\}$, and $B_{(0)} = 0$. In this formulation, $f(y_{(i)})$ is multiplied by the increase in worth or importance, i.e., by $g(B_{(i)}) - g(B_{(i-1)})$, when $y_{(i)}$ is added to the pool. If $y_{(i)}$ brings little new information, then the measure should be chosen such that $g(B_{(i)}) - g(B_{(i-1)})] \approx 0$. It can be easily verified that the Choquet integral reduces to a weighted average when the fuzzy measure g is additive. Also, if $g(B)$ depends only on the cardinality of B, i.e., if $g(A) = g(B)$ whenever $|A| = |B|$, then it is reduced to the ordered weighted average (OWA) operator [50]. It is well-known that OWA can yield any order statistic (such as the median) and can simulate concepts such as "at least k out of n" or "any k out of n." It can be easily shown that $\min_i f(y_{(i)}) \leq \mathcal{C}_g(f) \leq \max_i f(y_{(i)})$.

In the context of computing the similarity between two images, $f(y_i)$ denotes the similarity of the images with respect to feature (i), and $g(B_{(i)})$ denotes the relevance or importance of the feature subset $B_{(i)}$. In order to utilize the full potential of the Choquet integral, we need to estimate the underlying fuzzy measure to be used for a given query. This can be based on the relevance feedback provided by the user (see Section 2.5). Another possibility is to use the fuzzy integral [51] with the Sugeno measure [48]. However, the Sugeno measure is somewhat restrictive in that it assumes that all pairs of features interact the same way.

2.4 Image Indexing

A recent survey [5] concludes that the problem of indexing images in a database for efficient retrieval has not received the attention it deserves. While it is feasible to retrieve a desired image from a small collection by exhaustive search, more effective techniques are needed with larger databases. The main idea in indexing is to extract features from an image, map the features into points in multidimensional space, and then employ access structures to retrieve matches efficiently. The key issue here is to use access structures that are proven to be efficient in high dimensional spaces. A comprehensive survey of the various spatial access methods can be found in [52]. Traditional indexing methods such as B-trees used with textual databases are not well suited to handle pictorial information. Popular multidimensional indexing techniques [53], [54] include k-d tree, quad-tree, R-tree, and its variants packed R-tree, V-P tree, TV-Tree, R+-tree, the R*-tree, and the SS +-tree. The R-tree family of access structures is a generalization of the B-tree for multidimensional data.

Another approach is to flatten the multidimensional space into one-dimensional space by using space-filling curves [55] and use one-dimensional access structures to retrieve data efficiently. One of the earliest treatments of hierarchical algorithms for fast search is by Fukunaga and Narendra [56]. In addition to these approaches, clustering and neural nets have also been used [57], [58]. These methods, however, come with a lot of overhead complexity and do not fare well when the dimensionality is high. In Section 5, we present a new scalable indexing scheme based on fuzzy matching.

2.5 Relevance Feedback

Relevance feedback [39] is used in CBIR systems for two reasons: 1) There can be a big gap between high level concepts perceived by the user and low level features that are used in the system, and 2) human perception of similarity is subjective. Most research in relevance feedback uses one or both of the following approaches: 1) query-point moving and 2) weight updating. The query-point moving approach tries to improve the estimate (in terms of low-level features) of the ideal query point by moving the current query point (i.e., estimate) by a certain amount based on user feedback. Some researchers generate pseudo document vectors from image feature vectors [59]. Other researchers estimate the distribution of the relevant samples based on a parametric or nonparametric estimator [60]. The weight updating approach is a refinement method based on modifying the weights or parameters used in the computation of similarity based on the user's feedback [46], [61], [62]. Choi et al. [63] have described a method to learn the similarity measure based on the Choquet integral and show that it generally outperforms the weighted average method. Our current implementation does not incorporate relevance feedback.

3 Fuzzy Attributed Relational Graphs and Graph Matching

3.1 Fuzzy Attributed Relational Graphs

A graph $G = (V_G, E_G)$ is an ordered pair of a set of nodes (vertices) V_G and a set of edges E_G. An edge in G connecting nodes u and v is denoted by (u, v), where $(u, v) \in E_G$. A Fuzzy Attributed Relational Graph (FARG) [17] is an extension of the attributed relational graph [64] and can be used to model the vagueness associated with the attributes of nodes and edges. In our application, each node in the FARG represents a region in the image, and edges between the corresponding nodes represent the relationships between the regions. All nodes have attributes from the set $A = \{a_i | i = 1, \ldots, n_A\}$. We denote the set of linguistic values (labels) associated with attribute a_i by $\Lambda_i = \{L_{ik} | k = 1, \ldots, n_{a_i}\}$. The value of an attribute a_i at node j is a fuzzy set $\mathcal{A}_{ji}$ defined over Λ_i. For example, the node-attribute $a_1 = class\_label$ may be a fuzzy set defined over the linguistic category set $\Lambda_1 = \{sky, water, vegetation\}$, and $class\_label$ of node j may have memberships 0.9, 0.2, and 0.1 in the three categories, respectively, i.e., $\mathcal{A}_{ji} = (0.9, 0.2, 0.1)$. Similarly, the node-attribute $a_2 = size$ may be a fuzzy set defined over the set of linguistic values $\Lambda_2 = \{small, medium, large\}$. We denote the node label of node j by $\lambda(j) = \{(a_i, \mathcal{A}_{ji}) | \mathcal{A}_{ji} \in \mathcal{F}(\Lambda_i);\ i = 11, \ldots, n_A\}$, where $\mathcal{F}(\Lambda_i)$ denotes the fuzzy power set of Λ_i. Each node-attribute a_i is allowed to occur only once in $\lambda(j)$.

Edge-attributes are treated similarly. Each edge in the FARG has attributes from the set $R = \{r_i | i = 1, \ldots, n_R\}$. We denote the set of linguistic values associated with edge-attribute r_i by $\Sigma_i = \{S_{ik} | k = 1, \ldots, n_{r_i}\}$. The value of an edge-attribute r_i for an edge $e = (j, l)$ is a fuzzy set $\mathcal{R}_{ei}$ defined over Σ_i. For example, the edge-attribute $r_1 = spatial\_relation$ may be a fuzzy set defined over the set of linguistic values

$$\Sigma_1 = \{left\_of, right\_of, above, below, surrounded\_by\}.$$

The edge-attribute $r_2 = adjacency$ may be a fuzzy set defined over the set of linguistic values $\Sigma_2 = \{low, moderate, high\}$. We denote the edge label of edge $e = (j, l)$ by $\rho(e) = \{(r_i, \mathcal{R}_{ei}) | \mathcal{R}_{ei} \in \mathcal{F}(\Sigma_i); i = 1, \ldots, n_R\}$, where $\mathcal{F}(\Sigma_i)$ denotes the fuzzy power set of Σ_i. Each attribute r_i is allowed to occur only once in $\rho(e)$. The use of fuzzy sets for node and edge-attribute values enables a FARG to handle imprecise information and linguistic queries.

3.2 Graph Matching

3.2.1 The Fuzzy Graph Matching (FGM) Algorithm

In this section, we outline a fuzzy graph matching algorithm called FGM [65], [66] that uses ideas from relaxation labeling and fuzzy set theory to solve the subgraph isomorphism problem [18], [67]. Given two labeled graphs, the algorithm tries to match them and returns the best possible mapping between the two graphs. The algorithm can handle exact as well as inexact subgraph matching of weighted graphs. In the next section, we extend it further so that it can be used with FARGs.

Let A and B denote the two graphs being matched with vertex sets V_A and V_B, respectively. The complexity of the FGM algorithm is $O(n^2 m^2)$, where $n = |V_A|$ and $m = |V_B|$. Without loss of generality, we assume that $n \geq m$. The FGM algorithm uses a membership matrix $U = [u_{ij}]$, where u_{ij} represents the relative degree to which node $i \in V_A$ matches the node $j \in V_B$, i.e., $\mathbf{U}$ is the fuzzy assignment matrix. The objective function used for the FGM algorithm is:

$$J(\mathbf{U}, \mathbf{C}) = \sum_{\substack{i=1 \\ (i,j) \neq (n+1, m+1)}}^{n+1} \sum_{j=1}^{m+1} u_{ij}^2 f(c_{ij}) + \eta \sum_{\substack{i=1 \\ (i,j) \neq (n+1, m+1)}}^{n+1} \sum_{j=1}^{m+1} u_{ij}(1 - u_{ij}). \tag{2}$$

In (2), η is a constant that controls the relative influence of the two terms in the minimization process, c_{ij} represents the absolute compatibility between nodes $i \in V_A$, $j \in V_B$ (given the fuzzy assignments $\mathbf{U}$), taking into account the attributes of the edges incident on nodes i and j and those of the neighboring nodes of i and j. In other words, $C = [c_{ij}]$ is the compatibility matrix. The function $f() = exp(-\beta c_{ij})$ is a decreasing function that converts c_{ij} to a kind of "dissimilarity." In Section 3.2.2, we provide a more detailed discussion on how c_{ij}s can be chosen. As mentioned above, the compatibilities c_{ij} depend on U. Similarly, the assignments $\mathbf{U}$ depend on the compatibilities $\mathbf{C}$. We update $\mathbf{U}$ and $\mathbf{C}$ in an alternating fashion, giving rise to a relaxation process. To accomplish robust matching, we introduce dummy nodes in each of the graphs being compared. Node $n + 1$ in graph A and node $m + 1$ in graph B represent dummy nodes. When a particular node in graph A does not match any of the nodes in graph B, it can be assigned to the dummy node of graph B and vice versa. The dummy node enables us to minimize the objective function J subject to the following constraints:

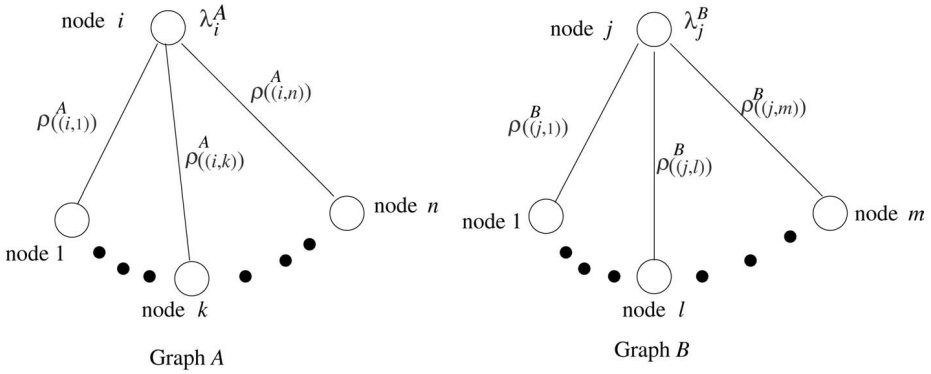


Fig. 1. Computation of compatibility.

$$\left.\begin{array}{ll} \sum_{j=1}^{m+1} u_{ij} = 1, & for \quad i = 1, \ldots, n \\ \sum_{i=1}^{n+1} u_{ij} = 1, & for \quad j = 1, \ldots, m \\ u_{ij} \geq 0 \quad \forall & i \text{ and } j \end{array}\right\}. \tag{3}$$

The first term in (2) is minimized if the matching degrees u_{ij} are high whenever the compatibilities c_{ij} are high. However, ideally, we want $u_{ij} \in \{0, 1\}$. To accomplish this goal, we add the second (entropy) term in (2) which tries to push the values of u_{ij} toward either zero or one. The update equations for the FGM algorithm are based on the Lagrange method and use the Karush-Kuhn-Tucker conditions. The details can be found in [65], [66].

3.2.2 Extension of FGM to FARGs

To extend FGM to FARGs, we need to redefine the compatibility $c_{ij} \in [0, 1]$, which is a quantitative measure of the (absolute) degree of match between node $i \in V_A$ and node $j \in V_B$, given the current fuzzy assignment matrix U. We start with the definition of compatibility c_{ij} as

$$c_{ij} = w_{ij}^{0.5} \sum_{\substack{k=1 \\ (k \neq i),(l \neq j)}}^{n+1} \sum_{l=1}^{m+1} \frac{m_{kl} m'_{kl}}{n_j^B} \tag{4}$$
$$i = 1, \ldots, n+1, \text{ and } j = 1, \ldots, m+1,$$

where w_{ij} is the degree of match between (the attributes of) node $i \in V_A$ and node $j \in V_B$, $m_{kl} \in [0, 1]$ is the matching score between the edge $(i, k) \in E_A$ and edge $(j, l) \in E_B$, M is the matrix $[m_{kl}]$, $M' = [m'_{kl}]$ is the crisp assignment matrix closest to M satisfying the constraints in (3) for $i = 1, \ldots, n+1$ and $j = 1, \ldots, m+1$, and n_j^B is a normalization factor equal to the number of edges (with nonzero weights or attribute values) that are incident on node $j \in V_B$. Note that M' acts as a filter so that each edge in graph B that is incident on node j contributes to c_{ij} only once. In other words, out of the double summation in (4), only (m) terms survive. Also, w_{ij} is raised to the power 0.5 for enhancement purposes. Fig. 1 illustrates the notation used.

To compute M', rather than apply the Sinkhorn technique [68] or other standard algorithms for the quadratic assignment problem, we use a greedy method. We identify the largest element in M, set it equal to 1, and zero out the remaining elements of the row and column corresponding to the largest element. We repeat this process m times. In theory, this method has a somewhat higher complexity of $O(nm^2)$ than the Sinkhorn procedure, but, in our experience, it is faster in practice.

The pth node-attribute of the FARG, for example, has n_{a_p} linguistic values associated with it. The value of the pth attribute a_p of node i of graph A is denoted by $\mathcal{A}_{ip}^A = (a_{ip1}^A, \ldots, a_{ipn_{a_p}}^A)$. Similarly, the value of the pth attribute of edge (i, k) of graph A is denoted by $\mathcal{R}_{(i,k)p}^A = (r_{(i,k)p1}^A, \ldots, r_{(i,k)pn_{r_p}}^A)$. We define the matching degree w_{ij} between the (attributes of) node $i \in V_A$ and node $j \in V_B$ as

$$w_{ij} = \begin{cases} \dfrac{\sum_{p=1}^{n_A} W(p)(1 - \max\limits_{1 \leq q \leq n_{a_p}} |a_{ipq}^A - a_{jpq}^B|)}{\sum_{p=1}^{n_A} W(p)} & \text{if } i \neq n+1 \text{ and } j \neq m+1 \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

where $W(p)$ is the weight associated with attribute a_p as specified by the user. In the very first iteration, we initialize the compatibilities as $f(c_{ij}) = exp\{-w_{ij}\}$ and use (4) in later iterations. We also define

$$m_{kl} = u_{kl}^{0.5} \ \min\left(w_{kl}, \sum_{p=1}^{n_R} V(p)\left(1 - \max_{1 \leq q \leq n_{r_p}} \zeta_{(i,k)(j,l),pq}^{AB}\right)\right) \tag{6}$$
$$\text{if } k \neq n+1 \text{ and } l \neq m+1,$$

where $V(p)$ is the weight associated with relation r_p and

$$\zeta_{(i,k)(j,l),pq}^{AB} = \begin{cases} 1 & \text{if } r_{(i,k)pq}^A = 0 \text{ or } r_{(j,l)pq}^B = 0 \\ |r_{(i,k)pq}^A - r_{(j,l)pq}^B| & \text{otherwise.} \end{cases} \tag{7}$$

The above equations are used in (4) to compute the compatibility. For the dummy node and edges associated with the dummy node, the attribute values are initialized to zero.

4 Fuzzy Content-Based Retrieval System

4.1 Overview of FIRST

Fig. 2 shows a block diagram of FIRST. FIRST has three main components: FARG generation, database indexing,

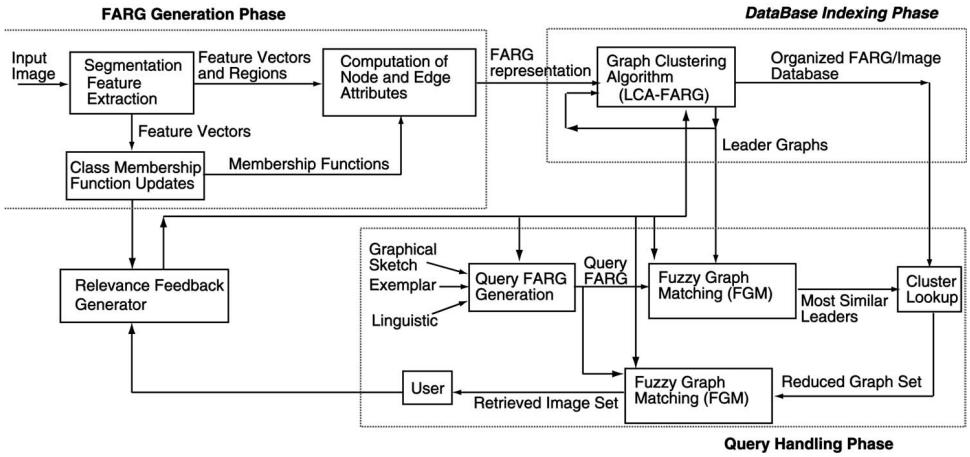


Fig. 2. Block diagram of FIRST.

and query handling. In this section, we briefly describe these components. More details are provided later in this section. Relevance feedback is still in the experimental stage and has not been integrated into our system at this point.

4.1.1 FARG Generation

The input image to be added to the database is converted to a FARG as follows: The image is first segmented into regions, and the regions are labeled. A feature vector x_j, that represents texture, color, or other features, is extracted from each region j. As described in Section 3.1, for each node-attribute a_i, we compute the value $\mathcal{A}_{ji} = (L_{i1}(x_j), \ldots, L_{in_{a_i}}(x_j))$ of node-attribute a_i of region j. In our application, we used six node-attributes, i.e., $n_A = 6$. The attributes were *class_label*, *intensity*, *hue*, *contrast*, *homogeneity*, and *size*. The number of linguistic labels for the six attributes were $n_{a_1} = 16$, $n_{a_2} = 3$, $n_{a_3} = 6$, $n_{a_4} = 3$, $n_{a_5} = 3$, and $n_{a_6} = 3$, respectively. The estimation of the membership functions $L_{ik}()$ for the six attributes is discussed in Sections 4.2 and 4.4. In addition to node-attributes, we use an edge-attribute called *spatial_relation* with the five linguistic labels $left\_of$, $right\_of$, $above$, $below$, and $surrounded\_by$. Computation of the edge-attribute is discussed in Sections 4.2 and 4.3.

4.1.2 Database Organization

We use a leader approach to cluster incoming FARGs incrementally so that the FARG database is organized in clusters of similar images [69]. Each cluster is represented by a leader FARG. The incremental nature of the proposed graph clustering algorithm allows us to build the image database gradually and add new images with minimal cost. The graph clustering algorithm is explained in more detail in Section 5.

4.1.3 Query Handling

The queries are first converted to a FARG representation. In the case of an exemplar query, the system displays a set of images picked randomly from the database. The user can either select one of them as the exemplar image or request the system to display more examples. The system uses the FARG representation of the selected image to search for similar ones in the database. In the case of a graphical sketch, the user is allowed to draw regions and ascribe attribute values to each region based on a menu. The sketch is converted to a FARG by creating a node for each region, recording the ascribed attributes at each node, and computing the edge attributes based on the spatial placement of the regions in the sketch. For a linguistic query, a FARG node is created for each category label (e.g., *sky*) that appears in the query, and a pair of nodes in the FARG representation is joined by an edge whenever a spatial relation is specified in the query (e.g., water region *left of* land region). If the linguistic description specifies any attributes (e.g., a *small* water region), then the attribute in the corresponding node is assigned an appropriate value (see Section 2.1). Otherwise, the attribute is assigned a default value (e.g., *medium*). Once the query is converted to a FARG, it is compared with the leader FARGs of the database using FGM. The leader(s) closest to the query FARG are found, and the FARGs belonging to the corresponding cluster(s) are obtained by looking up the FARG database. Only the FARGs in this reduced set are compared with the query FARG and ranked for retrieval.

4.2 FARG Generation for the Synthetic Database

In this section, we outline the process of computing the FARG representation for the synthetic database of 1,240 images. Each image in this data set was synthesized by filling randomly placed shapes (such as rectangles, squares, ellipses and circles) with textures selected from 2,384 source images obtained from the VisTex database of MIT Media Lab. The 2,384 source images came from 16 classes. The data set will be discussed in more detail in Section 6.1.

To generate the membership functions for the linguistic labels $L_{1k}()$ of attribute $a_1 = class\_label$, we used Gabor features at four scales and six orientations to obtain a 48D vector from each of the source 2,384 images. The 48D feature vectors were then compressed to 12D vectors by using principal component analysis. To estimate $L_{1k}()$, we applied the RAGMD algorithm [57] to the set $\{\mathbf{x}_j\}$ of 12D feature vectors with class label k. RAGMD automatically determines the "optimal" number c_k of Gaussian components required to model the set $\{\mathbf{x}_j\}$ and produces a set of mean vectors $\{\mathbf{m}_{k1}, \mathbf{m}_{k2}, \ldots, \mathbf{m}_{kc_k}\}$, covariance matrices $\{\mathbf{C}_{k1}, \mathbf{C}_{k2}, \ldots, \mathbf{C}_{kc_k}\}$,

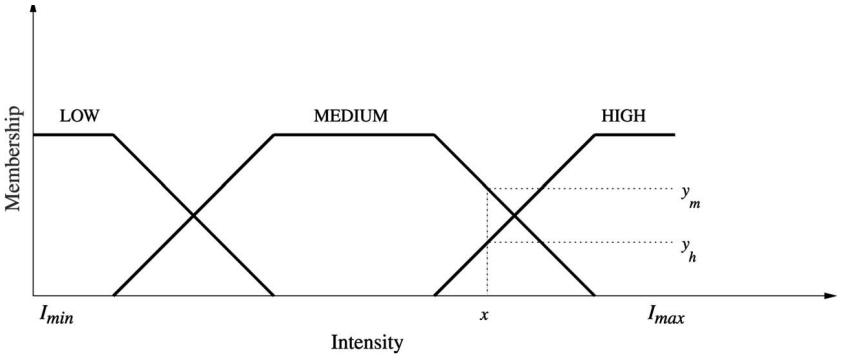


Fig. 3. Fuzzy membership functions for intensity attribute. The value of $a_2 = (0, y_m, y_h)$.

and prior probabilities $\{P_{k1}, P_{k2}, \ldots, P_{kc_k}\}$, for the c_k components. We repeated this process for all class labels $k = 1, 2, \ldots, 16$. We then defined the multidimensional membership functions

$$L_{1k}(\mathbf{x}) = \frac{\mu_{1k}(\mathbf{x})}{\sum_{j=1}^{16} \mu_{1j}(\mathbf{x})}, k = 1, 2, \ldots, 16, \tag{8}$$

where

$$\mu_{1k}(\mathbf{x}) = \sum_{p=1}^{c_k} P_{kp} \exp\left\{ -\frac{(\mathbf{x} - \mathbf{m}_{kp})^T \mathbf{C}_{kp}^{-1} (\mathbf{x} - \mathbf{m}_{kp})}{2} \right\}, k = 1, \ldots, 16. \tag{9}$$

To compute the intensity feature value for a region, the intensity values were averaged over all pixels in the region. The membership functions for the linguistic labels of *Intensity* were defined over the domain $[I_{min}, I_{max}]$, where I_{min} and I_{max} were, respectively, the minimum and maximum of the intensity values in the entire database. The membership functions for $a_2 = Intensity$ were then modeled as trapezoids over this domain, as shown in Fig. 3.

The hue attribute for a region is calculated using the standard formula [70], and six equally spaced trapezoidal membership functions were used for this feature. The saturation feature was not found to be useful and, hence, we did not use it. We calculated the values of attributes $a_4 = contrast$ and $a_5 = homogeneity$ for a region as described in [23]. The last attribute we used was $a_6 = size$. The size of the region is computed as the number of pixels in that region. For a_4, a_5, and a_6, the procedure used for computing of the membership values was the same as that of intensity.

4.3 Computation of Edge Attributes

We use only one edge-attribute ("spatial_relation") in our application. However, other edge-attributes (e.g., "adjacency") could also be used. Let B_j and B_l denote the jth and lth regions, respectively. For the edge-attribute, $r_1 = spatial\_relation$, the membership functions, $S_{1k}(B_j, B_l)$, $k = 1, \ldots, n_{r1}$, where $n_{r1} = 5$ is the number of linguistic labels for $spatial\_relation$, are used to compute the value of the fuzzy attribute $\mathcal{R}_{e1} = (S_{11}(B_j, B_l), \ldots, S_{1n_{r_1}}(B_j, B_l))$ for each edge $e = (j, l)$. The five linguistic labels we used were $left\_of$, $right\_of$, $above$, $below$, and $surrounded\_by$. We make use of a fuzzy morphological approach [37] to determine $S_{1k}(B_j, B_l)$. We use a fast implementation of this method [37] to compute $S_{11}(B_j, B_l) = left\_of, \ldots, S_{14}(B_j, B_l) = below$. The membership value for $S_{15}(B_j, B_l) = "surrounded\_by"$ is simply the minimum of the fuzzy membership values for the other four spatial relations, i.e., $S_{15}(B_j,B_l) = min\{S_{11}(B_j,B_l), S_{12}(B_j,B_l), S_{13}(B_j, B_l), S_{14}(B_j, B_l)\}$.

4.4 Computation of FARGs for the Outdoor Scene Images

The outdoor scene image database consisted of 500 images from the NETRA data set, which originally come from Corel Inc., and another 526 images obtained directly from Corel Inc. This gave us a total of 1,026 images in the database. NETRA uses an automatic segmentation algorithm and, as is to be expected, there are many instances of over and undersegmentation. We improved the segmentation by merging oversegmented regions in a semi-automated fashion. Since the focus of our work was on image retrieval and not on image segmentation algorithms, we decided to segment the remaining Corel images manually.

For the $class\_label$ attribute for each region, the linguistic values were chosen from the set {*Sky*, *Snow*, *Tree*, *Mountain*, *Water*, *Ice*, *Rock*, *Clouds*, *Leaves*, *Flower*, *Stem*, *None*, *Sea*, *Beach*, *Land*, *Boat*, *Structure*, *Animal*, *Grass*, *Bushes*, *Road*, *Sun*}. Again, since our main objective was to test the image retrieval algorithm and not to build an effective classifier for outdoor scene objects, we assigned the membership values in each of these linguistic labels for a given region manually. The membership functions and values for the attributes *intensity*, *hue*, *contrast*, *homogeneity*, and *size*, as well as for the edge-attribute, were computed exactly as in the case of the synthetic images.

5 LCA-FARG: A Leader Clustering Algorithm for FARGs

The Leader Clustering Algorithm (LCA) [69] is an efficient sequential (one-pass) clustering algorithm that has been used in pattern recognition extensively. We use this approach to design an indexing scheme to partition graphs into clusters and aid faster retrieval. Its complexity is linear in n, where n is the number of images in the data set. This algorithm partitions the input data (FARGs) by using a leader (prototype) FARG for each cluster such that every element (FARG) in a cluster is within a specified distance

(threshold) T of the leader. The leader is constructed in such a manner as to be a good representative of that particular cluster. This clustering approach, applied to an image database, is centered around a dissimilarity measure δ between two graphs G_1 and G_2 defined as:

$$\delta(G_1, G_2; \mathbf{V}) = \frac{1}{m} \sum_{p=1}^{n} \sum_{q=1}^{m} v_{pq}^2 f(c_{pq}) \quad 0 \leq \delta() \leq 1. \tag{10}$$

In (10), $\mathbf{V} = [v_{pq}]$ is the closest crisp assignment matrix to $\mathbf{U}$, which is obtained after the FGM process applied to G_1 and G_2. Note that the dummy rows and columns in $\mathbf{V}$ are not included in the measure and that $\delta()$ is normalized to have values between 0 and 1. The LCA-FARG algorithm is straightforward and is summarized below.

LCA-FARG Algorithm
Initialize the first Leader G_{L_1} as the first input graph; $c = 1$;
REPEAT Read a new input graph G_I;
For $j = 1, \ldots, c$, run FGM and compute $\delta(G_I, G_{L_j})$;
Find closest leader $G_{Lj_{min}}$ such that
$\delta(G_I, G_{Lj_{min}}) \leq \delta(G_I, G_{L_j})$ for $j = 1, \ldots, c$;
IF $\delta(G_I, G_{Lj_{min}}) \leq T$ {
Assign graph G_I to cluster j_{min};
Update leader $G_{Lj_{min}}$ of cluster j_{min} using (11);
Update no. of graphs assigned to cluster j_{min} $n_{j_{min}} = n_{j_{min}} + 1$;
}
ELSE create new cluster { $c = c + 1$; $G_{L_c} = G_I$; }
UNTIL (all graphs are handled).

We now describe the leader updating step of the algorithm in more detail. The crisp assignment matrix $\mathbf{V}$ plays a central role in this update process. Let $k \in V_{G_i}$ and $l \in V_{G_{Lj_{min}}}$ be a matching pair. If the matching node l corresponding to the leader graph is not the dummy node, then the fuzzy value $\mathcal{A}_{li}^{G_{Lj_{min}}}$ of the attribute a_i at node l of the updated leader becomes a weighted average of the current fuzzy attribute value and the value $\mathcal{A}_{ki}^{G_I}$ of the corresponding attribute of the matching node k from the input graph G_I as shown below:

$$\mathcal{A}_{li}^{G_{Lj_{min}}} = \frac{n_{j_{min}} \mathcal{A}_{li}^{G_{Lj_{min}}} + \mathcal{A}_{ki}^{G_I}}{n_{j_{min}} + 1}, i = 1, \ldots, n_A. \tag{11}$$

If, however, the matching node from the leader graph is a dummy node (i.e., node k from the input graph does not match any nodes in the leader graph), a new node is created in the leader graph with the same attribute values as the matching node in the input graph except that these values are weighted down by the factor $(n_{j_{min}} + 1)$. This weighting down procedure ensures that the leader graph does not change drastically. The attribute values of the edges corresponding to this node are updated by the same weighted averaging process.

LCA-FARG allows us to index new images in an incremental fashion. The disadvantage is that the clustering result depends on the order in which the images arrive. However, since our main goal is to group the data in a convenient form to minimize the search and not to find *clusters*, this should not be a serious issue. Our experimental results confirm this as well.

6 Experimental Results

6.1 Results on the Synthetic Image Database

In this section, we describe the synthetic image database in more detail and present retrieval results based on an exhaustive search of the database. As mentioned in Section 4.2, the synthetic database has been created out of the VisTex texture images of MIT Media Lab. We use a total of 149 images of size 512×512 from the VisTex database. The 149 images come from 16 classes, namely, *Bark*, *Brick*, *Clouds*, *Fabric*, *Flowers*, *Food*, *Grass*, *Leaves*, *Metal*, *Paintings*, *Sand*, *Stone*, *Terrain*, *Tile*, *Water*, and *Wood*. Each of the 149 images are divided into 16 nonoverlapping regions to generate 2,384 images of size 128×128. Based on these 2,384 "source images," we synthesize images that contain multiple regions. In the first step, two or three predefined shapes (such as rectangles, squares, ellipses, and circles) are randomly selected, sized, and placed in random locations in the synthetic image. The shapes are then filled with a texture selected randomly from one of the 2,384 images mentioned above. We generate a total of 1,000 images in this manner. To this set of 1,000 images, we add an additional set of 240 images, thus making it a database of 1,240 images. These extra 240 images are generated as follows: We first pick 20 images randomly from the set of synthetic 1,000 images. These 20 images are utilized in generating two sets of images, one containing 160 images, which we refer to as Extra Image Set 1, and the other containing 80, which we refer to as Extra Image Set 2. To produce Extra Image Set 1, two regions in each of the 20 images are chosen at random. The two regions are then displaced in the positive and negative horizontal and vertical directions, one direction at a time. This method produces eight images from each of the 20 images, all of which are roughly similar in terms of spatial relations. To generate the 80 images of Extra Image Set 2, we use the following procedure. In each of the 20 images chosen randomly from the 1,000 image data set, a region is chosen at random and the selected texture is replaced by a different texture with the same class label (i.e., by choosing a different image with the same class label). This replacement is performed four times to generate four similar images for each of the 20 images. Extra Image Set 2 is added to make it harder for the system to identify and retrieve the relevant images.

We use the standard measures, precision and recall, in different forms, to evaluate the results [39].

$$\text{Recall} = \frac{\text{Number of images retrieved and relevant}}{\text{Total number of relevant images in the database}}. \tag{12}$$

$$\text{Precision} = \frac{\text{Number of images retrieved and relevant}}{\text{Total number of retrieved images}}. \tag{13}$$

Based on these measures, we define the following:
The Average Recall Rate (*AVRR*) is given by

$$AVRR = \frac{1}{Q} \left\{ \sum_{j=1}^{Q} \frac{\sum_{i=1}^{32} Rank_i}{N_r} \right\}, \tag{14}$$

TABLE 1
Evaluation of Results for Multiattributed FGM Using the Synthetic Image Database (1,240 images)

Measure	Multiple Attributed FGM		
	Weights 1	Weights 2	Weights 3
AVRR	7.04	6.87	6.61
Recall8	86.38	85.88	86.75
Recall16	89.50	90.00	91.25
Recall32	89.88	93.63	92.13

where the rank of any of the retrieved images is defined to be its position in the list of retrieved images, provided that image is one of the relevant images in the database. The rank is defined to be zero otherwise. N_r is the number of relevant images in the database, and Q is the number of queries performed. In our case, the number of images retrieved was 32, and N_r was less than 32. Hence, when all relevant images are in the retrieved set, ideally, $AVRR = (N_r + 1)/2$.

We define *Recall8* as the percentage of images in a list of eight retrieved images that belong to the set of relevant images in the database (see definition in (12)). Similarly, we define *Recall16* and *Recall32* for retrieved sets of 16 and 32 images, respectively. Note that, in all three cases, the denominator is the same.

We apply FGM to this entire data set of 1,240 images. One hundred images are randomly chosen from the data set of 240 extra images (i.e., Extra Image Set 1 plus Extra Image Set 2) as queries and are used to query the entire database by exhaustive search. The average values of *AVRR*, *Recall8*, *Recall16*, and *Recall32* measures are used to evaluate the retrieval results. In these measures, we use a retrieved set of size 32 and the number of queries made was $Q = 100$. For each query, the number of relevant images in the database of 1,240 images is eight, i.e., the eight similar images from Extra Image Set 1. (In addition, four other images from Extra Image Set 2 will be somewhat similar, but we do not count these). Note that the ideal value for $AVRR$ is 4.5.

Table 1 shows the results of the aforementioned experiment. We consider three different weighting schemes for this experiment. In the first weighting scheme, we assign equal weights to all six attributes. In the second weighting scheme, we use weights of (2, 0.5, 1, 1, 1, 0.5)/6, respectively, for each of the attributes and, in the third weighting scheme, we use weights of (2, 0.5, 2, 0.5, 0.5, 0.5, 0.5)/6, respectively. Ideally, these weights are subjective, and should be determined by relevance feedback for each user. As can be seen in Table 1, by both *AVRR* and *Recall8* measures, Weighting Scheme 3 produces the best results. We also have run the experiment considering only one attribute at a time. However, it is not possible to come up with an optimal weight distribution scheme based on these results. This indicates that the attributes interact with each other in a complex way.

6.2 Results on the Outdoor Scene Image Database

The outdoor scene image database consists of 1,026 images (see Section 4.4). Since we do not have the ground truth that gives us information about how many and which images are relevant for any given query, we use a different set of measures to evaluate the results. We now describe these measures.

The original Corel images are organized into directories. Each directory contains several images pertaining to a specific theme, e.g., *wild flowers*, *sunset and nature*, *sea shore*, *fish*, etc. The directory classification, as provided by Corel, is not always accurate. However, we use it as a measure to define the Directory Classification Rate (*DCR*). *DCR* is defined to be the percentage of images in the retrieved set that belong to the same directory as that of the query. Another measure we use is Absolute Displacement (*AD*). Let $r_1, \ldots, r_{m-1}$, and r_m be the ranks given by the system to the set of retrieved images. Let $h_1, \ldots, h_{m-1}$, and h_m be the corresponding ranks as given by the user. Then, *AD* was defined as follows:

$$AD = \sum_{i=1}^{m} |r_i - h_i|. \tag{15}$$

In addition, we also use the Relative Weighted Displacement (*RWD*) measure from [6]. Let the user label each image in the retrieved set as "a," "b," or "c," where "a" denotes an image that is similar to the query as perceived by the user, "b" one that is somewhat similar, and "c" one that is dissimilar. Then, RWD is defined as follows:

$$RWD = \frac{\sum_{i=1}^{m} w_i |r_i - h_i|}{\sum_{i=1}^{m} w_i}, \tag{16}$$

where w_i is 0.8, 0.5, and 0.05 for the cases when the image is labeled "a," "b," and "c," respectively. The weights used in the case of *RWD* do not punish the measure if nonsimilar images are ranked high. We therefore modify w_i to be 0.8 for the case when the image is labeled "c."

We use the weights (2,0.5,2,0.5,0.5,0.5,0.5)/6 for the six node-attributes: *Label, Intensity, Hue, Contrast, Homogeneity, and Size* (see Section 4.4). One hundred random queries are picked from all 24 directories. Each query is used to retrieve the 10 best matches from the entire database by exhaustive search. In the results reported here, the database is queried by five persons and the performance is averaged. The results of evaluation are shown in Table 2. It can be seen that, according to the *AD* measure, probably the most reliable measure that has been used, the performance of the algorithm is good.

6.3 Evaluation of the Indexing Method

In this section, we describe the results obtained when the indexing algorithm (based on LCA-FARG) was tested for its performance on the outdoor scene image database. We have tried different values for the threshold T in each of these runs. The number of clusters found by the algorithm when $T = 0.35$ is 73. For $T = 0.40$ and 0.45, the numbers are 42 and 25, respectively. The choice $T = 0.40$ has been found to

TABLE 2
Evaluation of FGM Results on Outdoor Scene Database

Measure	Multi-Attribute
Absolute Displacement	1.58
RW Displacement	0.13
MRW Displacement	0.17
No. of similar images	3.59
No. of somewhat-similar images	2.67
Number of non-similar images	3.67
Classification Rate (DCR)	56.22

be a good compromise between computational expense and performance. For this value of T, LCA-FARG takes about 15 minutes on a 450 MHz Pentium II machine to cluster the image graphs offline. Figs. 4 and 5 show the 42 images that are closest to the 42 prototype FARGs obtained by LCA-FARG. It can be seen that the prototypes capture the diversity of the data set very well.

Once the clustering process is complete, we use each image in the database as a query and find the 10 best matches based on an exhaustive search of the database. We compare these results with those obtained when indexing has been used. In the case of the indexed database, the query is initially matched with only the leaders of the clusters, and a predetermined number of the best matching leaders are then chosen. The query is then matched with each of the images in the clusters corresponding to the chosen leaders, and the 10 best matches from these clusters are retrieved. Assuming that the total number of clusters in the database is C, the number of images in the database is N, and the predetermined number of clusters to be searched

Fig. 4. Images closest to the cluster prototypes (1-20) in the outdoor scene image data set.

Fig. 5. Images closest to the cluster prototypes (21-42) in the outdoor scene image data set.

is J, then the number of matches to be performed is $M = C + N_{C_1} + \ldots + N_{C_J}$, where $C_i, i = 1, \ldots, J$ are the J clusters corresponding to the top J best matching cluster prototypes, N_{C_i} is the number of images in cluster i. If all clusters had an equal number of images, this expression would reduce to $M = C + (JN)/C$. When compared to an exhaustive search involving N matches, this is a significant saving. For example, if $N = 1,026$, C is 42, and J is 4, we would have to perform only about 100 matches, whereas an exhaustive approach would require 1,026 matches. In our case, the retrieval time in the indexed case is 4 to 5 times shorter than that for the exhaustive search.

As explained above, each query results in a prespecified number of retrieved images. We refer to the set of retrieved images as the retrieved set. The retrieved sets for each query in both experiments (exhaustive and cluster-based) are then used to evaluate the accuracy of retrieval from the clustered data set. We use the following two measures for evaluation:

TABLE 3
Evaluation of Clustering on Outdoor Scene Image Data Set

No. of Clusters Searched	Missing Images		Rank Difference		Time Ratio	
	5	10	5	10	5	10
2	0.92	1.79	0.98	1.91	5.1	5.1
3	0.61	1.35	0.9	1.51	4.9	4.9
4	0.35	0.9	0.38	1.22	4.8	4.8
5	0.33	0.88	0.35	1.19	4.8	4.8

Cardinality of retrieved set = 5 or 10.

Fig. 6. Comparison between (a) exhaustive and (b) clustered searches for the NETRA data set when the image at the top is used as the query.

- **Missing Images**. The average number of missing images is the average (over all queries) of the number of images from the retrieved set of the exhaustive search that are missing from the retrieved set of the cluster-based search.
- **Rank Difference**. The average rank difference is the average (over all queries) of the sum of the absolute differences in ranks between the retrieved set of the exhaustive search and the retrieved set of the cluster-based search.

We carry out the comparison for several values of J, i.e., the predetermined number of clusters to be searched. Table 3 shows the results in these cases, with comparisons being performed when the cardinality of the retrieved set is 5 and 10, respectively. As can be seen in the table, the results are better when the top five clusters were searched. In fact, there is very little difference between the results for the top four and five clusters.

Fig. 6 shows a comparison of the (a) exhaustive and (b) index-based searches for an example query shown at the top of the figure. Only the last retrieved image in (b) is

different and seems to be a result of the leader of the cluster acquiring slightly different properties (in this case, it seems to be the presence of red flowers) due to the introduction of other images into the cluster.

7 Summary and Conclusions

Uncertainty pervades every aspect of CBIR. This is because image content cannot be described and represented easily, user queries are ill-posed, the similarity measure to be used is not precisely defined, and relevance feedback given by the user is approximate. To address these issues, fuzzy sets can be used to model the vagueness that is usually present in the image content, image indexing, user query, and the similarity measure. This allows us to retrieve relevant images that might be missed by traditional approaches. The plethora of aggregation connectives in fuzzy set theory permit us to define a similarity measure that is tailored to the application domain or the user's taste.

The fuzzy attributed relational graph (FARG) is a powerful model for representing image content in terms of regions and spatial relations between them. It is well-known that object labels are not crisp, and attribute values such as *small* and *somewhat*, as well as spatial relations such as *left of* and *below*, are handled much better by fuzzy techniques. Therefore, the representation can incorporate the vagueness associated with the attributes of the regions as well as those of the relations between the regions. FIRST uses a fast and efficient graph matching algorithm to compute the similarity between graphs. To improve the speed of the retrieval process, FIRST indexes FARGs by using a novel leader-clustering algorithm.

The FIRST system has been tested using synthetic and natural image databases of more than a 1,000 images. The experiments clearly illustrate the feasibility of the proposed approach. We have not shown the results of linguistic or sketch-based queries because these queries are also converted to FARGs and handled the same way as exemplar-based queries. Therefore, the results are very similar. A suite of performance measures, both standard and tailored, were used in evaluating the system. A comparison with exhaustive search clearly demonstrates the speedup associated with the proposed indexing scheme while preserving the accuracy.

In our experiments, we have used semi-automatic segmentation. A fully automated segmenter would be more error-prone. The FARG approach is quite robust to over-segmentation because, in that case, a single region is split into multiple regions with the same label. The corresponding FARG would be a supergraph of the actual one. However, since our retrieval is based on subgraph matching, this does not pose a serious problem. On the other hand, if two or more distinct regions are merged due to undersegmentation, then the resulting graph will be a subgraph of the actual one. For the merged region, the features extracted can sometimes be quite different from the correct values. When this happens, the region is likely to be misclassified, leading to poor matches.

Acknowledgments

The authors would like to gratefully acknowledge the support of this work by Korea Telecom's Multimedia Technology Research Laboratory.

References

[1] Y. Rui, T.S. Huang, and S.F. Chang, "Image Retrieval: Current Techniques, Promising Directions and Open Issues," *J. Visual Comm. and Image Representation,* vol. 10, no. 4, pp. 39-62, Apr. 1999.

[2] R.C. Jain, "Special Issue on Visual Information Management," *Comm. ACM,* vol. 40, no. 12, pp. 30-32, Dec. 1997.

[3] V.N. Gudivada and V. Raghavan, "Special Issue on Content-Based Image Retrieval Systems," *Computer,* vol. 28, no. 9, Sept. 1995.

[4] A. Pentland and R. Picard, "Special Issue on Digital Libraries," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 8, pp. 783-789, Aug. 1996.

[5] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early Years," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 12, pp. 1349-1380, Dec. 2000.

[6] S. Santini and R. Jain, "Similarity Measures," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 9, pp. 871-883, Sept. 1999.

[7] V.N. Gudivada and V.V. Raghavan, "Design and Evaluation of Algorithms for Image Retrieval by Spatial Similarity," *ACM Trans. Information Systems,* vol. 13, no. 2, pp. 115-144, Apr. 1995.

[8] T. Hermes, C. Klauck, J. Kreys, and J. Zhang, "Image Retrieval for Information Systems," *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases,* pp. 394-407, Feb. 1995.

[9] D.A. Forsyth, J. Malik, T.K. Leung, C. Bregler, C. Carson, H. Greenspan, and M.M. Fleck, "Finding Pictures of Objects in Large Collections of Images," *Proc. Int'l Workshop Object Recognition for Computer Vision,* pp. 335-360, Apr. 1996.

[10] J.R. Smith and S.F. Chang, "VisualSEEk: A Fully Automated Content-Based Image Query System," *Proc. ACM Int'l Conf. Multimedia,* pp. 87-98, Nov. 1996.

[11] A. Del Bimbo and E. Vicario, "Using Weighted Spatial Relationships in Retrieval by Visual Contents," *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries,* pp. 35-40, June 1998.

[12] W.Y. Ma and B.S. Manjunath, "NETRA: A Toolbox for Navigating Large Image Databases," *Proc. IEEE Int'l Conf. Image Processing,* pp. 568-571, June 1997.

[13] G.M. Petrakis and C. Faloutsos, "Similarity Searching in Large Image Databases," *IEEE Trans. Knowledge and Data Eng.,* vol. 9, no. 3, pp. 435-447, May/June 1997.

[14] S. Medasani and R. Krishnapuram, "A Fuzzy Approach to Content-Based Image Retrieval," *Proc. IEEE Int'l Conf. Fuzzy Systems,* vol. 3, pp. 1251-1260, 1999.

[15] J. Freeman, "The Modeling of Spatial Relations," *Computer Graphics and Image Processing,* vol. 4, pp. 156-171, 1975.

[16] R. Krishnapuram, J.M. Keller, and Y. Ma, "Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 3, pp. 222-233, Aug. 1993.

[17] K.P. Chan and Y.S. Cheung, "Fuzzy-Attribute Graph with Application to Chinese Character Recognition," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 22, no. 1, pp. 153-160, Jan./Feb. 1992.

[18] S. Gold and A. Rangarajan, "A Graduated Assignment Algorithm for Graph Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, pp. 377-387, Apr. 1996.

[19] M.J. Swain and D.H. Ballard, "Color Indexing," *Int'l J. Computer Vision,* vol. 7, no. 1, pp. 11-32, 1991.

[20] M. Stricker and M. Orengo, "Similarity of Color Images," *Proc. SPIE Conf. on Storage and Retrieval for Image and Video Databases III,* W.R. Niblack and R.C. Jain, eds., pp. 381-392, 1995.

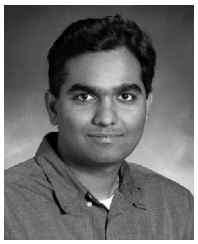
[21] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Region-Based Image Querying," *Proc. CVPR 97 Workshop Content-Based Access of Images and Video Libraries,* pp. 42-49, 1997.

[22] J.R. Smith and S.F. Chang, "Tools and Techniques for Color Image Retrieval," *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases IV,* pp. 426-437, 1995.

[23] M.R. Haralick, K. Shanmugam, and I. Dinstein, "Texture Features for Image Classification," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 3, no. 6, pp. 610-621, 1973.

[24] H. Tamura, S. Morei, and T. Yamawaki, "Texture Features Corresponding to Visual Perception," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 8, no. 6, pp. 460-473, 1978.
[25] B.S. Manjunath and W.Y. Ma, "Texture Features for Browsing and Retrieval of Image Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 8, pp. 837-842, Aug. 1996.
[26] F. Liu and R.W. Picard, "Periodicity, Directionality, and Randomness: Wold Features for Image Modeling and Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 7, pp. 722-733, July 1996.
[27] B.S. Manjunath and W.Y. Ma, "A Pattern Thesaurus for Browsing Large Aerial Photographs," Technical Report 96-10, Univ. of California at Santa Barbara, 1996.
[28] W. Niblack, "The GBIC Project: Querying Images by Content Color Texture and Shape," *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases,* pp. 173-197, 1993.
[29] R. Mehrotra and J.E. Gary, "Similar-Shape Retrieval in Shape Data Management," *Computer,* vol. 28, no. 9, pp. 57-62, Sept. 1995.
[30] A. Del Bimbo and P. Pala, "Visual Image Retrieval by Elastic Matching of User Sketches," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 2, pp. 121-132, Feb. 1997.
[31] A. Jain and A. Vailaya, "Image Retrieval Using Color and Shape," *Pattern Recognition,* vol. 29, no. 8, pp. 1233-1244, 1996.
[32] B. Kimia, J. Chan, D. Bertrand, S. Coe, Z. Roadhouse, and H. Tek, "A Shock-Based Approach for Indexing of Image Databases Using Shape," *Proc. SPIE's Multimedia Storage and Archiving Systems II,* pp. 288-302, Nov. 1997.
[33] M.A. Eshera and K.S. Fu, "An Image Understanding System Using Attributed Symbolic Representation and Inexact Graph-Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, pp. 604-619, Sept. 1986.
[34] J.M. Keller and L. Sztandera, "Spatial Relations among Fuzzy Subsets of an Image," *Proc. Int'l Symp. Uncertainty Modeling and Analysis,* pp. 207-211, 1990.
[35] K. Miyajima and A. Ralescu, "Analysis of Spatial Relations between 2D Segmented Regions," *Proc. European Congress Fuzzy and Intelligent Technologies,* pp. 48-54, 1993.
[36] J.M. Keller and X. Wang, "Learning Spatial Relationships in Computer Vision," *Proc. Int'l Conf. Fuzzy Systems,* pp. 118-124, 1996.
[37] I. Bloch, "Fuzzy Relative Position between Objects in Image Processing: A Morphological Approach," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 7, pp. 657-664, July 1999.
[38] P. Matsakis and L. Wendling, "A New Way to Represent Relative Position between Areal Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 7, pp. 634-643, July 1999.
[39] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison Wesley, 1989.
[40] C.T. Meadow, R.R. Boyce, and D.H. Kraft, *Text Information Retrieval Systems,* second ed. Academic Press, 2000.
[41] D.H. Kraft and A. Bookstein, "Evaluation of Information Retrieval Systems: A Decision Theory Approach," *J. Am. Soc. Information Science,* vol. 29, pp. 31-40, 1978.
[42] H. Dyckhoff and W. Pedrycz, "Generalized Mean as a Model of Compensation Connectives," *Fuzzy Sets and Systems,* vol. 14, pp. 143-154, 1984.
[43] D. Dubois and H. Prade, "A Review of Fuzzy Set Aggregation Connectives," *Information Sciences,* vol. 36, nos. 1/2, pp. 85-121, 1985.
[44] M. Mizumoto, "Pictorial Representations of Fuzzy Connectives I: Cases of T-Norms, T-Conorms, and Averaging Operators," *Fuzzy Sets and Systems,* vol. 31, pp. 217-242, 1989.
[45] R. Krishnapuram and J. Lee, "Fuzzy Connective-Based Hierarchical Networks for Decision Making," *Fuzzy Sets and Systems,* vol. 46, no. 1, pp. 11-27, Feb. 1992.
[46] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra, "Fuzzy Connective-Based Hierarchical Networks for Decision Making," *IEEE Trans. Circuits and Video Technology,* vol. 8, no. 5, pp. 644-655, 1995.
[47] B. Bhanu, J. Peng, and S. Quing, "Learning Feature Relevance and Similarity Metrics in Image Databases," *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries,* pp. 14-18, 1998.
[48] M. Sugeno, "Fuzzy Measures and Fuzzy Integrals: A Survey," *Fuzzy Automata and Decision Process,* M.M. Gupta et al., eds., pp. 89-102, 1977.
[49] M. Grabisch, "On Equivalence Classes of Fuzzy Connectives: The Case of Fuzzy Integrals," *IEEE Trans. Fuzzy Systems,* vol. 8, no. 1, pp. 96-109, 1995.
[50] R.R. Yager, "Ordered Weighted Averaging Operators in Multicriteria Decision Making," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 18, pp. 183-190, 1998.
[51] H. Frigui, "Adaptive Image Retrieval Using the Fuzzy Integral," *Proc. North Am. Fuzzy Information Processing Soc. Conf.,* pp. 575-578, 1999.
[52] H. Samet, *The Design and Analysis of Spatial Data Structures.* Addison Wesley, 1990.
[53] D.A. White and R. Jain, "Similarity Indexing: Algorithms and Performance," *Proc. SPIE Storage and Retrieval for Image and Video Databases IV,* vol. 2670, pp. 62-73, Feb. 1996.
[54] R. Kurniawati, J.S. Jin, and J.A. Shepherd, "Techniques for Supporting Efficient Content-Based Retrieval in Multimedia Databases," *The Australian Computer J.,* vol. 29, no. 4, pp. 122-130, 1997.
[55] C. Faloutsos, *Searching Multimedia Databases by Content.* Kluwer Academic, 1996.
[56] K. Fukunaga and P.M. Narendra, "A Branch and Bound Algorithm for Computing k-Nearest Neighbors," *IEEE Trans. Computers,* vol. 24, no. 7, pp. 750-753, July 1975.
[57] S. Medasani and R. Krishnapuram, "Categorization of Image Databases for Efficient Retrieval Using Robust Mixture Decomposition," *Proc. IEEE Workshop Content-Based Access of Image and Video Lib,* pp. 50-54, June 1998.
[58] H.J. Zhang and D. Zhong, "A Scheme for Visual Feature Based Image Retrieval," *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases III,* pp. 36-46, 1995.
[59] Y. Rui, T.S. Huang, S. Mehrotra, and M. Ortega, "A Relevance Feedback Architecture for Content-Based Multimedia Information Retrieval System," *Proc. IEEE Workshop Content-Based Access of Images and Video Libraries,* pp. 82-89, 1997.
[60] C. Meilhac and C. Nastar, "Relevance Feedback and Category Search in Image Database," *Proc. IEEE Int'l Conf. Multimedia Computing and Systems,* pp. 512-517, June 1999.
[61] Y. Rui, T.S. Huang, and S. Mehrotra, "Relevance Feedback Techniques in Interactive Content-Based Image Retrieval," *Proc. IS&T and SPIE Conf. Storage and Retrieval of Image and Video Databases,* pp. 25-36, 1998.
[62] M.E.J. Wood, N.W. Campbell, and B.T. Thomas, "Iterative Refinement by Relevance Feedback in Content-Based Digital Image Retrieval," *Proc. Sixth ACM Int'l Conf. Multimedia,* pp. 13-20, Sept. 1998.
[63] Y. Choi, D. Kim, and R. Krishnapuram, "Relevance Feedback for Content-Based Image Retrieval Using the Choquet Integral," *Proc. IEEE Int'l Conf. Multimedia and Expo,* vol. 2, pp. 1207-1210, 2000.
[64] L. Shapiro and R.M. Haralick, "A Metric for Comparing Relational Descriptions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, pp. 90-94, 1985.
[65] R. Krishnapuram and R. Medasani, "A Fuzzy Approach to Graph Matching," *Proc. IFSA Congress Conf.,* pp. 1029-1033, Aug. 1999.
[66] S. Medasani, R. Krishnapuram, and Y. Choi, "Graph Matching by Relaxation of Fuzzy Assignments," *IEEE Trans. Fuzzy Systems,* vol. 9, no. 1, pp. 173-182, 2001.
[67] M.P. Windham, "Numerical Classification of Proximity Data with Assignment Measure," *J. Classification,* vol. 2, pp. 157-172, 1985.
[68] R. Sinkhorn, "A Relationship between Arbitrary Positive Matrices and Doubly Stochastic Matrices," *Annals of Math. Statistics,* vol. 35, pp. 876-879, 1964.
[69] J. Hartigan, *Clustering Algorithms.* Wiley, 1975.
[70] D.H. Ballard and C.M. Brown, *Computer Vision.* Prentice Hall, 1982.

Raghu Krishnapuram received the PhD degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, in 1987. From 1987 to 1997, he was on the faculty of the Department of Computer Engineering and Computer Science at the University of Missouri, Columbia. In 1997, he joined the Department of Mathematical and Computer Sciences at the Colorado School of Mines (CSM), Golden, Colorado, as a full professor. Currently, he is at IBM India Research Lab, New Delhi, managing the knowledge management group. His research encompasses many aspects of fuzzy set theory, neural networks, pattern recognition, computer vision, and image processing. He has published more than 160 papers in journals and conferences in these areas. He is currently an associate editor of the *IEEE Transactions on Fuzzy Systems* and he is a coauthor (with J. Bezdek, J. Keller, and N. Pal) of the book *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing* (Kluwer Academic, 1999). His current research interests include Web mining and e-commerce. He is a senior member of the IEEE and IEEE Computer Society.



Swarup Medasani received the PhD degree in computer engineering and computer science from the University of Missouri in 1998. During his graduate study, he worked as a research assistant/lecturer from 1994 to 1997 at the University of Missouri. He also worked as a visiting scholar at the Colorado school of Mines from 1997 to 1998. After graduation, he worked as a postdoctoral fellow at the Colorado School of Mines from 1998 to 1999 on fuzzy modeling approaches for content-based image retrieval. Since 1999, he has been employed by HRL Laboratories in Malibu, California, as a senior research staff scientist. He is currently working on developing vision-based applications for surveillance and intelligent vehicles. His research interests are in content-based image retrieval, fuzzy clustering, pattern recognition, machine learning, and computer vision. He has more than 30 publications in book chapters, refereed journals, and conferences. He is a member of the IEEE and IEEE Computer Society.

Sung-Hwan Jung received the BS, MS, and PhD degrees from Kyungpook National University, Korea, in 1979, 1983, and 1988, respectively. From 1983 to 1985, he was with the Electronics and Telecommunications Research Institute in Korea as a member of the research staff, where he developed a portable computer. In 1988, he joined the faculty of the Department of Computer Engineering at Changwon National University in Korea, where he is currently a full professor. From 1992 to 1994, he was a postdoctoral research staff member in the Department of Electrical and Computer Engineering at the University of California at Santa Barbara. From 1999 to 2000, he also worked for the Colorado School of Mines in Golden as a visiting professor. His research interests are in content-based image retrieval, image steganography and watermarking, internet-based remote monitoring, text mining, and bioinformatics. He is a member of the IEEE.

Young-Sik Choi received the BS and MS degrees in electronics from Yonsei University, Seoul, Korea, in 1985 and 1987, respectively, and the PhD degree in computer engineering from the University of Missouri, Columbia, in 1996. From 1988 to 1991, he was with the Research Center, Korea Telecom. From 1996 to 2001, he was a senior researcher in the Multimedia Technology Research Laboratory, Korea Telecom. He is currently an assistant professor in the Department of Computer Engineering, Hankuk Aviation University, Korea. His research encompasses many aspects of pattern recognition, information retrieval, and data mining. His current research interests include image and video retrieval, Web data mining, and intrusion detection systems. He is a member of the IEEE.

Rajesh Balasubramaniam received the BTech degree (chemical engineering) from the Indian Institute of Technology, Mumbai, in 1996, and the MS degree (mathematical and computer sciences) from the Colorado School of Mines, Golden, in 2000. He has been with Sun Microsystems Inc., Broomfield, Colorado, since May 1999, developing knowledge engineering and data mining software. His research interests are in pattern recognition, graph/search algorithms, clustering, and optimization theory.