# Addressing CBIR Efficiency, Effectiveness, and Retrieval Subjectivity Simultaneously

Ruofei Zhang
Dept. of Computer Science
SUNY at Binghamton, Binghamton, NY 13902
+1-607-777-6931
rzhang@binghamton.edu

Zhongfei (Mark) Zhang
Dept. of Computer Science
SUNY at Binghamton, Binghamton, NY 13902
+1-607-777-2935
zhongfei@cs.binghamton.edu

## ABSTRACT
This work is about Content Based Image Retrieval (CBIR), focusing on developing a Fast And Semantics-Tailored (FAST) image retrieval methodology. Specifically, the contributions of FAST methodology to the CBIR literature include: (1) development of a new indexing method based on fuzzy logic to incorporate color, texture, and shape information into a region based approach to improving the retrieval effectiveness and robustness (2) development of a new hierarchical indexing structure and the corresponding Hierarchical, Elimination-based A* Retrieval algorithm (HEAR) to significantly improve the retrieval efficiency without sacrificing the retrieval effectiveness; it is shown that HEAR is guaranteed to deliver a logarithm search in the average case (3) employment of user relevance feedbacks to tailor the semantic retrieval to each user's individualized query preference through the novel Indexing Tree Pruning (ITP) and Adaptive Region Weight Updating (ARWU) algorithms. Theoretical analysis and experimental evaluations show that FAST methodology holds a great promise in delivering fast and semantics-tailored image retrieval in CBIR.

## Categories and Subject Descriptors
H.3.1 [**Content Analysis and Indexing**]*: indexing methods*; H.3.3 [**Information Search and Retrieval**]: *clustering, relevance feedback, retrieval models, search process*; I.5.3 [**Pattern Recognition**]: Clustering - *similarity measures.*

## General Terms
Algorithms, Management, Performance, Design, Experimentation

## Keywords
Content Based Image Retrieval (CBIR), Fuzzy Logic, Region Based Features, Hierarchical Elimination-based A* Retrieval (HEAR), Indexing Tree Pruning (ITP), Adaptive Region Weight Updating (ARWU), Relevance Feedback, Fast And Semantics-Tailored Retrieval (FAST)


*MIR'03*, November 7, 2003, Berkeley, California, USA


## 1. INTRODUCTION
This work addresses the topic of general purpose content based image retrieval (CBIR). CBIR has received intensive attention in the literature of multimedia information indexing and retrieval since this area was started years ago, and consequently a broad range of techniques is proposed.

The majority of the early research focuses on global features of imagery. The most fundamental and popularly used feature is the color histogram and its variants, which was used in the classic systems such as IBM QBIC [5] and Berkeley Chabot [11]. Since color histograms do not carry spatial information, which was considered to be related to the semantics of image content, efforts have been reported in the literature to incorporate the spatial information into the histograms. One approach is to use the Color Correlograms, proposed by Huang et al. [6], to address this issue.

Recently region-based approaches have been shown to be more effective. A region-based retrieval system segments images into regions, and retrieves images based on the similarities derived from the regions. Berkeley Blobworld [1] and UCSB NeTra [10] compare images based on individual regions. Wang et al. [16] proposed an integrated region matching scheme called IRM allowing for matching a region in one image to several regions of another image. Later, Chen and Wang [4] proposed an improved approach called UFM based on applying fuzzy logic to the different region features to improve the retrieval effectiveness of IRM. Recently Jing et al. [7] presented a region-based inverted file structure analogous to that in text retrieval to index the image database, with each entry of the file corresponding to a cluster (called codeword) in the region space.

While the majority of the literature in CBIR focuses on the indexing and retrieval effectiveness, much less attention is paid to the indexing and retrieval efficiency issue. There are two reasons attributing to this status. First, historically CBIR research was motivated in the beginning to demonstrate that indexing directly in the image domain could deliver better retrieval effectiveness than indexing through collateral information such as key words, and this perception was carried on over the years; the retrieval efficiency issue, on the other hand, was not considered to be the focus of the research in the CBIR community, as the general perception was that this issue would be resolved by directly making use of the existing indexing methods developed from the high dimensional space data structure research. Examples of these data structures

include *k-d* tree, *R*-tree and its variants, *SS*-tree, and *TV*-tree. A good review on these data structures can be found in [9]. Second, technically many CBIR algorithms involve complicated distributions in a high dimensional feature space, and it is difficult to directly "order" features in such a high dimensional space using "normal" data structures.

While theoretically any high dimensional feature space indexing method (which almost all the CBIR methods employ) can use the existing high dimensional indexing data structures to address the retrieval efficiency, practically this is not the case because when the dimensionality becomes very high (which is true for almost all the CBIR methods and thus is called *curse of dimensionality* [9]) the overhead for online bookkeeping becomes so demanding that the overall savings in efficiency becomes questionable. Some data structures, such as *SS*-tree and *TV*-tree, attempt to address this problem, but their overall performances are limited due to the assumptions they are subject to [9]. Consequently, it is observed that the efficiency issue must be addressed with the specific indexing method [2]. Due to this consideration, a few efforts in the literature started to address the efficiency issue directly in designing CBIR algorithms. For example, Castelli et al [2] proposed a clustering and singular value decomposition method for approximate similarity search in a high dimensional feature space.

Since the evaluation of CBIR retrieval is typically subjective, in recent years methods incorporating user relevance feedbacks start to show promise in resolving this issue. Two directions of research are observed in incorporating user relevance feedback in CBIR: (1) developing a weighting scheme to explicitly "guide" the retrieval [12]. (2) applying machine learning techniques such as Bayesian net and Support Vector Machine to reduce the problem to a standard reasoning and classification problem [14].

Based on an extensive literature review, we have identified three problems in the current status of CBIR research: (1) the indexing effectiveness still needs to be improved (2) the retrieval efficiency needs to be addressed directly in the indexing method (3) the retrieval subjectivity issue needs to be further addressed to better deliver a semantic retrieval. This work is motivated to address these three issues simultaneously. The ultimate goal of this project is to design a CBIR methodology that can deliver fast and semantics-tailored image retrieval capability. By fast, we mean the efficiency issue is well addressed; by semantics-tailored, we mean that the user query preference is inferred online to allow an individualized retrieval to better address the retrieval effectiveness and subjectivity issues. Consequently, we call the methodology FAST, which is a significant extension of our previous work FUZZYCLUB [17]. The contributions of FAST are reflected in these three aspects. Below we address these three aspects of FAST methodology, and report the experimental evaluations before we conclude this paper.

## 2. INDEXING SCHEME

Given an image, FAST first segments it into a group of regions based on a feature space consisting of color and texture features using the modified *k* means algorithm, similar to the segmentation process used in [16]. In each segmented region, we apply fuzzy logic to define the color, texture, and shape features, respectively. The motivation to incorporate fuzzy logic into the features is to address the typical feature representation impreciseness to improve the robustness and effectiveness of the indexing scheme such that a certain degree of variations of the feature values is allowed.

We use the color histogram as the color feature for a segmented region. Correspondingly, the fuzzy color histogram is defined as follows.

We assume that any color $c$ is a fuzzy set. As a result, for any color $c'$ of the color universe, function $\mu_c(c') : \mu \to [0,1]$ depicts the resemblance degree of the color $c'$ to the color $c$. Note that a good fuzzy resemblance function should admit that the resemblance degree decreases as the inter-color distance increases. Consequently, the natural choice is the Gaussian function defined as follows:

$$\mu_c(c') = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{d^2(c,c')}{2\sigma^2}\} \tag{1}$$

where $d$ is the Euclidean distance between color $c$ and $c'$ in the LAB color space, $\sigma$ is the average distance between any two colors

$$\sigma = \frac{2}{B(B-1)} \sum_{i=1}^{B-1} \sum_{k=i+1}^{B} d(c,c') \tag{2}$$

where $B$ is the number of bins in the color histogram. $\mu_c(c')$ for each bin pair is pre-computed and is implemented as a lookup table to reduce the online computation.

This fuzzy color model enables the typical impreciseness of a color feature to cover its neighboring colors in the color space. This means that each time a color $c$ is found in the image, it will influence all the colors according to their resemblance to the color $c$. The fuzzified color histogram is then expressed as:

$$h_2(c) = \sum_{c' \in U} h_1(c') \mu_c(c') \tag{3}$$

where $U$ is the color universe and $h_1(c')$ is the original color histogram. Note that $h_2(c)$ is in fact a linear convolution between $h_1(c')$ and $\mu_c(c')$.

A similar fuzzy resemblance function $\mu_i(f)$ is defined to represent the texture and shape features, *f*, of each region. The texture feature is represented using Gabor wavelets in 8 directions; the shape feature is represented using the first three ordered normalized inertia. Consequently, the fuzzy texture and fuzzy shape feature vectors of region $R_i$ are represented as

$$\hat{f}_i^T = \sum_{f^T \in U^T} f^T \mu_i(f^T) \tag{4}$$

$$\hat{f}_i^S = \sum_{f^S \in U^S} f^S \mu_i(f^S) \tag{5}$$

respectively, where $U^T$ ($U^S$) is the feature space composed of the texture (shape) feature vectors of all blocks (units of a region) in an image.

Based on the defined fuzzy features, the similarity between two regions $p$ and $q$ in the region feature space, $DIST(p,q)$, is defined as a metric function represented by the weighted sum of the L2 distances between the respective color,

texture, and shape features of the two regions. Suppose we have *M* regions in image 1 and *N* regions in image 2. Defining

$$R_{i\,\mathrm{Im}age2} = Min_{j=1}^{N}\{DIST(i,j)\} \quad (6)$$

$$R_{j\,\mathrm{Im}age1} = Min_{i=1}^{M}\{DIST(j,i)\} \quad (7)$$

The similarity between two images 1 and 2 is defined as

$$DI(1,2) = \frac{\sum_{i=1}^{M} w_{1i} R_{i\,\mathrm{Im}age2} + \sum_{j=1}^{N} w_{2j} R_{j\,\mathrm{Im}age1}}{2} \quad (8)$$

where $w_{1i}$ ($w_{2j}$) is the weight for region *i* (*j*) in image 1(2). The weight for each region is initially set to be the area ratio of the region to the corresponding image. These weights will be adaptively updated in the user relevance feedback.

This definition of the overall distance between two images is a balanced scheme in similarity measure between regional and global matching. As compared with many existing similarity measures in the literature, this definition strives to incorporate as much expressive and discriminating information as possible and, at the same time, achieves a relatively low computational complexity.

# 3. HIERARCHICAL INDEXING STRUCTURE AND "HEAR" ONLINE SEARCH

To achieve fast retrieval, we have designed a hierarchical indexing structure in the database and a related online search algorithm to avoid the linear search. An *optimal* indexing structure is defined in the region feature space as such that a query image only needs to be compared with those in the database that have at least one region that is most similar to a region in the query image.

Let *S* denote the set of all the nodes in the indexing structure, and *X* be the set of all the regions in the database. Each node $s \in S$ is a set of regions $X_s \subset X$ with a feature vector $z_s$, the centroid of the region feature set $F_s$ the node represents. The children of a node $s \in S$ are denoted as $c(s) \subset S$. The child nodes partition the region feature space of the parent node such that

$$X_s = \bigcup_{r \in c(s)} X_r \quad (9)$$

Now the question is how to construct such an optimal indexing structure. Recall that we used a modified *k*-means algorithm for image segmentation in the beginning to form all the regions. After all the images in the database are indexed based on the indexing scheme presented in Section 2, we apply the modified *k*-means algorithm [16] again to all the *feature vectors corresponding to every regions* of every images recursively to form the hierarchy of the indexing structure. Here the *k* for each level of the hierarchy is not fixed; it is determined by the modified *k*-means algorithm dynamically. All the nodes represent centroid feature vectors of the corresponding sets of regions except for the leaf nodes; a leaf node, on the other hand, represents a set of regions with each region pointing to a set of images which all share this region in the feature space. The depth of the indexing structure is determined adaptively based on the size of the image database (see the proof of Theorem 1 for details). The resulting indexing tree is called the *Hierarchical Indexing Structure.* Below we introduce an online query search algorithm developed for this indexing structure, and show the average time complexity of this algorithm.

Based on the Hierarchical Indexing Structure, the image set associated with a leaf node is *significantly* smaller than the original image set in the database. We show that this reduced image set may be further filtered in the online query search by exploiting the triangle inequality principle. Recall that the similarity function between two regions defined in Section 2 is metric. Given two regions *p* and *q* associated with two images in the image set of a leaf node in the indexing tree, we have:

$$DIST(p,q) \geq \left|DIST(p,z) - DIST(q,z)\right| \quad (10)$$

where *z denotes* a key region feature represented by the centroid of the corresponding leaf node (cluster). Consider a set of *I* regions $X_h = \{x_{h1}, x_{h2}, ..., x_{hI}\}$ at leaf node *h* and a key region feature $z_h$. Pre-calculating $DIST(x_{hi}, z_h)$, for *i*=1 to *I*, results in a linear table of *I* entries. In order to find those regions $x \in X_h$ at node *h* such that $DIST(r,x) \leq t$ for a query region *r* and the pre-defined threshold *t*, we note that the lower bounds on $DIST(r,x)$ exist by determining $DIST(r,z_h)$, $DIST(x,z_h)$ and repeatedly applying Eq. (10). If $\left|DIST(r,z_h) - DIST(x,z_h)\right| > t$, *x* can be safely eliminated from the linear table of *I* entries, resulting in avoiding search for *all* the entries in the table. Thus, given a query, we have the following retrieval algorithm called Hierarchical, Elimination-based A* Retrieval (HEAR), and the theorem guaranteeing the logarithm complexity in the average case performance for HEAR.

The symbols used in the HEAR algorithm are introduced as follows. *m* is the number of regions in the query image; $s^*$ is the cluster whose centroid has the minimum distance to a query region *r*; $\Omega$ is the cluster set we have searched; $\left|c(s^*)\right|$ is the size of the child set of $s^*$; $z_s$ is the cluster centroid; *NodesSearched* records the number of the nodes we have searched so far; and *t* is the pre-defined threshold of the distance between a region and a query region. $\Psi$ is the final image set to be compared with the query image.

**Algorithm 1. HEAR**

1. For each region *r* in the query image $(1 \leq r \leq m)$
2. $s^* = root$
3. $\Omega = \{s^*\}$
4. NodesSearched = 0
5. While $s^*$ is not a node of the desired tree depth
   (1) $\Omega \leftarrow (\Omega - \{s^*\}) \bigcup c(s^*)$
   (2) $NodesSearched = NodesSearched + \left|c(s^*)\right|$
   (3) $s^* \leftarrow \arg\min_{s \in \Omega}(DIST(r, z_s))$
6. $\Phi = \{\}$
7. for each region *p* in the node $s^*$
   (4) if $\left|DIST(p,z_S) - DIST(r,z_S)\right| \leq t$
   (5) $\Phi \leftarrow \Phi \bigcup \{p\}$
8. $\Psi_r$ = {Images having regions in set $\Phi$}

9. $\Psi = \bigcup_{r=1}^{m} \Psi_r$

Typical search algorithms in CBIR literature would traverse an indexing tree top-down, selecting a branch that minimizes the distance between a query *q* and a cluster centroid $z_s$. However, this search strategy is not optimal since it does not allow backtracking. To achieve an optimal search, we apply A* search algorithm [13] by keeping track of all nodes which have been searched and always selecting the nodes with a minimum distance to the query region. The A* search is guaranteed to select the cluster whose centroid has the minimum distance in the set of visited nodes to the query region. Hence, it is optimal.

**Theorem 1.** *In average case, FAST achieves the logarithm retrieval efficiency based on the Hierarchical Indexing Structure and the HEAR online query search algorithm.*

**Proof.** Suppose *m* is the average branching factor of the index tree; *n* is the number of images in the database; *l* is the average number of regions of an image; *k* is the depth of the indexing tree. Then *nl* is the total number of regions. In the average case, it is clear that when $k \geq \log_m nl - \log_m(\log_m nl)$, the number of regions in a leaf node $w \leq \log_m nl$. In the selected leaf node $s^*$, the triangle inequality principle in Eq. (10) is applied. Without loss of generality, assume that the maximum distance between a region in the region set of $s^*$ and the centroid of the region set of $s^*$ is *a*, i.e., *a* is the radius of the region set of $s^*$ in the feature space, and assume that applying the triangle inequality principle in Eq. (10) allows only those regions within the radius of $a/\tau$ compared, where $1/\tau$ is the threshold. Consequently, the average number of regions selected to compare with the query region is $q = w/\tau^2 \leq \frac{\log_m nl}{\tau^2}$. We call these regions candidate regions. Since each candidate region at most corresponds to one different image in the database, the total number of images in the database to be compared with the query image is at most $ql = \frac{l \log_m nl}{\tau^2} = \frac{l}{\tau^2}(\log_m n + \log_m l)$. Since *l* is a constant, and determined by the resolution of the image segmentation, $l << n$. We immediately have $ql = O(\log_m n)$, which means that the complexity of HEAR is $O(\log_m n)$ for a database of *n* images. □

While any feature based CBIR methods could apply a clustering algorithm recursively to generate a hierarchy in the (typically high dimensional) feature space, we argue that this does not work in general, and thus show that the contributions reflected in Theorem 1 are unique and significant.

We define that a classification based clustering in a feature space is *spherically separable* [15] if for any cluster there is always a specific radius *R* for this cluster such that for any feature vector *v, v* is in the cluster iff $D(v,c) < R$ where *c* is the centroid vector of the cluster and *D* is a metric distance measure. Given a CBIR method, if the similarity measure is metric, and if the images are indexed in global features (i.e., each image is indexed by a single feature vector in the *image feature space*), in order to generate a hierarchy in the feature space by recursively clustering the features of the whole image database, it would require that all the clusters be spherically separable. Clearly this is not true as in a typical image feature space, the distribution of the semantic classes in the feature space is rather complicated (e.g., it is typical that one semantic class is contained by another, or two semantic classes are completely "mixed" together), and thus the spherically separable property is by no means satisfied. This is shown to be a well-known fact even in many special domain image classification or clustering problems such as in face image classification, not speaking for the general domain image retrieval. On the other hand, if the similarity measure is not metric, it would not be possible to generate a hierarchy in a feature space based on the recursive applications of a clustering algorithm as the clustering presumes a metric distance measure. Consequently, the only possibility to generate such an indexing hierarchy is to use a non-global feature, i.e., to build up this hierarchy in a feature space other than the image feature space. The significance of the development of the Hierarchical Indexing Structure as well as the related HEAR online search algorithm reflected through Theorem 1 is that we have explicitly developed an indexing scheme in the *regional feature space* and we have shown that even with this "detour" through the regional feature space, we can still promise a logarithm search complexity in the average case in the image feature space.

We develop the Hierarchical Indexing Structure in the regional feature space based on the following three reasons. First, since the features we have developed to index images are based on the regional feature space, it is natural to build up an indexing hierarchy in the regional feature space. Second, the similarity measure defined in FAST is not metric, and thus it is not possible to directly build up an indexing hierarchy in the image feature space. Third, after segmentations of an image into regions, the features in the regional feature space are essentially "uniform", and consequently they are able to satisfy the spherically separable property in the regional feature space, which is required for the construction of the hierarchy using clustering.

Apart from the above discussions of the *Hierarchical Indexing Structure*, it is also interesting to compare it with the existing high-dimensional indexing structures, e. g., R-tree and its derivative indexing trees. It has been demonstrated that the search efficiency of an R-tree is largely determined by coverage and overlap [9]. Coverage of a level of an R-tree is the total area of all the rectangles associated with the nodes of that level. Overlap of a level of an R-tree is the total area contained within two or more nodes. Efficient R-tree search demands that both coverage and overlap be minimized. From the point of view of R-tree based multidimensional access structures, the proposed *Hierarchical Indexing Structure* has two advantages. First, the nodes in each level have no overlap since each region feature belongs to only one cluster (node). With this property, multiple-path traversal is avoided, which improves the search efficiency significantly. Second, the search on the *Hierarchical Indexing Structure* does not depend on the node coverage because no dimension comparisons are required to decide the branch in HEAR. In other words, the minimum bounding region (MBR) of each internal node in the *Hierarchical Indexing Structure* is determined by the region features *per se* and can be any possible shape. With the non-overlapping property between internal nodes of each level and MBR coverage (shape) independent search in HEAR, the

efficiency of the *Hierarchical Indexing Structure* is enhanced as compared with the discussed R-tree as well as its derivative data structures for region based CBIR.

# 4. RELEVANCE FEEDBACK USING "ITP" AND "ARWU"

To achieve semantics-tailored retrieval, we must address the *human perception subjectivity* issue in CBIR. Since the relevance subjectivity in FAST resides at the region level as opposed to at the image level, ideally we would like to ask users to indicate the relevant regions in each retrieval, which would add complexity in user interface and users' interaction. As a compromise, FAST assumes that users only cast yes (+) or no (-) vote to each retrieved image as the data collected in the user relevance feedback. Since based on this very "qualitative" user relevance feedback, the feedback data is typically sparse and limited. We have developed an algorithm to infer the user preference in order to tailor to the intended retrieval semantics from a sparse distribution of this "qualitative" data. Moreover, we take the advantage of this user relevance feedback information to further expedite the subsequent query search, resulting in achieving the two goals of fast and semantics-tailored retrieval simultaneously.

We note the fact that the similar (common) regions among relevant images are important to characterize relevant images, whereas the similar (common) regions among irrelevant images are important to distinguish the retrieved irrelevant images from the relevant images. Assuming that a user personal preference of the intended retrieval semantics is consistent over the whole session of the retrieval, we develop a user relevance feedback algorithm called Indexing Tree Pruning (ITP). The idea of ITP is that we use the k-means algorithm to infer the "typical" regions from the images voted as relevant, and the "typical" regions from the images voted as irrelevant, based on which a standard two-class support vector machine (SVM) [15] is used to generate a separation hyperplane in the region feature space, which in turn "cuts" the space into two halves; the subsequent search may be further constrained to focus on the relevant side of the hierarchical indexing structure using HEAR. The generated "half" hierarchical indexing tree is called a session tree. Specifically, the ITP algorithm follows.

**Algorithm 2. ITP**

1. Initialization, set $regR = \{\}$ ; $regI = \{\}$ ;

2. Applying the modified *k*-means clustering algorithm to the region subspace consisting of relevant images, *m* clusters are obtained. They are sorted in terms of the number of regions, denoted as $SR = \{R_1, R_2 .... R_m\}$, where $\|R_i\| \geq \|R_j\|$ for *i<j*

3. Applying the modified *k*-means clustering algorithm to the region subspace consisting of irrelevant images, *n* cluster are obtained. They are sorted in terms of the number of regions, denoted as $SI = \{I_1, I_2 .... I_n\}$, where $\|I_i\| \geq \|I_j\|$ for *i<j*

4. $regR \leftarrow regR \cup R_k$, k=1, 2, .., p; $regI \leftarrow regI \cup I_k$, k=1, 2, …, q where *p(q)* is a threshold ratio relative to *m(n)*; In the FAST prototype, they are set empirically.

5. A Gaussian RBF based two-class SVM is applied to the relevant and irrelevant region sets *regR* and *regI* to learn the hyperplane *H*, which separate the region space into relevancy and irrelevancy halves.

6. *Y={}; X={};*
   For the centroid of each leaf node in the indexing tree, $t_i$,
   Calculate $H(t_i)$
   If $H(t_i) \geq 0$, $Y \leftarrow Y \cup \{t_i\}$ Else, $X \leftarrow X \cup \{t_i\}$

7. Pruning the indexing tree, reserving only ancestor nodes of *Y* to generate a session tree *ST*

8. Performing online search algorithm HEAR on the session tree *ST*

The actual pruning is done through applying DBT algorithm [8]. Note that ITP differs from the existing SVM-based user relevance feedback algorithms, such as [3], which typically require a large number of voted samples to obtain a classifier in the feature space. In ITP, the SVM is not used directly to perform image retrieval; instead it is used to guide a coarse filtering (pruning) such that the hierarchical indexing structure in the database can be tuned in favor of the user's relevancy preference. In Section 5, we shall see that with a relatively small number of leaf nodes (<1000) and reasonable dimensionality of a feature space (9 in our indexing scheme), a relatively small number (15 ~ 30) of voted images can boost the performance well, and further expedite the subsequent retrieval at the same time.

While ITP is able to infer the relevancy and irrelevancy from the voted images, we make a further effort in attempting to infer the *degree* of the relevancy and irrelevancy. Following the "most similar, highest priority (MSHP)" principle [16], we adaptively update the region weights in Eq. (8) based on the feedback data. We implement this idea and call it the Adaptive Region Weight Updating (ARWU) algorithm.

The idea of ARWU is as follows. The cardinality of a cluster to which a query region belongs in the relevant region space is an indicator of the commonality of this region to the relevant image set. With the voted relevant and irrelevant images, for every region in the query image, the weights of the regions similar to the regions in the relevant images but dissimilar to the regions in the irrelevant images should increase, and otherwise the weights should decrease. For regions in each target image (image to be compared with the query image), a higher weight is given to regions with a smaller distance to the query image, and meantime the weight is adjusted to a higher value if it is the most similar to a query region with a high weight already. Otherwise, the weight of the region is lowered accordingly. The nature of this weight adjustment algorithm is a discriminant whiten transform learnt from both inferred relevant and irrelevant regions. In addition, the weights adjusted still preserve a desired characteristic for the distance metric, i. e., *the distance between the same images equals to 0*. ARWU is used along with ITP for further improving semantics-tailored retrieval. The ARWU algorithm follows:

**Algorithm 3. ARWU**

1. For each region in the query image, $Q_i$, *i=1…M*

   1.1 the cluster to which the region $Q_i$ belongs in the relevant image set is obtained as $C_i$, $C_i \in SR$

   1.2 the cluster to which the region $Q_i$ belongs in the irrelevant image set is obtained as $D_i$, $D_i \in SI$ ;

1.3 $WR_i = \frac{\|C_i\|}{\sum_k \|R_k\|}$, $WI_i = \frac{\|D_i\|}{\sum_k \|I_k\|}$;

1.4 update the weight of the region $W_{1i} = \frac{\eta_i \bullet WR_i}{\sum_{k=1}^{M} (\eta_k \bullet WR_k)}$,

where $\eta_i = 1 - WI_i$;

2. For each region in a target image, $T_j$, $j=1...N$, its most similar region in the query image is denoted as $S_j$

2.1 $U_j = \frac{W_{1S_j}}{R_{j,\mathrm{Image1}}}$, where $R_{j,\mathrm{Image1}}$ is the distance between this region, $T_j$, and the query image, which is defined in Eq. (7)

2.2 The weight for each region $T_j$ is normalized as $W_{2j} = \frac{U_j}{\sum_{k=1}^{N} U_k}$;

3. Applying $W_{1i}$ and $W_{2j}$ to Eq. (8) to determine the distance between the query image and each target image.

In the algorithm, $\eta_i$ acts as a penalty, reflecting the effect of negative examples to each region in the query image.

# 5. EXPERIMENTAL EVALUATIONS

We have implemented the FAST methodology in a prototype system which is also called FAST on a platform of Pentium III 800 MHZ CPU and 256M memory. The following reported evaluations are performed in a general-purpose color image database containing 10,000 images from the COREL collection of 96 semantic categories. Each semantic category has 85-120 images. From the COREL collection 1,500 images were randomly selected from all categories as the query set. A retrieved image is considered as a match in the evaluation if it belongs to the same category of the query image.

We have indexed the 10,000 images using the FAST hierarchical indexing structure described in Section 3; it takes about 2 second for indexing each image based on the FAST indexing scheme. The average query time for returning top 30 images is less than 1 second.

FAST is evaluated against one of the state-of-the-art CBIR systems, UFM [4], on the effectiveness comparison. Considering that UFM is a linear search method, to test the effects of the hierarchical indexing structure and HEAR algorithm, we have ported FAST into two separate versions: one with the original FAST methodology (which uses the hierarchical indexing structure in the region feature space and the HEAR online search), called WIS, and the other with the hierarchical indexing structure and HEAR disabled, i.e., using the FAST indexing scheme in a linear search, called NIS. Both versions of FAST are compared with UFM in the 10,000 image database. The precision-scope data is recorded in Figure 1, which demonstrates that the retrieval effectiveness of FAST is in general superior to that of UFM, and the application of the hierarchical index structure and HEAR does not degrade the performance perceivably; in fact, the performance of NIS is always about the same as that of WIS.

The hierarchical indexing structure of FAST is generated as a tree with the depth of 4 and the maximum branching factor of 5. The tree is almost balanced, which verifies that the clustering in the region feature space based on FAST methodology is almost spherically separable [15]. In FAST, each node of the tree is implemented as an object serialized to a physical file in the disk. Thus, the parallel search of the indexing tree is made possible. To study the scalability of FAST, we incrementally sample the original 10,000 image database to generate two smaller databases, one with 3,000 images and the other with 6,000 images. These two databases contain sampled images from all the 96 categories. Consequently, the depths of the hierarchical indexing trees for the two databases are set to be $\lfloor \log_5 nl - 3 \rfloor$ accordingly based on the same protocol used for the original image database, resulting in 2 and 3, respectively. For each of the three databases, we randomly sample 100 images as the query set from the corresponding database for this evaluation. We have recorded the average numbers of images compared in each of the three databases using FAST hierarchical indexing structure and HEAR. The result is documented in Table 1. With the increase of the database size, the percentage of the images examined decreases accordingly, approximately confirming the logarithm complexity. This result, combined with the results observed in Fig. 1, indicates the promise of FAST for efficiently handling large image databases without sacrificing retrieval effectiveness.

**Table 1. Retrieval efficiency and scalability results**

| Database Size | Average # of compared images | Average percentage of images examined |
|---|---|---|
| 3,000 | 795 | 26.5% |
| 6,000 | 1032 | 17.2% |
| 10,000 | 1140 | 11.4% |

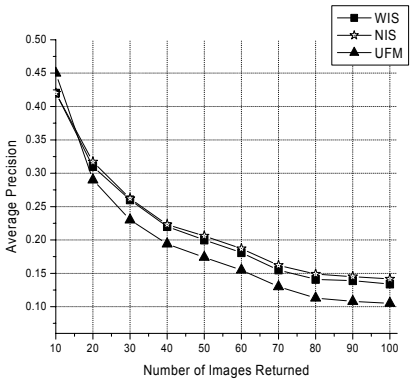


**Figure 1. Average precision/scope comparisons between two versions of FAST (WIS and NIS), and UFM**

In FAST indexing scheme we use a Gaussian function to correlate color descriptions and to smooth the regions so that the color perception uncertainty and segmentation inaccuracy issue are addressed explicitly. To evaluate the effectiveness of the indexing scheme to improve the robustness to color variations and segmentation-related uncertainties, we compare the performance of FAST and UFM w.r.t. the uncertainties of color variations and image segmentation. The uncertainty of color variations is simulated by changing colors to their

adjacent values, and the uncertainty of segmentation is characterized by the entropy. For image *i* with *C* segmented regions, the entropy, *E(i)*, is defined as

$$E(i) = -\sum_{j=1}^{C} P(R_j^i) \log[P(R_j^i)] \quad (11)$$

where $P(R_j^i)$ is the percentage of image *i* covered by region $R_j^i$. The larger the value of the entropy, the higher the uncertainty is. Clearly the entropy *E(i)* increases as the number of regions *C* increases. Thus, the uncertainty varies with changing the value of *C*. Different values of *C* are obtained by modifying the stop criteria of the modified *k*-means algorithm. To give a fair comparison between the FAST indexing scheme (the HEAR online search and relevance feedback mechanisms are disabled) and UFM for different color variations, we perform the same experiments for different degree of color changes and average values of *C* on the 3,000 image database. Specifically, we apply color changes to an image (target image) in the database. The modified image is then used as the query image, and the rank of the retrieved target image is recorded. Repeating the process for all the images in the database, the average ranks for the entire target images are computed for FAST and UFM, respectively, as shown in Figure 2, which indicates that the average rank of the target images of FAST for each variation is always lower than that of UFM. To evaluate the robustness to the segmentation uncertainty, the performances in terms of the overall average precisions in the returned top 30 images are evaluated for both techniques, as shown in Figure 3, which indicates that for every average number of regions, FAST performs better than or the same as UFM does. This evaluation demonstrates that FAST is more stable and robust than UFM to the uncertainties of the potential color variations and the segmentation.

In FAST with the user relevance feedback mode, the Gaussian kernel function used in SVM is $K(x, y) = e^{-\|x-y\|^2/2\sigma^2}$ with $\sigma = \sqrt{2}/2$. In order to evaluate the capability of ITP and ARWU, FAST is run on the 10,000 image database with the user relevance feedback mode for the 1,500 query-image set with varied numbers of retrieved images. Different users were invited to run FAST initially without relevance feedback interaction, and then to place their individual relevance feedbacks. Finally, the individualized retrieval precisions are averaged to plot the curves of the retrieval precision vs. the number of the returned images. Figure 4 shows the average precision in three and five rounds of the feedbacks, respectively. The semantic-tailored capability empowered by ITP and ARWU clearly enhances the retrieval effectiveness and the semantics-tailored retrieval in all the scenarios.

Another test is performed to verify the promise of FAST ITP itself on the basis of the three 100 image query sets for the three image databases we have constructed earlier on. We have recorded the average number of images compared in the initial retrieval and after the 3rd and 5th relevance feedback iterations, respectively, in the databases of the three different sizes. The results are shown in Fig. 5.

The reduction of the number of images compared in each relevance feedback iteration due to tree pruning in ITP is observed. The efficiency boost between the 3rd iteration and the initial retrieval is significant due to the relatively large portion of leaf nodes pruned in the first three relevance feedback iterations. With the progress of the subsequent relevance feedback iterations, the ratio of the nodes classified to the irrelevancy side of the SVM decreases quickly such that the efficiency boost decays accordingly. Since in FAST the non-leaf nodes of the hierarchical indexing structure are stored in main memory the I/O costs for each query are proportional to the number of images compared. The reduced I/O costs in the FAST are observed as shown in Figure 5.

To evaluate the effects of FAST ARWU for semantic similarity measurement, we compared the average precision for the same 1,500 query image set on the 10,000 image database, in the scenarios with and without running ARWU. The experiment is based on the top 30 returned images for each query. The results are shown in Figure 6. As shown, the retrieval effectiveness is improved substantially with ARWU.

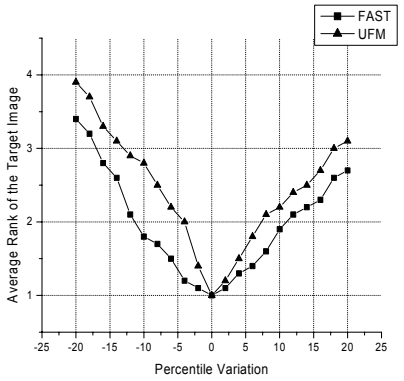


**Figure 2. Comparing FAST with UFM on the robustness to color variation uncertainty. Every image in the 3,000 image database is changed in color and used as a query image.**

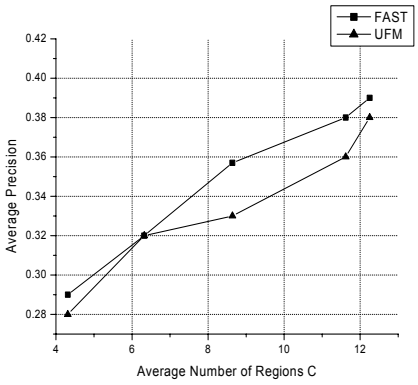


**Figure 3. Comparing FAST with UFM on the robustness to the image segmentation uncertainties.**

# 6. CONCLUSIONS

We have developed a new CBIR methodology based on the goal of delivering Fast And Semantics Tailored retrieval, and thus we call it FAST. FAST incorporates a new indexing scheme, a hierarchical indexing structure with a hierarchical, elimination-based A* retrieval online search algorithm called HEAR. We have shown that HEAR offers the significant strength to guarantee a logarithm search instead of a linear

search in the average case. FAST also offers the user relevance feedback capability to not only address the individualized retrieval based on the user intended semantics, but also take advantage of the hierarchical indexing tree and HEAR to achieve faster and more semantics-tailored retrieval through applying the ITP and ARWU algorithms. The promise FAST demonstrates and the benefits FAST offers are sufficiently supported by the extensive experimental evaluations.

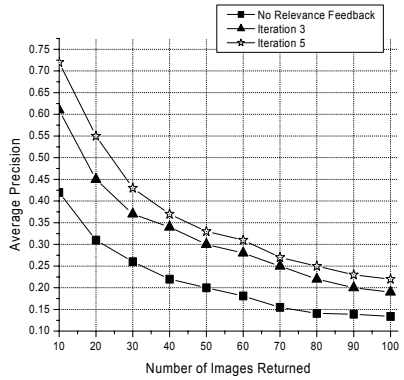


**Figure 4. Average precision vs. the number of retuned images with three and five rounds of user relevance feedback**

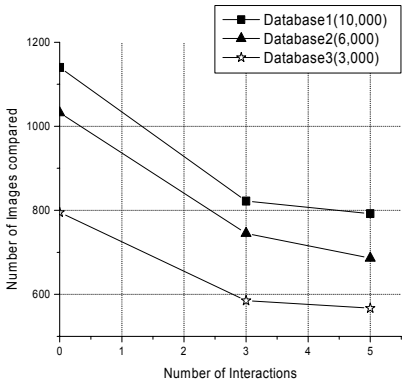


**Figure 5. The effect of ITP in iterations for the three databases**

# 7. REFERENCES


[1] C. Carson et al., "Blobworld: a system for region-based image indexing and retrieval", Proc. of the 3rd Int'l Conf. on Vis. Info. Sys., Amsterdam, Netherlands, Jun. 1999, pp. 509-516.

[2] V. Castelli et al., "CSVD: Clustering and singular value decomposition for approximate similarity search in high-dimensional spaces", IEEE T-KDE, 15(3):671-685, 2003

[3] Olivier Chapelle, Patrick Haffner, and Vladimir N. Vapnik, "Support vector machines for histogram-based image classification", IEEE Trans. Neural Networks, Vol. 10, No. 5, Sep. 1999

[4] Yixin Chen, James Z. Wang, "A region-based fuzzy feature matching approach to content-based image retrieval", IEEE T-PAMI, vol. 24, no. 9, 2002, pp. 1252-1267


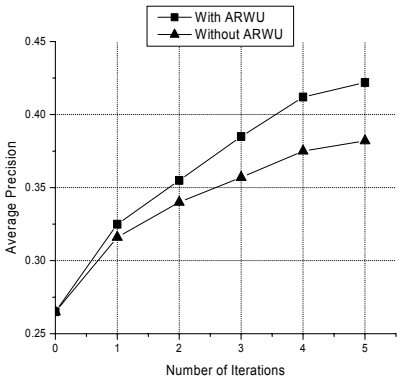


**Figure 6. Accuracy comparison when FAST ITP is run with and without ARWU algorithm**


[5] M. Flickner et al., "Query by image and video content: The QBIC system", IEEE Computer, 28(9):23-32, Sep. 1995

[6] Jing Huang, S. Ravi Kumar et al, "Image indexing using color correlograms", Proc. of the IEEE CVPR, Puerto Rico, Jun. 1997.

[7] Feng Jing, et al., "An efficient region-based image retrieval framework", Proc. ACMMM, Juan-les-Pins, France, Dec., 2002

[8] Y. Kearns, M. J. Mansour, "A fast, bottom-up decision tree pruning algorithm with near-optimal generalization", Proc. of the 15th Int'l Conf. on Machine Learning, Madison, WI, pp. 269-277

[9] Guojun Lu, "Techniques and data structures for efficient multimedia retrieval based on similarity", IEEE Trans. on Multimedia, Vol. 4, No. 3, Sep. 2002, pp. 372-384

[10] W.Y. Ma, B. Manjunath, "NeTra: a toolbox for navigating large image databases", Proc. IEEE Int'l Conf. Image Processing, Santa Barbara, CA, Oct. 1997, pp. 568-571.

[11] Virginia Ogle, Michael Stonebraker, "Chabot: Retrieval from a relational database of images", IEEE Computer, 28(9), Sep. 1995

[12] K. Porkaew, K. Chakrabarti, and S. Mehrotra, "Query refinement for multimedia similarity retrieval in MARS", Proc. of ACMMM, November, 1999

[13] Stuart Russell and Peter Norvig, Artificial Intelligence – A Mordern Approach, Prentice Hall, 1995

[14] S. Tong and E. Chang, "Support vector machine active learning for image retrieval", Proc. of ACM Multimedia, Ottawa, Canada, Sep. 2001

[15] V. Vapnik, The nature of Statistical Learning Theory. Springer-Verlag, New York, 1995

[16] James Z. Wang, Jia Li, Gio Wiederhold, "SIMPLIcity: semantics-sensitive integrated matching for picture libraries", IEEE Trans. PAMI, Vol. 23, No.9, Sep. 2001

[17] Ruofei Zhang, Zhongfei Zhang, "A Clustering Based Approach to Efficient Image Retrieval", Proc. of the 14th IEEE Int'l Conf. on Tool with Artificial Intelligence, Washington DC, USA, Nov., 2002