# Region-based Image Retrieval using an Object Ontology and Relevance Feedback

Vasileios Mezaris, Ioannis Kompatsiaris, and Michael G. Strintzis

*Abstract*— **In this paper, an image retrieval methodology suited for search in large collections of heterogeneous images is presented. The proposed approach employs a fully unsupervised segmentation algorithm to divide images into regions and endow the indexing and retrieval system with content-based functionalities. Low-level descriptors for the color, position, size and shape of each region are subsequently extracted. These arithmetic descriptors are automatically associated with appropriate qualitative intermediate-level descriptors, which form a simple vocabulary termed *object ontology*. The object ontology is used to allow the qualitative definition of the high-level concepts the user queries for (*semantic objects*, each represented by a *keyword*) and their relations in a human-centered fashion. When querying for a specific semantic object (or objects), the intermediate-level descriptor values associated with both the semantic object and all image regions in the collection are initially compared, resulting in the rejection of most image regions as irrelevant. Following that, a relevance feedback mechanism, based on support vector machines and using the low-level descriptors, is invoked to rank the remaining, potentially relevant image regions and produce the final query results. Experimental results and comparisons demonstrate in practice the effectiveness of our approach.**

**Keywords** : *image retrieval; image databases; image segmentation; ontology; relevance feedback; support vector machines; digital libraries*

## I. Introduction

In recent years, the accelerated growth of digital media collections and in particular still image collections, both proprietary and on the Web, has established the need for the development of human-centered tools for the efficient access and retrieval of visual information. As the amount of information available in the form of still images continuously increases, the necessity of efficient methods for the retrieval of the visual information becomes evident [1]. Additionally, the continuously increasing number of people with access to such image collections further dictates that more emphasis be put on attributes such as the user-friendliness and flexibility of any image retrieval scheme. These facts, along with the diversity of available image collections, varying from *restricted*, e.g. medical image databases and satellite photo collections, to general purpose collections, which contain heterogeneous images, and the diversity of requirements regarding the amount of knowledge about the images that should be used for indexing, have led to the development of a wide range of solutions [2].

The very first attempts for image retrieval were based on exploiting existing image captions to classify images to predetermined classes or to create a restricted vocabulary [3]. Although relatively simple and computationally efficient, this approach has several restrictions mainly deriving from the use of a restricted vocabulary that neither allows for unanticipated queries nor can be extended without reevaluating the possible connection between each image in the database and each new addition to the vocabulary. Additionally, such keyword-based approaches assume either the preexistence of textual image annotations (e.g. captions) or that annotation using the predetermined vocabulary is performed manually. In the latter case, inconsistency of the keyword assignments among different indexers can also hamper performance. Recently, a methodology for computer-assisted annotation of image collections was presented [4].

To overcome the limitations of the keyword-based approach, the use of the image visual contents has been proposed. This category of approaches utilizes the visual contents by extracting low-level indexing features for each image or image segment (region). Then, relevant images are retrieved by comparing the low-level features of each item in the database with those of a user-supplied sketch [5] or, more often, a key-image that is either selected from a restricted image set or is supplied by the user (*query-by-example*). One of the first attempts to realize this scheme is the *Query by Image Content* system [6], [7]. Newer contributions to query-by-example (QbE) include systems such as *Netra* [8], [9], *Mars* [10], *Photobook* [11], *VisualSEEK* [12] and *Istorama* [13]. They all employ the general framework of query-by-example, demonstrating the use of various indexing feature-sets either in the *image* or in the *region* domain.

A recent addition to this group, Berkeley's *Blobworld* [14], [15], proposes segmentation using the Expectation-Maximization algorithm and clearly demonstrates the improvement in query results attained by querying using region-based indexing features rather than global image properties, under the query-by-example scheme. Other works on segmentation, that can be of use in content-based retrieval, include segmentation by Anisotropic Diffusion [16], the RSST algorithm [17], the Watershed transformation [18], the Normalized Cut [19], and the Mean Shift approach [20]. While such segmentation algorithms can endow an indexing and retrieval system with exten-

V. Mezaris and M.G. Strintzis are with the Information Processing Laboratory, Electrical and Computer Engineering Dept., Aristotle University of Thessaloniki, Thessaloniki 54124, Greece, and with the Informatics and Telematics Institute, Centre for Research and Technology Hellas, 1st Km Thermi-Panorama road, Thessaloniki 57001, Greece. I. Kompatsiaris is with the Informatics and Telematics Institute, Centre for Research and Technology Hellas, 1st Km Thermi-Panorama road, Thessaloniki 57001, Greece. This work was supported by the EU project SCHEMA "Network of Excellence in Content-Based Semantic Scene Analysis and Information Retrieval" (IST-2001-32795). The assistance of COST211 quat is also gratefully acknowledged.

sive content-based functionalities, these are limited by the main drawback of query-by-example approaches: the need for the availability of an appropriate key-image in order to start a query. Occasionally, satisfying this condition is not feasible, particularly for image classes that are under-represented in the database.

Hybrid methods exploiting both keywords and the image visual contents have also been proposed [21], [22], [23]. In [21], the use of *probabilistic multimedia objects* (*Multijects*) is proposed; these are built using Hidden Markov Models and necessary training data. Significant work was recently presented on unifying keywords and visual contents in image retrieval. The method of [23] performs semantic grouping of keywords based on user relevance feedback to effectively address issues such as word similarity and allow for more efficient queries; nevertheless, it still relies on preexisting or manually added textual annotations. In well-structured specific domain applications (e.g sports and news broadcasting) domain-specific features that facilitate the modelling of higher level semantics can be extracted [24], [25]. A priori knowledge representation models are used as a knowledge base that assists semantic-based classification and clustering. In [26], *semantic entities*, in the context of the MPEG-7 standard, are used for knowledge-assisted video analysis and object detection, thus allowing for semantic-level indexing. However, the need for accurate definition of semantic entities using low-level features restricts this kind of approaches to domain-specific applications and prohibits non-experts from defining new semantic entities.

This paper attempts to address the problem of retrieval in generic image collections, where no possibility of structuring a domain-specific knowledge base exists, without imposing restrictions such as the availability of key-images or image captions. The adopted region-based approach employs still-image segmentation tools that enable the time-efficient and unsupervised analysis of still images to regions, thus allowing the "content-based" access and manipulation of visual data via the extraction of low-level indexing features for each region. To take further advantage of the human-friendly aspects of the region-based approach, the low-level indexing features for the spatial regions can be associated with higher-level concepts that humans are more familiar with. This is achieved with the use of an *ontology* and a relevance feedback mechanism [27], [28]. Ontologies [29], [30], [31] define a formal language for the structuring and storage of the high-level features, facilitate the mapping of low-level to high-level features and allow the definition of relationships between pieces of multimedia information; their potential applications range from text retrieval [32] to facial expression recognition [33]. The resulting image indexing and retrieval scheme provides flexibility in defining the desired semantic object/keyword and bridges the gap between keyword-based approaches and query-by-example approaches (figure 1).

The paper is organized as follows: The employed image segmentation algorithm is presented in section II. Section III presents in detail the components of the retrieval scheme. Section IV contains an experimental evaluation and comparisons of the developed methods, and finally, conclusions are drawn in section V.

## II. Color Image Segmentation

### A. Segmentation Algorithm Overview

A region-based approach to image retrieval has been adopted; thus, the process of inserting an image into the database starts by applying a color image segmentation algorithm to it, so as to break it down to a number of regions. The segmentation algorithm employed for the analysis of images to regions is based on a variant of the K-Means-with-connectivity-constraint algorithm (KMCC), a member of the popular K-Means family [34]. The KMCC algorithm classifies the pixels into regions $s_k, k = 1, \ldots, K$ taking into account not only the intensity of each pixel but also its position, thus producing connected regions rather than sets of chromatically similar pixels. In the past, KMCC has been successfully used for model-based image sequence coding [35] and content-based watermarking [36]. The variant used for the purpose of still image segmentation [37] additionally uses texture features in combination with the intensity and position features.

The overall segmentation algorithm consists of the following stages:

- Stage 1. Extraction of the intensity and texture feature vectors corresponding to each pixel. These will be used along with the spatial features in the following stages.
- Stage 2. Estimation of the initial number of regions and their spatial, intensity and texture centers, using an initial clustering procedure. These values are to be used by the KMCC algorithm.
- Stage 3. Conditional filtering using a moving average filter.
- Stage 4. Final classification of the pixels, using the KMCC algorithm.

The result of the application of the segmentation algorithm to a color image is a segmentation mask $M$, i.e. a grayscale image comprising the spatial regions formed by the segmentation algorithm, $M = \{s_1, s_2, \ldots, s_K\}$, in which different gray values $1, 2, \ldots, K$ correspond to different regions: $M(\mathbf{p} \in s_k) = k$, where $\mathbf{p} = [p_x \ p_y]^T$, $p_x = 1, \ldots, x_{max}$, $p_y = 1, \ldots, y_{max}$ are the image pixels and $x_{max}$, $y_{max}$ are the image dimensions. This mask is used for extracting the region low-level indexing features described in section III-A.

### B. Color and Texture Features

For every pixel $\mathbf{p}$ a color feature vector and a texture feature vector are calculated. The three intensity components of the CIE L*a*b* color space are used as intensity features, $\mathbf{I}(\mathbf{p}) = [I_L(\mathbf{p}) \ I_a(\mathbf{p}) \ I_b(\mathbf{p})]^T$, since it has been shown that L*a*b* is more suitable for segmentation than the widely used RGB color space, due to its being approximately perceptually uniform [38].

In order to detect and characterize texture properties in the neighborhood of each pixel, the Discrete Wavelet Frames (DWF) [39] decomposition of two levels is used.

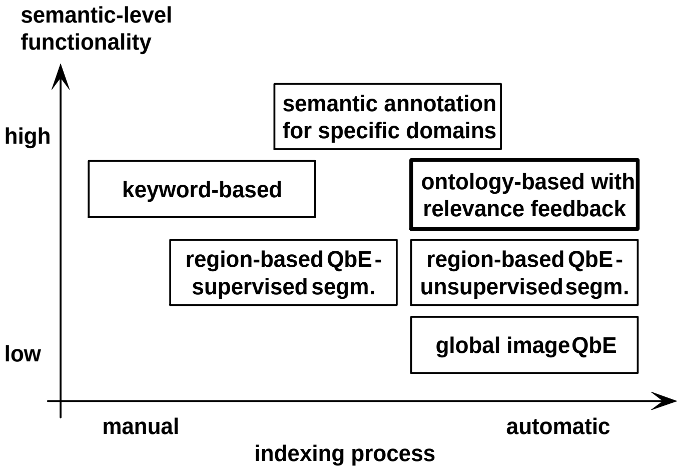

semantic-level functionality
high
low
semantic annotation for specific domains
keyword-based
ontology-based with relevance feedback
region-based QbE - supervised segm.
region-based QbE - unsupervised segm.
global image QbE
manual
automatic
indexing process

Fig. 1. Overview of image retrieval techniques. Techniques exploiting pre-existing textual information (e.g. captions) associated with the images would lie in the same location on the diagram as the proposed approach, but are limited to applications where such a priori knowledge is available.

The employed filter bank used is based on the lowpass Haar filter $H(z) = \frac{1}{2}(1 + z^{-1})$, which satisfies the lowpass condition $H(z)|_{z=1} = 1$. The complementary highpass filter $G(z)$ is defined by $G(z) = zH(-z^{-1})$. The filters of the filter bank are then generated by the prototypes $H(z)$, $G(z)$, as described in [39]. Despite its simplicity, the above filter bank has been demonstrated to perform surprisingly well for texture segmentation, while featuring reduced computational complexity. The texture feature vector $\mathbf{T}(\mathbf{p})$ is then made of the standard deviations of all detail components, calculated in a square neighborhood $\Phi$ of pixel $\mathbf{p}$.

### C. Initial Clustering

An initial estimation of the number of regions in the image and their spatial, intensity and texture centers is required for the initialization of the KMCC algorithm. In order to compute these initial values, the image is broken down to square, non-overlapping blocks of dimension $f \times f$. In this way, a reduced image composed of a total of $L$ blocks, $b_l, l = 1, \ldots, L$, is created. A color feature vector $\mathbf{I}^b(b_l) = [I^b_L(b_l)\ I^b_a(b_l)\ I^b_b(b_l)]^T$ and a texture feature vector $\mathbf{T}^b(b_l)$ are then assigned to each block; their values are estimated as the averages of the corresponding features for all pixels belonging to the block. The distance between two blocks is defined as follows:

$$D^b(b_l, b_n) = \|\mathbf{I}^b(b_l) - \mathbf{I}^b(b_n)\| + \lambda_1\|\mathbf{T}^b(b_l) - \mathbf{T}^b(b_n)\|, \quad (1)$$

where $\|\mathbf{I}^b(b_l)-\mathbf{I}^b(b_n)\|$, $\|\mathbf{T}^b(b_l)-\mathbf{T}^b(b_n)\|$ are the Euclidean distances between the block feature vectors. In our experiments, $\lambda_1 = 1$, since experimentation showed that using a different weight $\lambda_1$ for the texture difference would result in erroneous segmentation of textured images if $\lambda_1 \ll 1$, respectively non-textured images if $\lambda_1 \gg 1$. As shown in the experimental results section, the value $\lambda_1 = 1$ is appropriate for a variety of textured and non-textured images.

The number of regions of the image is initially estimated by applying a variant of the maximin algorithm [40] to this set of blocks. The distance $C$ between the first two centers identified by the maximin algorithm is indicative of the intensity and texture contrast of the particular image. Subsequently, a simple K-Means algorithm is applied to the set of blocks, using the information produced by the maximin algorithm for its initialization. Upon convergence, a recursive four-connectivity component labelling algorithm [41] is applied, so that a total of $K'$ connected regions $s_k$, $k = 1, \ldots, K'$ are identified. Their intensity, texture and spatial centers, $\mathbf{I}^s(s_k)$, $\mathbf{T}^s(s_k)$ and $\mathbf{S}(s_k) = [S_x(s_k)\ S_y(s_k)]^T$, $k = 1, \ldots, K'$, are calculated as follows:

$$\mathbf{I}^s(s_k) = \frac{1}{A_k}\sum_{\mathbf{p}\in s_k}\mathbf{I}(\mathbf{p}), \quad \mathbf{T}^s(s_k) = \frac{1}{A_k}\sum_{\mathbf{p}\in s_k}\mathbf{T}(\mathbf{p}), \\ \mathbf{S}(s_k) = \frac{1}{A_k}\sum_{\mathbf{p}\in s_k}\mathbf{p}, \quad (2)$$

where $A_k$ is the number of pixels belonging to region $s_k$: $s_k = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{A_k}\}$.

### D. Conditional Filtering

Images may contain parts in which intensity fluctuations are particularly pronounced, even when all pixels in these parts of the image belong to a single object (Fig. 2). In order to facilitate the grouping of all these pixels in a single region based on their texture similarity, a moving average filter is employed. The decision of whether the filter should be applied to a particular pixel $\mathbf{p}$ or not is made by evaluating the norm of the texture feature vector $\mathbf{T}(\mathbf{p})$ (Sec. II-B); the filter is not applied if that norm is below a threshold $\tau$. The output of the conditional filtering module can thus be expressed as:

$$\mathbf{J}(\mathbf{p}) = \begin{cases} \mathbf{I}(\mathbf{p}) & \text{if } \|\mathbf{T}(\mathbf{p})\| < \tau \\ \frac{1}{f^2}\sum \mathbf{I}(\mathbf{p}) & \text{if } \|\mathbf{T}(\mathbf{p})\| \geq \tau \end{cases} \quad (3)$$

Correspondingly, region intensity centers calculated similarly to Eq. (2) using the filtered intensities $\mathbf{J}(\mathbf{p})$ instead of $\mathbf{I}(\mathbf{p})$ are symbolized $\mathbf{J}^s(s_k)$.

An appropriate value of threshold $\tau$ was experimentally found to be

$$\tau = \max\{0.65 \cdot T_{max}, 14\} \tag{4}$$

where $T_{max}$ is the maximum value of the norm $\|\mathbf{T}(\mathbf{p})\|$ in the image. The term $0.65 \cdot T_{max}$ in the threshold definition serves to prevent the filter from being applied outside the borders of textured objects, so that their boundaries are not corrupted. The constant bound 14, on the other hand, is used to prevent the filtering of images composed of chromatically uniform objects; in such images, the value of $T_{max}$ is expected to be relatively small and would correspond to pixels on edges between objects, where filtering is obviously undesirable.

### *E. The K-Means with Connectivity Constraint Algorithm*

The KMCC algorithm applied to the pixels of the image consists of the following steps:

- Step 1. The region number and the region centers are initialized, using the output of the initial clustering procedure described in Sec. II-C.
- Step 2. For every pixel $\mathbf{p}$, the distance between $\mathbf{p}$ and all region centers is calculated. The pixel is then assigned to the region for which the distance is minimized. A generalized distance of a pixel $\mathbf{p}$ from a region $s_k$ is defined as follows:

$$\begin{aligned} D(\mathbf{p}, s_k) = \|\mathbf{J}(\mathbf{p}) - \mathbf{J}^s(s_k)\| + \lambda_1 \|\mathbf{T}(\mathbf{p}) - \mathbf{T}^s(s_k)\| \\ + \lambda_2 \frac{\bar{A}}{A_k} \|\mathbf{p} - \mathbf{S}(s_k)\|, \end{aligned} \tag{5}$$

where $\|\mathbf{J}(\mathbf{p}) - \mathbf{J}^s(s_k)\|$, $\|\mathbf{T}(\mathbf{p}) - \mathbf{T}^s(s_k)\|$ and $\|\mathbf{p} - \mathbf{S}(s_k)\|$ are the Euclidean distances between the pixel feature vectors and the corresponding region centers; pixel number $A_k$ of region $s_k$ is a measure of the area of region $s_k$, and $\bar{A}$ is the average area of all regions, $\bar{A} = \frac{1}{K}\sum_{k=1}^{K} A_k$. The regularization parameter $\lambda_2$ is defined as $\lambda_2 = 0.4 \cdot \frac{C}{\sqrt{x_{max}^2 + y_{max}^2}}$, while the choice of the parameter $\lambda_1$ has been discussed in Sec. II-C.

In (5), normalization of the spatial distance, $\|\mathbf{p} - \mathbf{S}(s_k)\|$ by division by the area of each region $\frac{A_k}{\bar{A}}$, is necessary in order to encourage the creation of large connected regions; otherwise, pixels would tend to be assigned to smaller rather than larger regions due to greater spatial proximity to their centers. The regularization parameter $\lambda_2$ is used to ensure that a pixel is assigned to a region primarily due to their similarity in intensity and texture characteristics, even in low-contrast images, where intensity and texture differences are small compared to spatial distances.

- Step 3. The connectivity of the formed regions is evaluated; those which are not connected are broken down to the minimum number of connected regions using a recursive four-connectivity component labelling algorithm [41].
- Step 4. Region centers are recalculated (Eq. (2)). Regions whose area size lies below a threshold $\xi$ are dropped. In our experiments, the threshold $\xi$ was equal to 0.5% of the total image area. The number of regions $K$ is then recalculated, taking into account only the remaining regions.
- Step 5. Two regions are merged if they are neighbors and if their intensity and texture distance is not greater than an appropriate merging threshold:

$$\begin{gathered} D^s(s_{k_1}, s_{k_2}) = \\ \|\mathbf{J}^s(s_{k_1}) - \mathbf{J}^s(s_{k_2})\| + \lambda_1 \|\mathbf{T}^s(s_{k_1}) - \mathbf{T}^s(s_{k_2})\| \leq \mu \end{gathered} \tag{6}$$

Threshold $\mu$ is image-specific, defined in our experiments by

$$\mu = \begin{cases} 7.5 & \text{if } C < 25 \\ 15 & \text{if } C > 75 \\ 10 & \text{otherwise} \end{cases} \tag{7}$$

where $C$ is an approximation of the intensity and texture contrast of the particular image, as defined in Sec. II-C

- Step 6. Region number $K$ and region centers are reevaluated.
- Step 7. If the region number $K$ is equal to the one calculated in *Step 6* of the previous iteration and the difference between the new centers and those in *Step 6* of the previous iteration is below the corresponding threshold for all centers, then stop, else goto *Step 2*. If index "old" characterizes the region number and region centers calculated in *Step 6* of the previous iteration, the convergence condition can be expressed as $K = K^{old}$ and

$$\begin{gathered} \|\mathbf{J}^s(s_k) - \mathbf{J}^s(s_k^{old})\| \leq c_I, \\ \|\mathbf{T}^s(s_k) - \mathbf{T}^s(s_k^{old})\| \leq c_T, \\ \|\mathbf{S}(s_k) - \mathbf{S}(s_k^{old})\| \leq c_S, \end{gathered}$$

for $k = 1, \ldots, K$. Since there is no certainty that the KMCC algorithm will converge for any given image, the maximum allowed number of iterations was chosen to be 20; if this is exceeded, the method proceeds as though the KMCC algorithm had converged.

## III. Region-based Retrieval Scheme

### *A. Low-level Indexing Descriptors*

As soon as the segmentation mask is produced, a set of descriptors that will be useful in querying the database are calculated for each region. These region descriptors compactly characterize each region's color, position and shape. All descriptors are normalized so as to range from 0 to 1.

The color and position descriptors of a region are the normalized intensity and spatial centers of the region. In particular, the **color descriptors** of region $s_k$, $F_1, F_2, F_3$, corresponding to the L, a, b components, are defined as follows:

$$F_1 = \frac{1}{100 \cdot A_k} \sum_{\mathbf{p} \in s_k} I_L(\mathbf{p}),$$

$$F_2 = \frac{\frac{1}{A_k}\sum_{\mathbf{p} \in s_k} I_a(\mathbf{p}) + 80}{200}, \quad F_3 = \frac{\frac{1}{A_k}\sum_{\mathbf{p} \in s_k} I_b(\mathbf{p}) + 80}{200}$$

where $A_k$ is the number of pixels belonging to region $s_k$. Similarly, the **position descriptors** $F_4, F_5$ are defined as:

$$F_4 = \frac{1}{A_k \cdot x_{max}} \sum_{\mathbf{p} \in s_k} p_x, \quad F_5 = \frac{1}{A_k \cdot y_{max}} \sum_{\mathbf{p} \in s_k} p_y,$$

Although quantized color histograms are considered to provide a more detailed description of a region's colors than

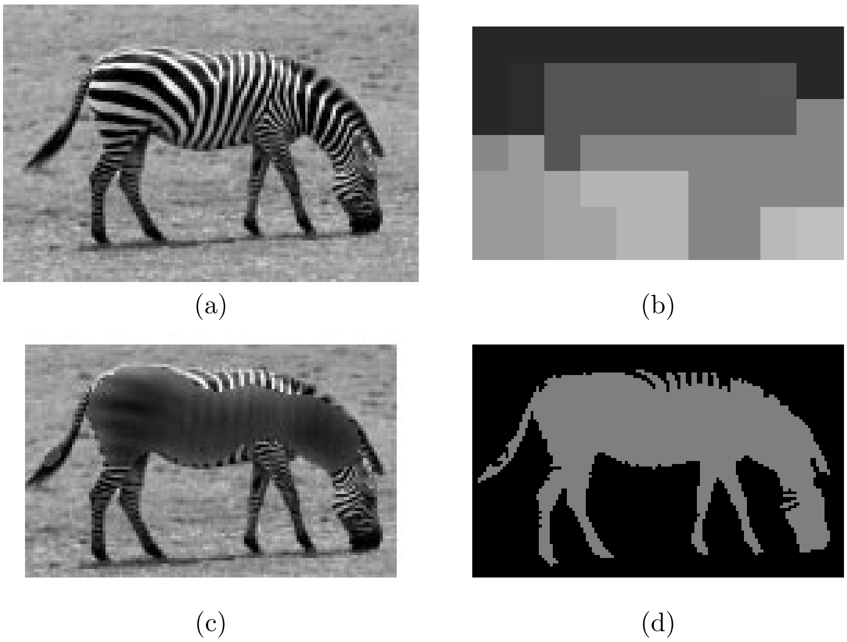


Fig. 2. Segmentation process: starting from the original image (a), initial clustering (b) and conditional filtering (c) are performed and final results (d) are produced.

intensity centers, they were not chosen as color descriptors, since this would significantly increase the dimensionality of the feature space, thus increasing the time-complexity of the query execution.

The **shape descriptors** $F_6, F_7$ of a region are its normalized area and eccentricity. We chose not to take into account the orientation of regions, since orientation is hardly characteristic of an object. The normalized area $F_6$ is expressed by the number of pixels $A_k$ that belong to region $s_k$, divided by the total number of pixels of the image:

$$F_6 = \frac{A_k}{x_{max} \cdot y_{max}}$$

The eccentricity is calculated using the covariance or scatter matrix $\mathbf{C}_k$ of the region. This is defined as:

$$\mathbf{C}_k = \frac{1}{A_k} \sum_{\mathbf{p} \in s_k} (\mathbf{p} - \mathbf{S}(s_k))(\mathbf{p} - \mathbf{S}(s_k))^T \ .$$

where $\mathbf{S}(s_k) = [S_x(s_k) \ S_y(s_k)]^T$ is the region spatial center. Let $\rho_i$, $\mathbf{u}_i, i = 1, 2$ be its eigenvalues and eigenvectors: $\mathbf{C}_k \, \mathbf{u}_i = \rho_i \, \mathbf{u}_i$ with $\mathbf{u}_i^T \, \mathbf{u}_i = 1, \quad \mathbf{u}_i^T \, \mathbf{u}_j = 0, \quad i \neq j$ and $\rho_1 \geq \rho_2$. According to Principal Component Analysis (PCA), the principal eigenvector $\mathbf{u}_1$ defines the orientation of the region and $\mathbf{u}_2$ is perpendicular to $\mathbf{u}_1$. The two eigenvalues provide an approximate measure of the two dominant directions of the shape. Using these quantities, an approximation of the eccentricity $\varepsilon_k$ of the region is calculated, as follows:

$$\varepsilon_k = 1 - \frac{\rho_1}{\rho_2}$$

The normalized eccentricity descriptor $F_7$ is then defined as $F_7 = e^{\varepsilon_k}$.

The seven region descriptors defined above form a region descriptor vector $\mathbf{F}$,

$$\mathbf{F} = [F_1 \ \ldots \ F_7], \tag{8}$$

This region descriptor vector will be used in the sequel both for assigning intermediate-level qualitative descriptors to the region and as input to the relevance feedback mechanism. In both cases the existence of these low-level descriptors is not apparent to the end user.

### B. Object Ontology

In this work, an ontology is employed to allow the user to query an image collection using semantically meaningful concepts (semantic objects), as in [42]. As opposed to [42], though, no manual annotation of images is performed. Instead, a simple *object ontology* is used to enable the user to describe semantic objects, like "tiger", and relations between semantic objects, using a set of *intermediate-level descriptors* and *relation identifiers* (Fig. 3). The architecture of this indexing scheme is illustrated in Fig. 4. The simplicity of the employed object ontology serves the purpose of it being applicable to generic image collections without requiring the correspondence between image regions and relevant identifiers be defined manually. The object ontology can be expanded so as to include additional descriptors and relation identifiers corresponding either to low-level region properties (e.g. texture) or to higher-level semantics which, in domain-specific applications, could be inferred either from the visual information itself or from associated information (e.g. text), should there be any. Similarly to [43], an ontology is defined as follows.

**Definition.** An *object ontology* is a structure (Fig. 3)

$$\mathcal{O} := (\mathcal{D}, \leq_{\mathcal{D}}, \mathcal{R}, \sigma, \leq_{\mathcal{R}})$$

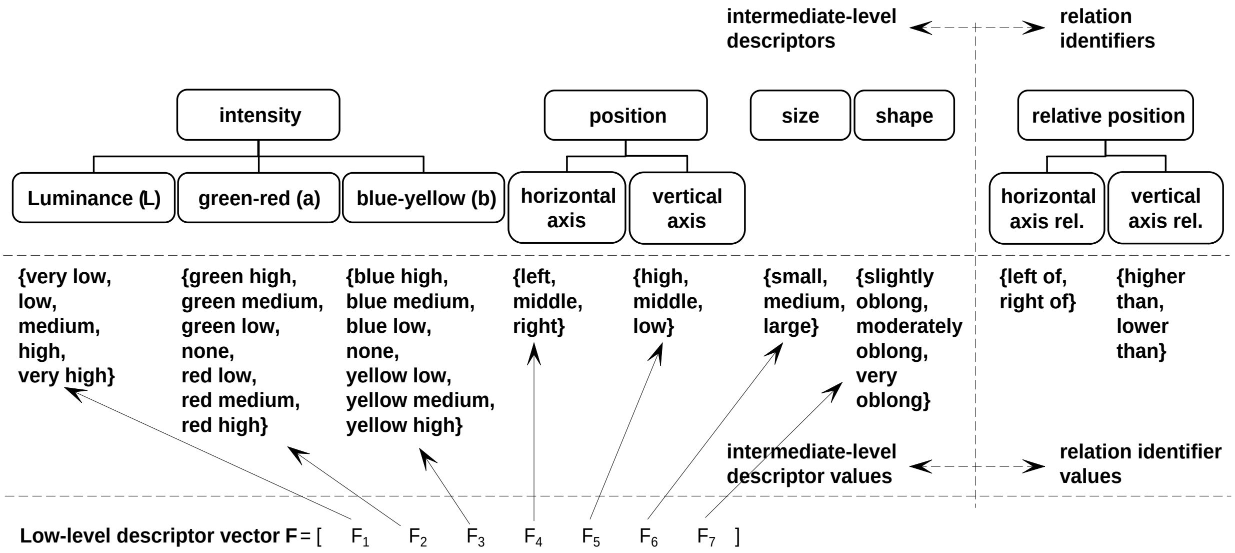

intermediate-level descriptors
relation identifiers
intensity
position
size
shape
relative position
Luminance (L)
green-red (a)
blue-yellow (b)
horizontal axis
vertical axis
horizontal axis rel.
vertical axis rel.
{very low, low, medium, high, very high}
{green high, green medium, green low, none, red low, red medium, red high}
{blue high, blue medium, blue low, none, yellow low, yellow medium, yellow high}
{left, middle, right}
{high, middle, low}
{small, medium, large}
{slightly oblong, moderately oblong, very oblong}
{left of, right of}
{higher than, lower than}
intermediate-level descriptor values
relation identifier values
Low-level descriptor vector F = [ F1 F2 F3 F4 F5 F6 F7 ]

Fig. 3. Object ontology. The intermediate-level descriptors are the elements of set $\mathcal{D}$, whereas the relation identifiers are the elements of set $\mathcal{R}$.

consisting of: (i) Two disjoint sets $\mathcal{D}$ and $\mathcal{R}$ whose elements $d$ and $r$ are called respectively, intermediate level descriptors (e.g. "intensity", "position", etc.) and relation identifiers (e.g. "relative position"). To simplify the terminology, relation identifiers will often be called *relations* in the sequel. The elements of set $\mathcal{D}$ are often called *concept identifiers* or *concepts* in the literature. (ii) A partial order $\leq_{\mathcal{D}}$ on $\mathcal{D}$, called concept hierarchy or taxonomy (e.g. "luminance" is a subconcept of "intensity"). (iii) A function $\sigma : \mathcal{R} \rightarrow \mathcal{D}^+$ called *signature*; $\sigma(r) = (\sigma_{1,r},\ \sigma_{2,r}, \dots\ \sigma_{\Sigma,r})$, $\sigma_{i,r} \in \mathcal{D}$ and $|\sigma(r)| \equiv \Sigma$ denotes the number of elements of $\mathcal{D}$ on which $\sigma(r)$ depends. (iv) A partial order $\leq_{\mathcal{R}}$ on $\mathcal{R}$, called relation hierarchy, where $r_1 \leq_{\mathcal{R}} r_2$ implies $|\sigma(r_1)| = |\sigma(r_2)|$ and $\sigma_{i,r_1} \leq_{\mathcal{D}} \sigma_{i,r_2}$ for each $1 \leq i \leq |\sigma(r_1)|$.

For example, the signature of relation $r$ "relative position" is by definition $\sigma(r) =$ ("position", "position"), indicating that it relates a position to a position; $|\sigma(r)| = 2$ denotes that $r$ involves two elements of set $\mathcal{D}$. Both the intermediate-level "position" descriptor values and the underlying low-level descriptor values can be employed by the "relative position" relation.

In figure 3, the possible intermediate-level descriptors and descriptor values are shown. Each value of these intermediate-level descriptors is mapped to an appropriate range of values of the corresponding low-level, arithmetic descriptor. The various value ranges for every low-level descriptor are chosen so that the resulting intervals are equally populated. This is pursued so as to prevent an intermediate-level descriptor value from being associated with a majority of image regions in the database, because this would render it useless in restricting a query to the potentially most relevant ones. Overlapping, up to a point, of adjacent value ranges, is used to introduce a degree of fuzzyness to the descriptors; for example, both "low luminance" and "medium luminance" values may be used to describe a single region.

Let $d_{q,z}$ be the $q$-th descriptor value (e.g. "low luminance") of intermediate-level descriptor $d_z$ (e.g. "luminance") and $R_{q,z} = [L_{q,z}, H_{q,z}]$ be the range of values of the corresponding arithmetic descriptor $F_m$ (Eq. (8)). Given the probability density function $pdf(F_m)$, the overlapping factor V expressing the degree of overlapping of adjacent value ranges, and given that value ranges should be equally populated, lower and upper bounds $L_{q,z}$, $H_{q,z}$ can be easily calculated according to equations (9), (10) and (11).

$$L_{1,z} = L_m, \qquad \int_{L_{q-1,z}}^{L_{q,z}} pdf(F_m)dF_m = \tfrac{1-V}{Q_z - V\cdot(Q_z-1)}, \quad q = 2, \dots, Q_z, \tag{9}$$

$$\int_{L_{1,z}}^{H_{1,z}} pdf(F_m)dF_m = \frac{1}{Q_z - V\cdot(Q_z-1)}, \tag{10}$$

$$\int_{H_{q-1,z}}^{H_{q,z}} pdf(F_m)dF_m = \frac{1-V}{Q_z - V\cdot(Q_z-1)},\ q = 2, \dots, Q_z, \tag{11}$$

where $Q_z$ is the number of descriptor values defined for descriptor $d_z$ (for example, for "luminance", $Q_z = 5$), and $L_m$ is the lower bound of the values of $F_m$. Note that for descriptors "green-red" and "blue-yellow", the above process is performed twice: once for each of the two complementary colors described by each descriptor, taking into account each time the appropriate range of values of the corresponding low-level descriptor. Lower and upper bounds for value "none" of descriptor "green-red" are chosen so as to associate with this value a fraction $V$ of the population of descriptor value "green low" and a fraction $V$ of the population of descriptor value "red low"; bounds for value "none" of descriptor "blue-yellow" are defined accordingly. The overlapping factor $V$ is defined as $V = 0.25$ in our experiments. The boundaries calculated by the above method

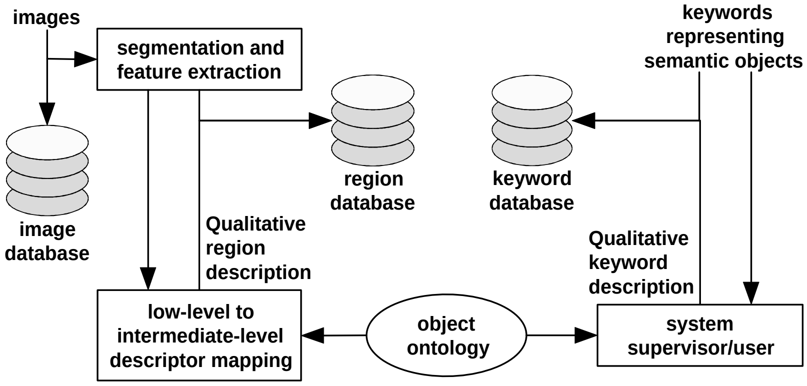

images
segmentation and feature extraction
image database
region database
keyword database
keywords representing semantic objects
Qualitative region description
Qualitative keyword description
low-level to intermediate-level descriptor mapping
object ontology
system supervisor/user

Fig. 4. Indexing system overview. Low-level and intermediate-level descriptor values for the regions are stored in the region database; intermediate-level descriptor values for the user-defined keywords (semantic objects) are stored in the keyword database.

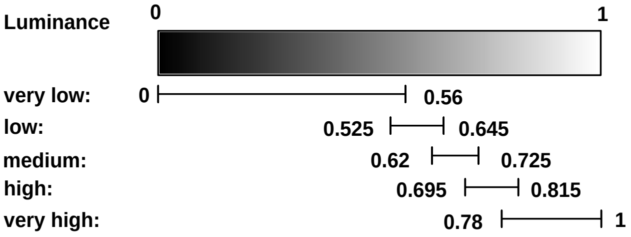


Fig. 5. Correspondence of low-level and intermediate-level descriptor values for the "luminance" descriptor.

for the "luminance" descriptor, using the image database defined in section IV, are presented in figure 5.

### C. Query Process

A query is formulated using the object ontology to provide a qualitative definition of the sought object or objects (using the intermediate-level descriptors) and the relations between them. Definitions previously imported to the system by the same or other users can also be employed, as discussed in the sequel. As soon as a query is formulated, the intermediate-level descriptor values associated with each desired object/keyword are compared to those of each image region contained in the database. Descriptors for which no values have been associated with the desired object (e.g. "shape" for object "tiger", defined in Fig. 6) are ignored; for each remaining descriptor, regions not sharing at least one descriptor value with those assigned to the desired object are deemed irrelevant (e.g. a region with "size": "large" is not a potentially relevant region for a "tiger" query, as opposed to a region assigned both "large" and "medium" values for its "size" descriptor). In the case of dual-keyword queries, the above process is performed for each keyword separately and only images containing at least two distinct potentially relevant regions, one for each keyword, are returned; if desired spatial relations between the queried objects have been defined, compliance with them is checked using the corresponding region intermediate-level and low-level descriptors, to further reduce the number of potentially relevant images returned to the user.

After narrowing down the search to a set of potentially relevant image regions, relevance feedback is employed to produce a quantitative evaluation of the degree of relevance of each region. The employed mechanism is based on a method proposed in [44], where it is used for image retrieval using global image properties under the query-by-example scheme. This combines support vector machines (SVMs) [45], [46] with a constrained similarity measure (CSM) [44]. SVMs employ the user-supplied feedback (training samples) to learn the boundary separating the two classes (positive and negative samples, respectively). Each sample (in our case, image region) is represented by its low-level descriptor vector $\mathbf{F}$ (section III-A). Following the boundary estimation, the CSM is employed to provide a ranking; in [44], the CSM employs the Euclidean distance from the key-image used for initiating the query for images inside the boundary (images classified as relevant) and the distance from the boundary for those classified as irrelevant. Under the proposed scheme, no key-image is used for query initiation; the CSM is therefore modified so as to assign to each image region classified as relevant the minimum of the Euclidean distances between it and all positive training samples (i.e. image regions marked as relevant by the user during relevance feedback). The query procedure is graphically illustrated in Fig. 7.

The relevance feedback process can be repeated as many times as necessary, each time using all previously supplied training samples. Furthermore, it is possible to store the parameters of the trained SVM and the corresponding training set for every keyword that has already been used in

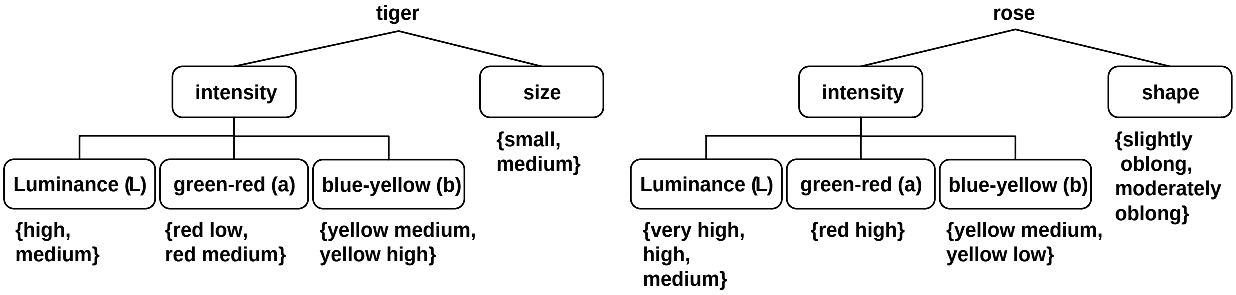


Fig. 6. Exemplary keyword definitions using the object ontology

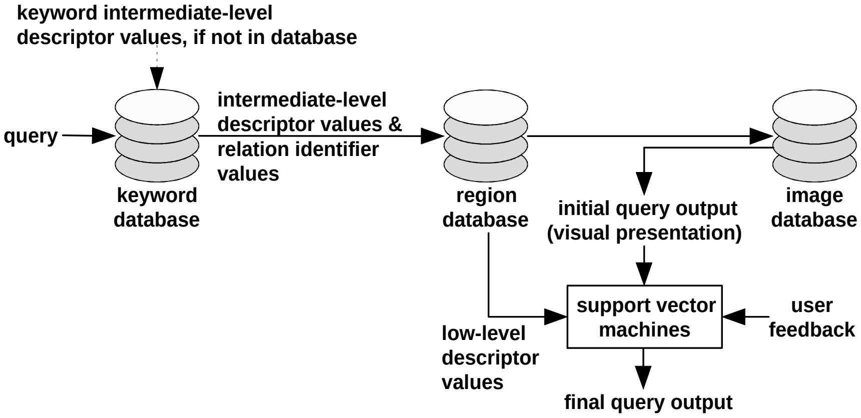

keyword intermediate-level descriptor values, if not in database
query
keyword database
intermediate-level descriptor values & relation identifier values
region database
image database
initial query output (visual presentation)
support vector machines
user feedback
low-level descriptor values
final query output

Fig. 7. Query process overview

a query at least once. This endows the system with the capability to respond to anticipated queries without initially requiring any feedback; in a multiuser (e.g. web-based) environment, it additionally enables different users to share knowledge, either in the form of semantic object descriptions or in the form of results retrieved from the database. In either case, further refinement of retrieval results is possible by additional rounds of relevance feedback.

## IV. Experimental Results

The proposed algorithms were tested on a collection of 5000 images from the Corel gallery [47]. Application of the segmentation algorithm of section II to these images resulted to the creation of a database containing 34433 regions, each represented by a low-level descriptor vector, as discussed in section III-A. The segmentation and low-level feature extraction required on the average 27.15 sec. and 0.011 sec. respectively, on an 2 GHz Pentium IV PC. The proposed algorithm was compared with the Blobworld segmentation algorithm [15]. Segmentation results demonstrating the performance of the proposed and the Blobworld algorithms are presented in figures 8 and 9. Although segmentation results are imperfect, as is generally the case with segmentation algorithms, most regions created by the proposed algorithm correspond to a semantic object or a part of one; even in the latter case, most indexing features (e.g. luminance, color) describing the semantic object appearing in the image can be reliably extracted.

Objective evaluation of segmentation quality was performed using images belonging to various classes and manually generated reference masks (figures 8 and 9). The employed evaluation criterion is based on the measure of *spatial accuracy* proposed in [48] for foreground/background masks. For the purpose of evaluating still image segmentation results, each reference region $g_\kappa$, $\kappa = 1, \ldots, K_g$ of the reference mask (ground truth) is associated with a different region $s_k$ of the created segmentation mask on the basis of region overlapping considerations (i.e. $s_k$ is chosen so that $g_\kappa \cap s_k$ is maximized). Then, the spatial accuracy of the segmentation is evaluated by separately considering each reference region as a foreground reference region and applying the criterion of [48] for the pair of $\{g_\kappa, s_k\}$; during this process, all other reference regions are treated as background. A weighted sum of misclassified pixels for each reference region is the output of this process. The sum of these error measures for all reference regions is used for the objective evaluation of segmentation accuracy; values of the sum closer to zero indicate better segmentation. Numerical evaluation results and comparison using the segmentation masks of figures 8 and 9 are reported in table I.

Following the creation of the region low-level-descriptor database, the mapping between these low-level descriptors and the intermediate-level descriptors defined by the object ontology was performed; this was done by estimating the low-level-descriptor lower and upper boundaries corre-

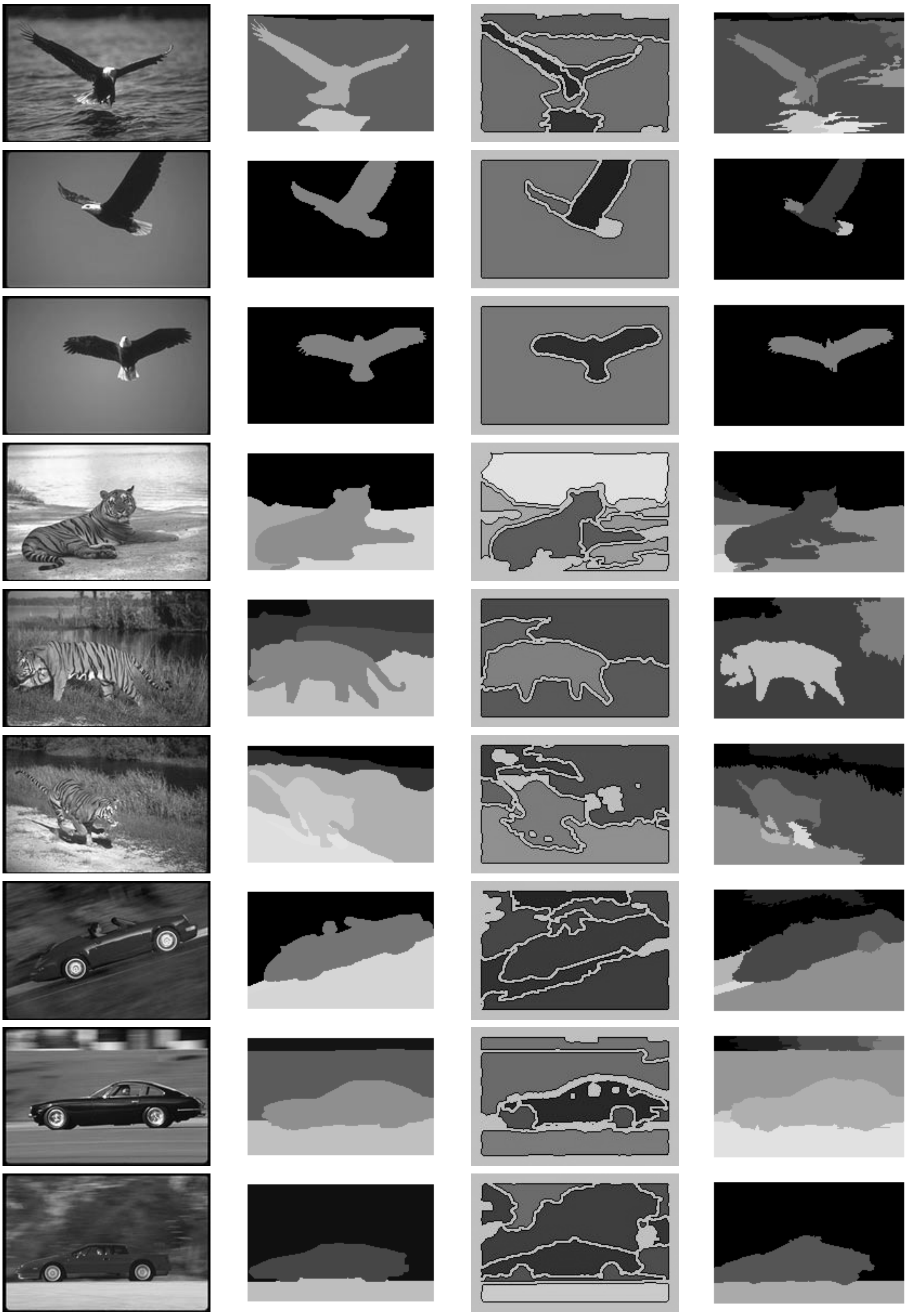


Fig. 8. Segmentation results for images belonging to classes "eagles", "tigers", and "cars". Images are shown in the first column, followed by reference masks (second column), results of the Blobworld segmentation algorithm (third column) and results of the proposed algorithm (fourth column).

sponding to each intermediate-level descriptor value, as discussed in section III-B. Since a large number of heterogeneous images was used for the initial boundary calculation, future insertion of heterogeneous images to the database is not expected to significantly alter the proportion of image regions associated with each descriptor. Thus, the mapping

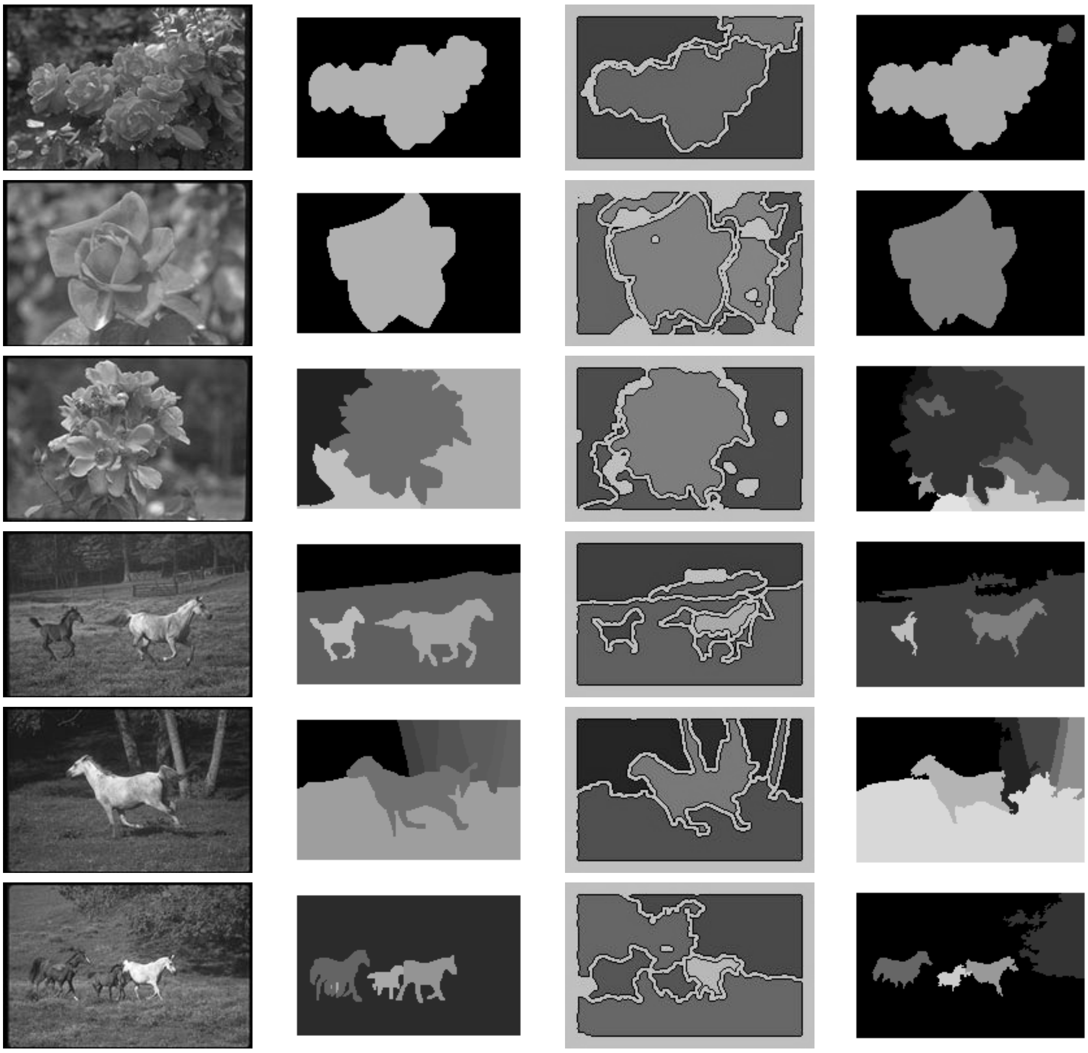


Fig. 9. Segmentation results for images belonging to classes "roses" and "horses". Images are shown in the first column, followed by reference masks (second column), results of the Blobworld segmentation algorithm (third column) and results of the proposed algorithm (fourth column).

TABLE I

Numerical evaluation of segmentation results of Figs. 8 and 9

| | Image 1 | | Image 2 | | Image 3 | |
|---|---|---|---|---|---|---|
| | Blobworld | Proposed | Blobworld | Proposed | Blobworld | Proposed |
| eagle | 163.311871 | **44.238528** | 16.513599 | **7.145284** | 11.664597 | **2.346432** |
| tiger | 90.405821 | **12.104017** | **47.266126** | 57.582892 | 86.336678 | **12.979979** |
| car | 133.295750 | **54.643714** | 54.580259 | **27.884972** | 122.057933 | **4.730332** |
| rose | 37.524702 | **2.853145** | 184.257505 | **1.341963** | **22.743732** | 53.501481 |
| horse | 65.303681 | **17.350378** | 22.099393 | **12.115678** | 233.303729 | **120.862361** |

between low-level and intermediate-level descriptors is not to be repeated, unless the database drastically changes.

The next step in experimenting with the proposed system was to use the object ontology to define, using the available intermediate-level descriptors/ descriptor values, high-level concepts, i.e. real-life objects. Since the purpose of the first phase of each query is to employ these definitions to reduce the data set by excluding obviously irrelevant regions, the definitions of semantic objects need not be particularly restrictive (figure 6). This is convenient from the users' point of view, since the user can not be expected to have perfect knowledge of the color, size, shape and position characteristics of the sought object.

Subsequently, several experiments were conducted using single-keyword or dual-keyword queries, to retrieve images belonging to particular classes, e.g. images containing tigers, fireworks, roses, etc. In most experiments class population was 100 images; under-represented classes were also

TABLE II

Average number of image region pages evaluated for the first round of relevance feedback

| Query | Pages |
|---|---|
| tiger + grass | 2.8 |
| brown horse + grass | 2 |
| bald eagle + blue sky | 2.8 |
| fireworks + black sky | 2 |
| rose | 2.2 |
| sunset | 2.2 |
| red car | 2 |
| yellow car | 5.8 |

used, with population ranging from 6 to 44 images. Performing ontology-based querying resulted in initial query results being produced by excluding the majority of regions in the database, that were found to be clearly irrelevant. As a result, one or more pages of twenty randomly-selected, potentially relevant image regions were presented to the user to be manually evaluated; this resulted in the "relevant" check-box being checked for those that were actually relevant. Usually, evaluating two pages of image regions was found to be sufficient; the average number of image region pages evaluated, when querying for each object class, is presented in table II. Note that in all experiments, each query was submitted five times, to accommodate for varying performance due to different randomly chosen image sets being presented to the user. The average time required for the SVM training and the subsequent region ranking was 0.12 seconds for single-keyword and 0.3 seconds for dual-keyword queries, on an 2 GHz Pentium IV PC. Relevance feedback was then repeated, by manually evaluating the regions contained in the first page of the output of the first relevance feedback round. On the average, 0.13 seconds for single-keyword and 0.33 seconds for dual-keyword queries were required for this round. Results after the second round of relevance feedback are presented in figure 10; precision-recall diagrams for each class of queries after one and two rounds of relevance feedback are presented in figures 11, 12. The term *precision* is defined as the fraction of retrieved images which are relevant, and the term *recall* as the fraction of relevant images which are retrieved [15].

In order to further evaluate the above results, experiments were also conducted using the query by example paradigm and global image histograms, that were introduced in [49] and used widely ever since. The histograms were based on bins of width 20 in each dimension of the L*a*b* color space. Again, each query was submitted five times, each time using a different, randomly selected key-image belonging to the desired class. Comparison (figures 11 and 12) reveals that even after a single stage of relevance feedback the proposed method generally yields significantly better results; a second round of relevance feedback leads to further improvement. In figure 12, results are presented, among others, for a query for a severely under-represented class of objects: *yellow cars*, only 6 images of which are contained in the collection of 5000 images. It can be seen that after the second round of relevance feedback the proposed scheme performs better than global histograms and manages to rank highly at least one such image, whereas the global histogram method [49] retrieves nothing but the key-image used for initiating the query, which already was at the users disposal. Additionally, the diagram calculated for the global histogram method relies on the assumption that it is possible to provide the user with such a key-image, so as to enable initiation of the query; this is at least debatable when it comes to such under-represented classes.

## V. Conclusions

A methodology was presented in this paper for the flexible and user-friendly retrieval of color images, combining a number of image processing and machine learning tools, such as a time-efficient and unsupervised segmentation algorithm, a simple ontology defining intermediate-level descriptors and a relevance feedback mechanism based on support vector machines. The resulting methodology is applicable to generic image collections, where no possibility of structuring a domain-specific knowledge base exists. The proposed scheme overcomes the restrictions of conventional methods, such as the need for the availability of key-images or image captions, and requires no manual tuning of weights, thus offering flexibility and user-friendliness. Experiments conducted on a large collection of images demonstrate the effectiveness of our approach in terms of precision versus recall.

## References

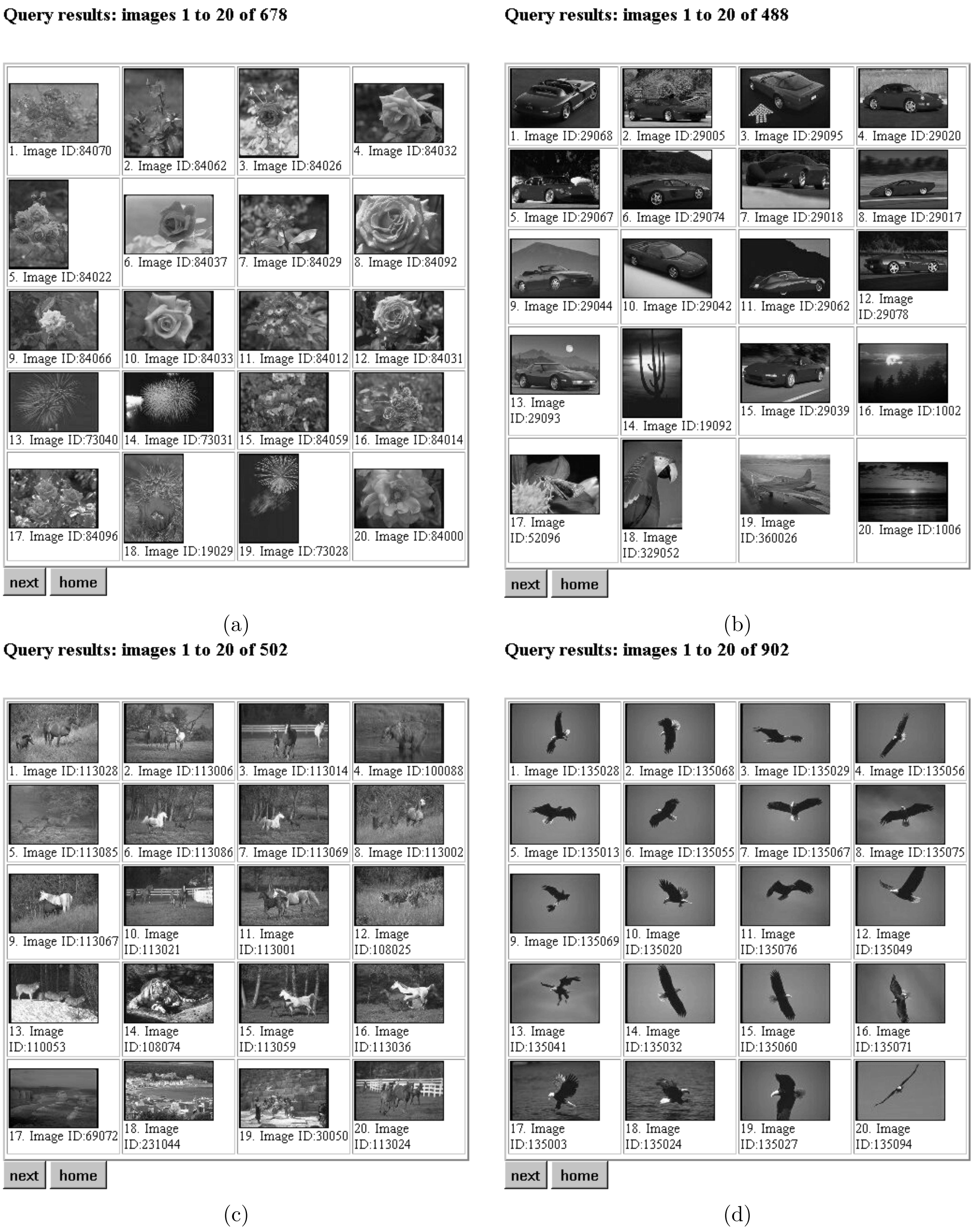
[1] M.R. Naphade and T.S. Huang, "Extracting semantics from audio-visual content: the final frontier in multimedia retrieval," *IEEE Trans. on Neural Networks*, vol. 13, no. 4, pp. 793–810, July 2002.

[2] 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. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, December 2000.

[3] S. Christodoulakis, M. Theodoridou, F. Ho, M. Papa, and A. Pathria, "Multimedia Document Presentation, Information Extraction, and Document Formation in MINOS: A Model and a System," *ACM Transactions on Office Information Systems*, vol. 4, no. 4, pp. 345–383, 1986.

[4] C. Zhang and T. Chen, "An active learning framework for content-based information retrieval," *IEEE Trans. on Multimedia*, vol. 4, no. 2, pp. 260–268, 2002.

[5] S.-F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong, "A Fully Automated Content-Based Video Search Engine Sup-



Query results: images 1 to 20 of 678

1. Image ID:84070
2. Image ID:84062
3. Image ID:84026
4. Image ID:84032
5. Image ID:84022
6. Image ID:84037
7. Image ID:84029
8. Image ID:84092
9. Image ID:84066
10. Image ID:84033
11. Image ID:84012
12. Image ID:84031
13. Image ID:73040
14. Image ID:73031
15. Image ID:84059
16. Image ID:84014
17. Image ID:84096
18. Image ID:19029
19. Image ID:73028
20. Image ID:84000

(a)

Query results: images 1 to 20 of 488

1. Image ID:29068
2. Image ID:29005
3. Image ID:29095
4. Image ID:29020
5. Image ID:29067
6. Image ID:29074
7. Image ID:29018
8. Image ID:29017
9. Image ID:29044
10. Image ID:29042
11. Image ID:29062
12. Image ID:29078
13. Image ID:29093
14. Image ID:19092
15. Image ID:29039
16. Image ID:1002
17. Image ID:52096
18. Image ID:329052
19. Image ID:360026
20. Image ID:1006

(b)

Query results: images 1 to 20 of 502

1. Image ID:113028
2. Image ID:113006
3. Image ID:113014
4. Image ID:100088
5. Image ID:113085
6. Image ID:113086
7. Image ID:113069
8. Image ID:113002
9. Image ID:113067
10. Image ID:113021
11. Image ID:113001
12. Image ID:108025
13. Image ID:110053
14. Image ID:108074
15. Image ID:113059
16. Image ID:113036
17. Image ID:69072
18. Image ID:231044
19. Image ID:30050
20. Image ID:113024

(c)

Query results: images 1 to 20 of 902

1. Image ID:135028
2. Image ID:135068
3. Image ID:135029
4. Image ID:135056
5. Image ID:135013
6. Image ID:135055
7. Image ID:135067
8. Image ID:135075
9. Image ID:135069
10. Image ID:135020
11. Image ID:135076
12. Image ID:135049
13. Image ID:135041
14. Image ID:135032
15. Image ID:135060
16. Image ID:135071
17. Image ID:135003
18. Image ID:135024
19. Image ID:135027
20. Image ID:135094

(d)

Fig. 10. Results for single-object queries (a) "rose" and (b) "red car" and dual-object queries (c) "brown horse - grass" and (d) "bald eagle - blue sky", after the second round of relevance feedback.

porting Spatiotemporal Queries," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 602–615, September 1998.

[6] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and effective querying by image content," *Journal of Intelligent Information Systems*, vol. 3, no. 3/4, pp. 231–262, 1994.

[7] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: the QBIC system," *Computer*, vol. 28, no. 9, pp. 23–32, Sept. 1995.

[8] B.S. Manjunath and W.Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, August 1996.

[9] W.Y. Ma and B.S. Manjunath, "NeTra: A Toolbox for Navigat-

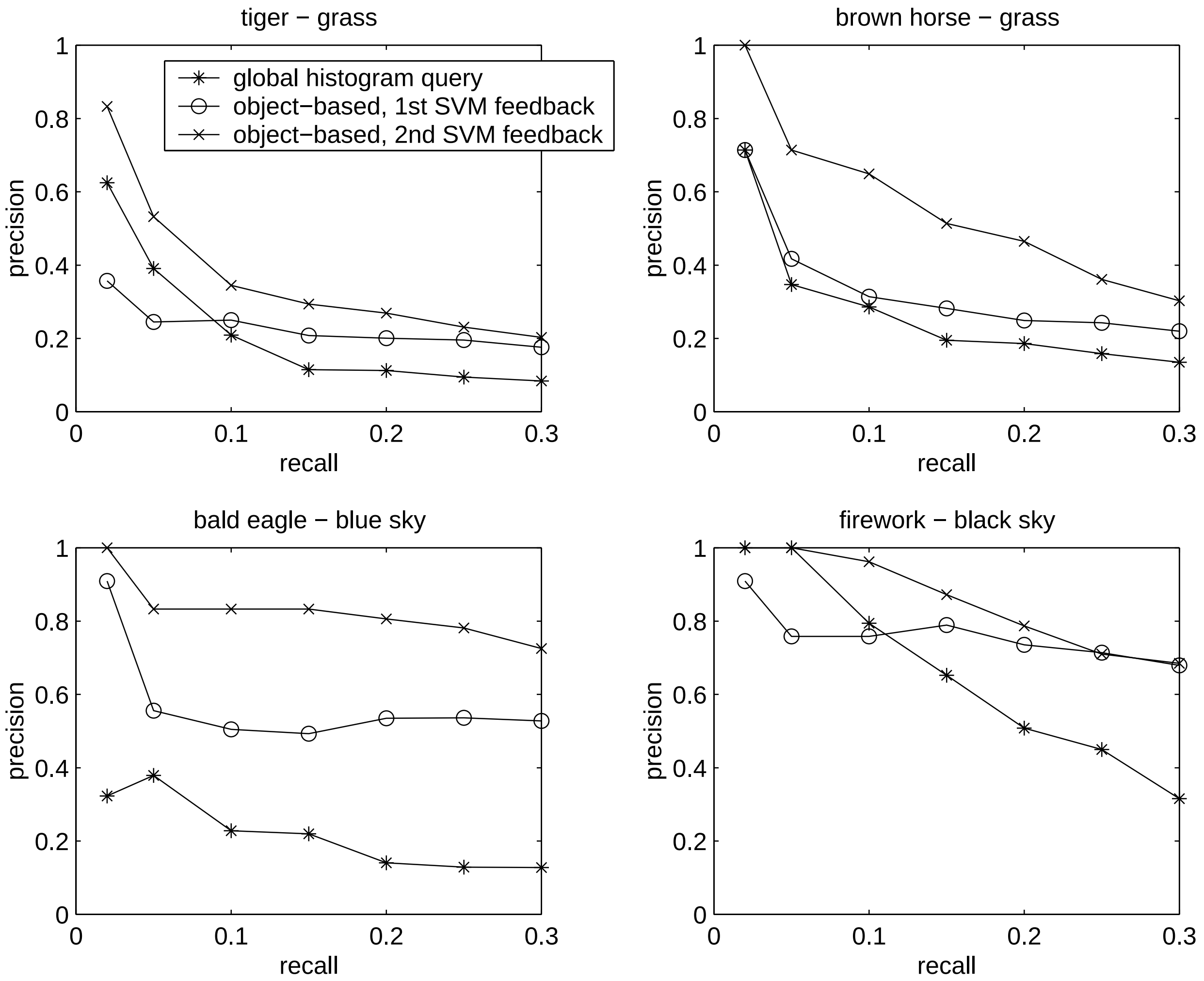


Fig. 11. Precision - Recall diagrams for four two-keyword queries and comparison with global histogram method

ing Large Image Databases," *Multimedia Systems*, vol. 7, no. 3, pp. 184–198, 1999.

[10] T.S. Huang, S. Mehrotra, and K. Ramchandran, "Multimedia analysis and retrieval system (mars) project," in *Proc of 33rd Annual Clinic on Library Application of Data Processing - Digital Image Access and Retrieval*, 1996.

[11] A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *Int. J. Computer Vision*, vol. 18, no. 3, pp. 233–254, 1996.

[12] J.R. Smith and S.-F. Chang, "Visualseek: A fully automated content-based image query system," in *ACM Multimedia*, 1996, pp. 87–98.

[13] I. Kompatsiaris, E. Triantafillou, and M. G. Strintzis, "Region-Based Color Image Indexing and Retrieval," in *Proc. IEEE International Conference on Image Processing*, Thessaloniki, Greece, October 2001.

[14] S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Color- and texture-based image segmentation using EM and its application to content-based image retrieval," in *Proc. Sixth International Conference on Computer Vision*, January 1998.

[15] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, 2002.

[16] P. Perona and J. Malik, "Scale Space and Edge-Detection Using Anisotropic Diffusion," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.

[17] E. Tuncel and L. Onural, "Utilization of the recursive shortest spanning tree algorithm for video-object segmentation by 2-D affine motion modeling," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 776–781, Aug. 2000.

[18] H. Gao, W.-C. Siu, and C.-H. Hou, "Improved techniques for automatic image segmentation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp. 1273–1280, December 2001.

[19] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, August 2000.

[20] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[21] M.R. Naphade, T. Kristjansson, B. Frey, and T.S. Huang, "Probabilistic multimedia objects (multijects): a novel approach to video indexing and retrieval in multimedia systems," in *Proc. International Conference on Image Processing*, 1998, vol. 3, pp. 536–540.

[22] Y. Lu, C. Hu, X. Zhu, H. Zhang, and Q. Yang, "A unified framework for semantics and feature based relevance feedback in image retrieval systems," in *Proc. ACM Multimedia Conf.*, 2000, pp. 31–37.

[23] X.S. Zhou and T.S. Huang, "Unifying keywords and visual contents in image retrieval," *IEEE Multimedia*, vol. 9, no. 2, pp. 23–33, April-June 2002.

[24] A. Yoshitaka and T. Ichikawa, "A survey on content-based retrieval for multimedia databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 1, pp. 81–93, Jan/Feb 1999.

[25] W. Al-Khatib, Y.F. Day, A. Ghafoor, and P.B. Berra, "Se-

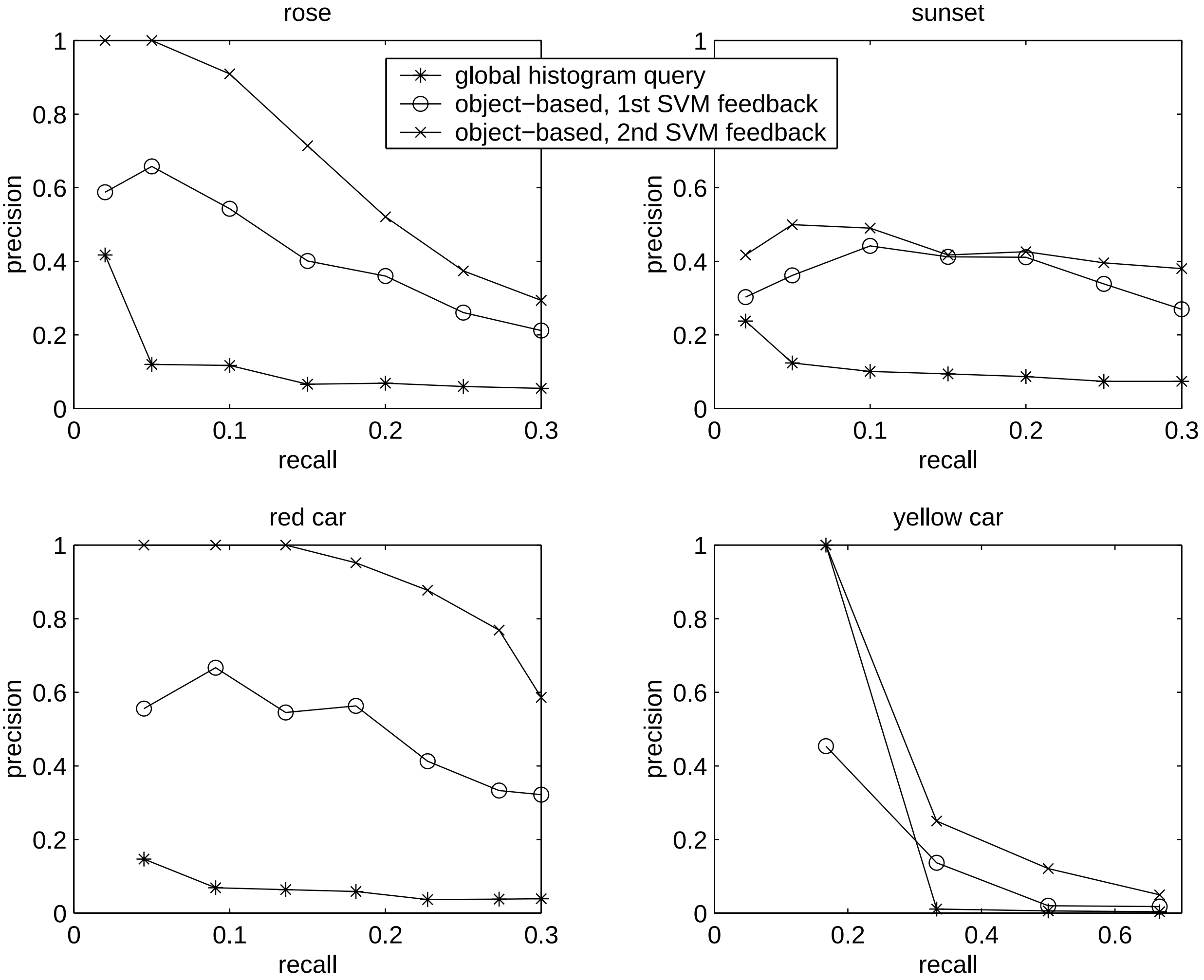


Fig. 12. Precision - Recall diagrams for four single-keyword queries and comparison with global histogram method. An example of querying for a severely under-represented class of images (yellow car, 6 such images contained in the collection) is also shown.

mantic modeling and knowledge representation in multimedia databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 1, pp. 64–80, Jan/Feb 1999.

[26] G. Tsechpenakis, G. Akrivas, G. Andreou, G. Stamou, and S.D. Kollias, "Knowledge-Assisted Video Analysis and Object Detection," in *Proc. European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems (Eunite02)*, Algarve, Portugal, September 2002.

[27] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 644–655, September 1998.

[28] N.D. Doulamis, A.D. Doulamis, and S.D. Kollias, "Nonlinear relevance feedback: improving the performance of content-based retrieval systems," in *Proc. IEEE International Conference on Multimedia and Expo (ICME 2000)*, 2000, vol. 1, pp. 331–334.

[29] V. Kashyap, K. Shah, and A.P. Sheth, "Metadata for building the multimedia patch quilt," in *Multimedia Database System: Issues and Research Direction, Springer Verlag*, pp. 297–319. 1995.

[30] B. Chandrasekaran, J.R. Josephson, and V.R. Benjamins, "What are ontologies, and why do we need them?," *IEEE Intelligent Systems*, vol. 14, no. 1, pp. 20–26, Jan.-Feb. 1999.

[31] S. Staab, R. Studer, H.-P. Schnurr, and Y. Sure, "Knowledge processes and ontologies," *IEEE Intelligent Systems*, vol. 16, no. 1, pp. 26–34, Jan.-Feb. 2001.

[32] P. Martin and P.W. Eklund, "Knowledge retrieval and the World Wide Web," *IEEE Intelligent Systems*, vol. 15, no. 3, pp. 18–25, May-June 2000.

[33] A. Raouzaiou, N. Tsapatsoulis, V. Tzouvaras, G. Stamou, and S.D. Kollias, "A Hybrid Intelligence System for Facial Expression Recognition," in *Proc. European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems (Eunite02)*, Algarve, Portugal, September 2002.

[34] J. McQueen, "Some Methods for Classification and Analyis of Multivariate Observations," in *5th Berkely Symp. on Math. Stat. and Prob.*, 1967, vol. 1, pp. 281–296.

[35] I. Kompatsiaris and M. G. Strintzis, "Spatiotemporal Segmentation and Tracking of Objects for Visualization of Videoconference Image Sequences," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 8, December 2000.

[36] N. V. Boulgouris, I. Kompatsiaris, V. Mezaris, and M. G. Strintzis, "Content-based Watermarking for Indexing Using Robust Segmentation," in *Proc. Workshop on Image Analysis For Multimedia Interactive Services*, Tampere, Finland, May 2001.

[37] V. Mezaris, I. Kompatsiaris, and M.G. Strintzis, "A framework for the efficient segmentation of large-format color images," in *Proc. International Conference on Image Processing*, 2002, vol. 1, pp. 761–764.

[38] S. Liapis, E. Sifakis, and G. Tziritas, "Color and/or Texture Segmentation using Deterministic Relaxation and Fast Marching Algorithms," in *Intern. Conf. on Pattern Recognition*, September 2000, vol. 3, pp. 621–624.

[39] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. on Image Processing*, vol. 4, no.

11, pp. 1549–1560, Nov. 1995.

[40] J.T. Tou and R.C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley, 1974.

[41] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*, McGraw-Hill International Editions, 1995.

[42] A.T. Schreiber, B. Dubbeldam, J. Wielemaker, and B. Wielinga, "Ontology-based photo annotation," *IEEE Intelligent Systems*, vol. 16, no. 3, pp. 66–74, May-June 2001.

[43] E. Bozsak, M. Ehrig, S. Handschuh, A. Hotho, A. Maedche, B. Motik, D. Oberle, C. Schmitz, S. Staab, L. Stojanovic, N. Stojanovic, R. Studer, G. Stumme, Y. Sure, J. Tane, R. Volz, and V. Zacharias, "KAON - Towards a large scale Semantic Web," in *Proc. Third Int. Conf. on E-Commerce and Web Technologies (EC-Web 2002)*, Aix-en-Provence, France, Sept. 2002.

[44] G.-D. Guo, A.K. Jain, W.-Y. Ma, and H.-J. Zhang, "Learning similarity measure for natural image retrieval with relevance feedback," *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 811–820, July 2002.

[45] V.N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[46] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[47] *Corel stock photo library*, Corel Corp., Ontario, Canada.

[48] X. Marichal P. Villegas and A. Salcedo, "Objective Evaluation of Segmentation Masks in Video Sequences," in *Proc. Workshop on Image Analysis For Multimedia Interactive Services (WIAMIS99)*, Berlin, Germany, May 1999, pp. 85–88.

[49] M. Swain and D. Ballard, "Color indexing," *Int. J. Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.