myjournal manuscript No.
(will be inserted by the editor)

Supporting Efficient Multimedia Database Exploration

Wen-Syan Li, K. Selçuk Candan*, Kyoji Hirata, Yoshinori Hara

C&C Research Laboratories, NEC USA, Inc.

110 Rio Robles, M/S SJ100, San Jose, CA 95134, USA

Email: {wen,candan,hirata,hara}@ccrl.sj.nec.com

Phone:(408)943-3008 Fax:(408)943-3099

Received: date / Revised version: date

Abstract Due to the fuzziness of query specification and media matching, multimedia retrieval is conducted by ways of exploration. It is essential to provide feedback so that users can visualize query reformulation alternatives and database content distribution. Since media matching is an expensive task, another issue is how to efficiently support exploration so that the system would not be overloaded by perpetual query reformulation. In

Send offprint requests to:

* *This work was performed when the author visited NEC, CCRL. The author's current address is Computer Science and Engineering Department College of Engineering and Applied Sciences Arizona State University Box 875406 Tempe, AZ 85287-5406, Email: candan@asu.edu*

this paper, we present a uniform framework to represent statistical information of both semantics and visual metadata for images in the databases. We propose the concept of *query verification*, which evaluates queries using statistics, and provides users with feedback, including the strictness and reformulation alternatives of each query condition as well as estimated numbers of matches. With query verification, the system increases the efficiency of the multimedia database exploration for both users and the system. Such statistical information is also utilized to support progressive query processing and query relaxation.

Keywords. Multimedia database, exploration, query relaxation, progressive processing, selectivity statistics, human computer interaction.

1 Introduction

Query processing in multimedia databases is different from query processing in traditional database systems. Contents stored in traditional database systems are generally precise and, as a result, query processing answers are deterministic. On the other hand, in both document retrieval and image retrieval, results are based on similarity calculations. In document retrieval, documents are represented as keyword lists. To retrieve a document, information systems compare keywords specified by users with the documents' keyword lists. Images, in a similar manner, are usually represented as media features. However, image matching is carried out through comparing these feature vectors. Comparison of these three types of query processing are

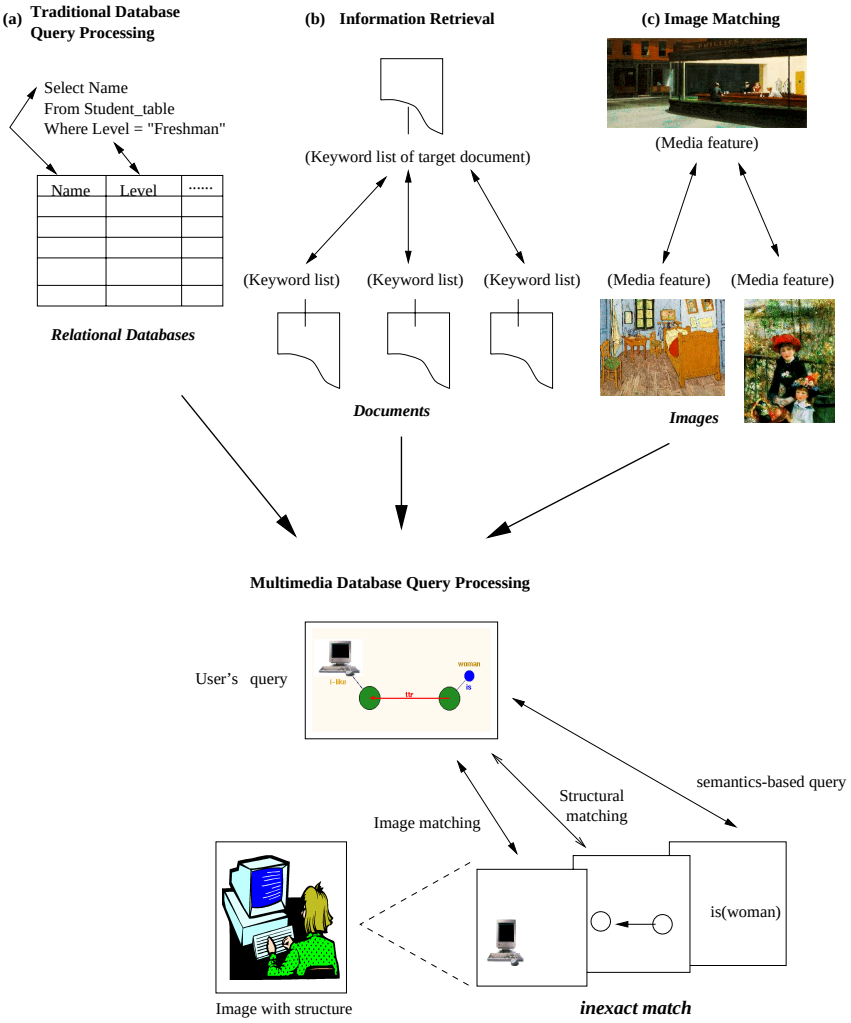

Fig. 1 Multimedia databases: integration of traditional query processing, IR, and image matching

illustrated at the top of Figure 1. We see that multimedia database query processing is truly an integration of these three types of query processing.

An image consists of three types of information representing its contents: visual features, structural layout (spatial relationships), and semantics of

image objects, as shown at the bottom of Figure 1. A multimedia database query requires similarity measures in all these aspects. For example, a query may be posed as "retrieve images in which there is a woman to the right of an object and the object is visually similar to the provided image". The query results are a list of candidate images ranked by their aggregated scores with degrees of uncertainty based on all above three aspects. Thus, we view multimedia database query processing as a combination of (1) information retrieval notions described in [1] (exploratory querying, inexact match, query refinement) and (2) ORDBMS or OODBMS database notions (recognition of specific concepts, variety of data types, spatial relationships between objects).

In most of multimedia applications, supporting partial match capability, in contrast to supporting *only* exact match functionalities in relational DBMSs, is essential and desirable. There are two major reasons.

1. There may not be a reasonable number of images which match with the user query. Figure 2(Query) shows the conceptual representation of the above query. Figures 2(a), (b), (c), and (d) shows examples of candidate images that may match this query. The numbers next to the objects in these candidate images denote the similarity values for the object level matching. The candidate image in Figure 2(a) satisfies object matching conditions but its layout does not match user specification. Figures 2(b) and (d) satisfy image layout condition but objects do not perfectly match

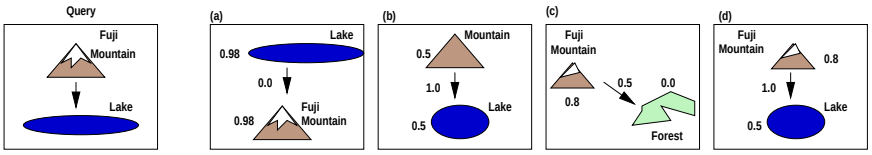

Fig. 2 Query Image and Candidates of Partial matches

the specification. Figure 2(c) has structural and object matching with low scores.

Note that in Figure 2(a), the spatial predicate, and in Figure 2(c), the image similarity predicate for lake completely fail (i.e., the match is 0.0). Such candidate images in general would not be returned by SQL/DBMS-based image retrieval systems. Query relaxation to include such types of partially matched images is needed.

2. The users may not be able to specify queries precisely or correctly due to so called "word mismatch problem" described and studied in the field of information retrieval[2]. Another reason for such misspecified image queries is that it is difficult for the users to describe a color or a shape without ambiguity.

Due to the large volume of data, the unstructured or semistructured nature of information, and the vagueness in the concept of a match, querying multimedia databases should be conducted in an exploratory fashion guided by system feedback. In other words, image retrieval should be performed through computer human interaction (CHI). We define a CHI exploration cycle as a user query followed by a sequence of query reformulations guided by the system for honing target images. In the design of a multi-

media database system, along with the optimization of query processing, we concentrate on the improvement of the speed of the image retrieval in a whole CHI exploration cycle.

We view media retrieval as transitions of corresponding query result space. By reformulating a query, the result space that a user sees is shifted from one to another until the target result space is reached. We believe that media retrieval process can be improved through well-designed computer human interaction. Li et al.[4] introduced the concept of *system facilitated exploration*, in which users interact with the multimedia database system to reformulate queries. This approach is beyond browsing which is supported by most existing systems. However, some drawback observed is that the system can only provide feedback *after* query processing and feedback is limited to *semantics-based* query criteria.

In this paper we present a new approach to media retrieval by systematically facilitating users to reformulate queries in the scope of the SEMCOG Multimedia Database System[3] developed at NEC C&C Research Laboratories in San Jose. SEMCOG provides *query verification* functionalities, which presents users various types of feedback regarding the query and the database content to assist users in exploring multimedia databases more efficiently. The feedback includes (1) query reformulation alternatives for both semantics- and visual characteristics-based query criteria; (2) estimated number of matched images; and (3) database content distribution related query conditions. With this information, users are facilitated to

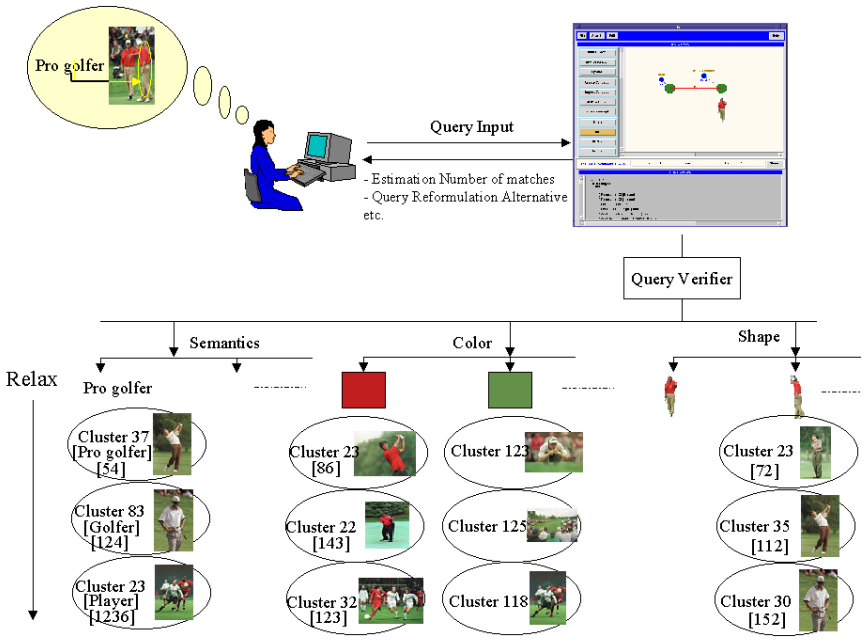

Fig. 3 Using multimedia statistics for facilitating query and exploration

"navigate" from current result space toward the target result space, rather than ad-hoc trial and error on image-by-image basis. The facilitation allows the users reformulate queries interactively based on system feedback and query analysis.

To support such facilitation efficiently, the system utilizes multimedia content statistics. Using content statistics for result estimation and query optimization has been studied for decades in the scope of traditional databases. To extend it to multimedia contents is not an easy task. We develop a uniform framework for representing statistics for both textual and media data. SEMCOG classifies media objects into clusters based on semantics and visual characteristics. The number of images in each cluster is viewed as the

selectivity for such criteria, which can be color, shape, or object position. In Figure 3, a user issues a query image and specifies one of the image objects as *Pro Golfer*. Based on the selectivity values of media and semantics specified in the query, SEMCOG estimates the number of matching images and provides alternative conditions, such as related semantic terms or similar colors, for the user to reformulate the query. Based on the indexing scheme, SEMCOG also supports automated query relaxation and progressive processing to generate the top ranked results incrementally.

The rest of the paper is organized as follows: We first present background information about our system. In Sections 3 and 4, we describe the schemes to compute the selectivity for semantics and visual characteristics of multimedia contents. In Section 5, we present the usage of these indices and selectivity values for (1) estimation of matching images; (2) automated query relaxation; and (3) progressive processing. We also present experimental results to estimate the efficiency of image retrieval by clustering. In Section 6, we review related work. In Section 7 we offer our concluding remarks.

2 Object-based Modeling and Query Language

In this section, we give a summary of modeling and language design in SEMCOG. We only describe the material related to the work presented in this paper. Additional information about the system architecture, language syntax, and user interface can be found in [3].

We view images as compound objects containing multiple component objects and their spatial relationships. The semantic interpretation of an object is its *real world meaning*. The visual interpretation of an object, on the other hand, is what it looks like based on perception of human or image matching engines. Our object extraction technique is based on region-division. A region is defined as homogeneous (in terms of colors) and continuous segments identified in an image. SEMCOG stores and retrieves atomic and compound objects. This enables the query verifier to create finer granularity feedback: instead of returning feedback that encompasses the whole image, the query verifier can now return feedback that describes alternatives for semantics, visual contents, and spatial relationships of the constituent objects. In addition, the hierarchical structure of the image content allows the system to choose the level of feedback granularity depending on the needs and inputs of the user. Furthermore, since the data model captures alternative semantic and visual representations and the corresponding *confidence* (or matching) values, the system can manipulate the objects for query reformulation very naturally.

When an image is *registered* in SEMCOG, content-independent metadata, such as size, format, etc., are automatically extracted, while some content-dependent metadata, such as semantics of images, cannot be extracted automatically. Figure 4 illustrates media object extraction and semantics specification in our system. On the left of the figure, there is a set of images. Regions in these images are extracted based on color homo-

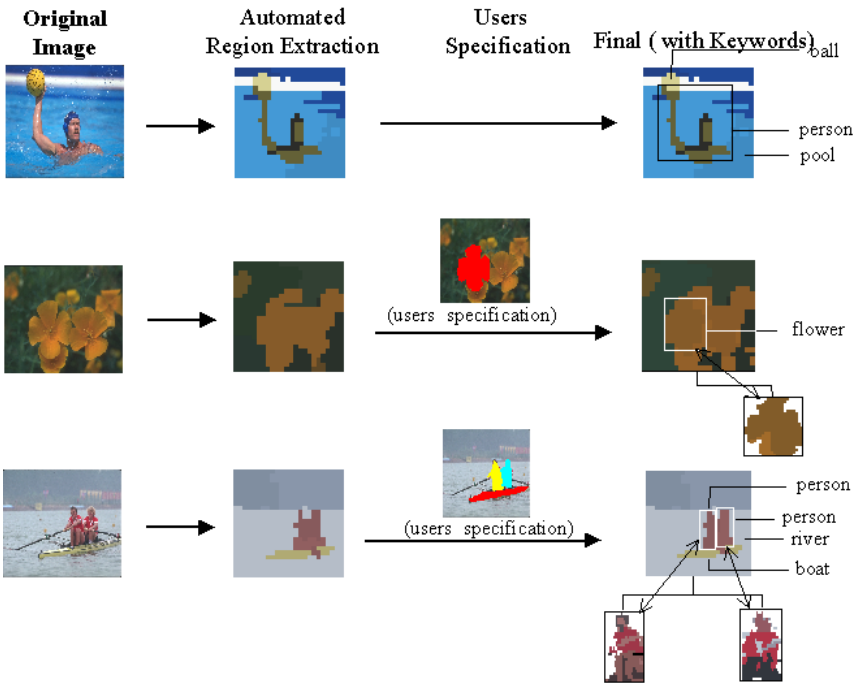

Fig. 4 Region and semantics specification

geneities and segment continuities. The region extraction results are shown to the user for verification. In this example, for the second and third images, the user overrides the system's recommendation and specifies proper region segmentations. After confirmation or modification of the segments, the user specifies the semantics of each region using the tool shown in Figure 5. The segmented regions and the corresponding semantics and visual characteristics are then stored in the database. Note that the number of the regions identified depends on the image content and the level of detail the user requests. In SEMCOG, we set a constraint that the number of regions for one level of the hierarchy must be less than or equal to 8.

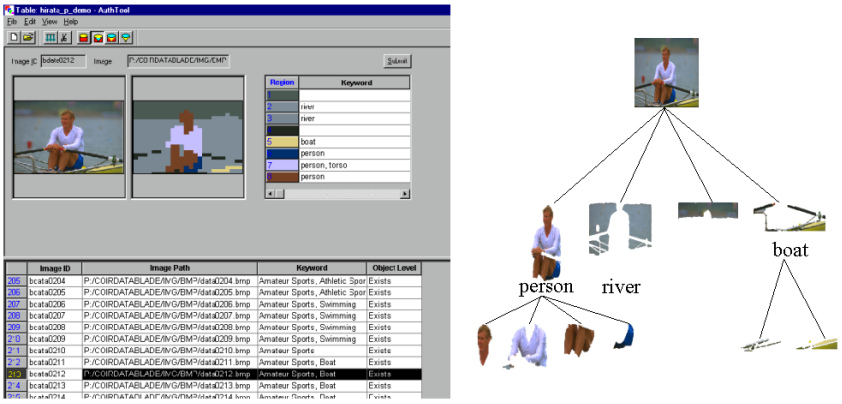

Fig. 5 Image object semantics specification process

Like the data model used in SEMCOG, the Cognition and Semantics-based Query Language, CSQL, used for image retrieval by SEMCOG, is inherently suitable for query reformulation. CSQL consists of predicates which correspond to different levels of query strictness:

- `Semantics-based selection predicates.` The following three predicates can be used to describe different levels of semantic relationships:
  - *is*: The *is* predicate returns true if and only if both arguments have identical semantics (man vs. man).
  - *is_a*: The *is_a* predicate returns true if and only if the second argument is a generalization of the first one (car is_a transportation).
  - *s_like*: The *s_like* (i.e. semantically like) predicate returns true if and only if both arguments are semantically similar (man vs. male).
- `Cognition-based selection predicate`: The cognition-based predicate introduced in CSQL is *i_like* (i.e. image like). In fact, *i_like* is a combination of group of feature-specific predicates, such as *i_like_hist*

and *i_like_shape*. If the user specifies *i_like*, then SEMCOG uses all possible features and returns a single, merged similarity value. The corresponding weights of the features are assumed to be set by the user earlier. Alternatively, users can specify a conjunction of feature-specific predicates along with the corresponding weights. SEMCOG uses only the specified features along with the specified weights to identify the final similarity value.

– `Spatial relationship-based selection predicate`: CSQL supports the following spatial predicates: *above*, and *below*, *to_the_right_of*, and *to_the_left_of*.

By definition, the *s_like* and *i_like* predicates act as implicit query relaxation tools: they let non-exact matches in the result. Furthermore, the query verifier can explicitly relax the query by changing the constant terms in a given query, as done by many relaxation systems, as well as by changing semantic, visual, and spatial predicates. Complete syntax definitions of the CSQL query language is described in [3].

3 Semantics Indexing and Selectivity Construction

As is the case in most database systems, SEMCOG periodically collects database statistics for query optimization purposes. One unique usage of image object semantics selectivity statistics in SEMCOG is to use the statistics to recommend query reformulation alternatives to users, estimate numbers

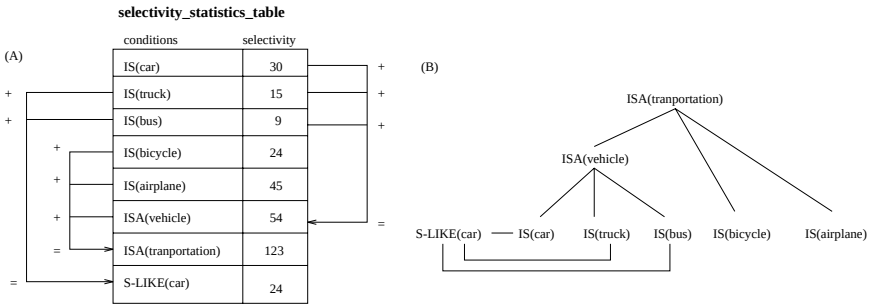

Fig. 6 *Selectivity* as a hierarchical structure

of matches, and automated query relaxation. In this section, we describe the indexing schemes for semantics selectivity values.

3.1 Semantics Selectivity Statistics

In SEMCOG, semantics are indexed within a hierarchical structure as shown in Figure 6. This structure benefits from the relationships between different predicates and their results in improving the utilization of database statistics. Figure 6 shows a hierarchical structure for semantic selectivity for objects related to transportation. This hierarchical structure is constructed by consulting with an on line dictionary, *WordNet*[5]. Two important uses of the hierarchical structure are as follows: (1) some predicates can be translated into a disjunction of other predicates. For example, *ISA(vehicle,X)* can be translated into a disjunction of *IS(car,X)*, *IS(truck,X)*, and *IS(bus,X)*; and (2) clustered predicates can be *reformulated* from/to each other. For example, *IS(car,X)* can be reformulated horizontally to *S-LIKE(car,X)* or vertically to *ISA(vehicle,X)* or *ISA(transportation,X)*.

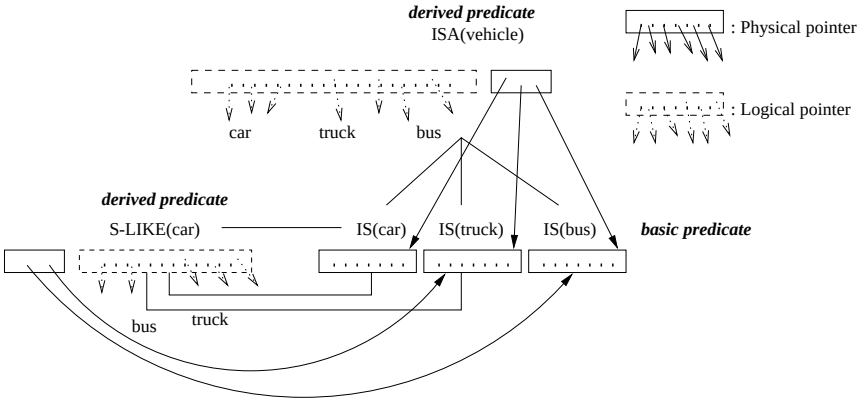

Fig. 7 Hierarchical structured logical pointers

Hierarchically structured indexing takes advantage of the first property above. In order to provide quick response time, we build indices for each predicate. There are two types of predicates: *basic* and *derived* predicates. All *IS()* predicates are *basic* predicates, while *S-LIKE()* and *ISA()* predicates are *derived*. As shown in Figure 7, *ISA(vehicle,X)* can be derived from a disjunction of *IS(car,X)*, *IS(truck,X)*. Similarly, *IS(bus,X)* and *ISA(transportation,X)* can be derived from a disjunction of *ISA(vehicle,X)*, *IS(bicycle,X)*, and *IS(airplane,X)*.

The semantic index construction is done in two steps: We first build indices for *basic* predicates. We call this type of indices *physical* (or *direct*) indices. For instance, in the above example, indices for *IS(car,X)*, *IS(bus,X)*, and *IS(truck,X)* are physical pointers pointing to corresponding objects. Then, we build indices for derived predicates. These indices, on the other hand, are logical. As shown in Figure 7, the index for *ISA(vehicle,X)* is a logical index for tuples whose semantics are car, truck, or bus. However,

the physical index consists of only three pointers pointing to the physical indices of *IS(car,X)*, *IS(car,X)*, and *IS(truck,X)*. Similarly, *S-LIKE(car,X)* consists of only two physical pointers while logically consists of pointers to all objects whose semantics are truck or bus.

3.2 Semantic Relevance

Relevant semantics and their selectivity are used for dealing with the word mismatch problems (i.e. authors and users may use different vocabularies). We build the relevant semantics database by collecting all the relevant terms associated with the semantic terms specified in the database. For a given word, the system retrieves its hypernyms, synonyms, and holonyms whose "similarity" (word distance) is within a given threshold.

For example, for the word "person", by consulting WordNet, we find a long list of terms which are relevant to "person", including *man, woman, human,* and *child*. Suppose that the database contains a set of objects whose semantics are *man, woman,* and *human*, but there is no object whose semantics is specified as *person* or *child*. Our system can provide feedback showing that the term *person* does not exist in the database and provides the user with alternative terms which are relevant to *person* and exist in the database. In this example, the system can show the user alternative terms for *IS(person,X)* as *IS(man,X), IS(woman,X),* and *ISA(human,X)*. Note that SEMCOG does not need to show the user the term *child* since there will not be a match for a query to retrieve images containing a child.

System feedback also includes term and predicate similarity values and selectivity values of each alternatives. The calculation of selectivity values is shown in Figure 6. WordNet[5], provides semantic distance[6], $0.0 \leq Distance(\alpha, \beta) \leq 1.0$, for given two terms α and β. Although the distance values are arguable, we leave this task to domain experts in the field of linguistics. We compute the similarities between different predicates, based on the term distance values returned by the dictionary, as follows:

- Similarity between *IS(α,X)* and *IS(β,X')*,[1] where α and β are two terms, is $1 - Distance(\alpha, \beta)$. The similarity between *IS(α,X)* and *IS(β,X')* is defined as the average similarity of pairs of terms, t_1 and t_2, where t_1 satisfies *IS(α,X)* and t_2 satisfies *IS(β,X')*. Since the only term which satisfies *IS(α,X)* is α and since the only term which satisfies *IS(β,X')* is β, the similarity of the predicates is equal to the similarity of α and β, which is, by definition, $1 - Distance(\alpha, \beta)$.
- Similarity between *IS(α,X)* and *S-LIKE(α,X')*, where α is a term, is $\frac{\sum_{i=1}^{n}(1-Distance(\alpha,s_i))}{n}$,

  n is the number of terms semantically similar to (or synonym of) α and each s_i $(1 \leq i \leq n)$ is such a term. Such a similarity value can be used for query reformulation from, for instance, from *IS(car,X)* to *S-LIKE(car,X')*. In this example, $s_i \in \{truck, bus, \ldots\}$.

[1] X and X' are the same variables, but their corresponding tuples which satisfy the predicates are different due to query reformulation from *IS(α,X)* and *IS(β,X')*. We use the same notation in the rest of this sub-section.

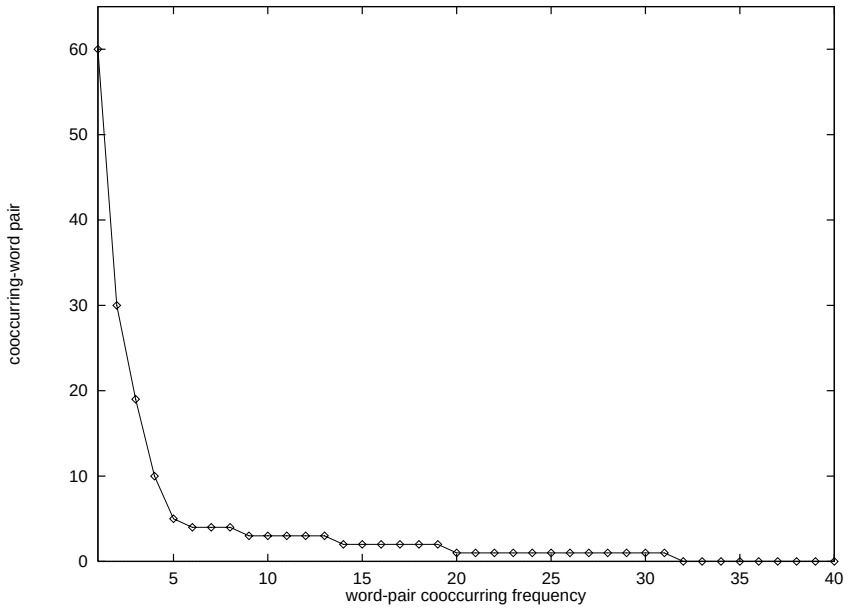

Fig. 8 Cooccurring-Word Pair Frequency Distribution

– Similarity between *IS(α,X)* and *ISA(β,X')* is $\frac{\sum_{i=1}^{m}(1-Distance(\alpha,s_i))}{m}$, α and β are two terms and β is a hyponym of α and m is the number of hypernyms (excluding α) of β and each s_i $(1 \leq i \leq m)$ is a hypernyms (different from α) of X'. Such a similarity value can be used for query reformulation from, for instance, from *IS(car,X)* and *IS-A(transportation,X')*. In this example, $\beta = transportation$ and $s_i \in \{truck, bus, airplane, \ldots\}$.

3.3 Semantics Cooccurrence

Note that all of cooccurring object instances can be computed in real time. We store a subset of cooccurring object instances which are of higher fre-

quency as indices in order to provide fast response for computer human interaction.

SEMCOG maintains two-object cooccurrence statistics. This information is used for providing feedback when the user specifies an object with a given semantics and wants to know what other objects are also in the same image. Note that all of cooccurring object instances can be computed in real time. We store a subset of cooccurring object instances which are of higher frequency as indices in order to provide fast response for computer human interaction. Note that our system uses object cooccurrence information as feedback to show users what objects are in the same image as the object they specify. To pre-compute and store all object cooccurrence instances is not necessary since showing users the cooccurring objects with higher frequency is sufficient for the feedback purposes.

Let's denote $I_{i,j} = Cooccurring(Keyword_i, Keyword_j)$ as the total number of times that $Keyword_i$ and $Keyword_j$ co-occur in the same image in the database. The Y-axis of Figure 8 denotes $\sum_{\forall i,j \; such \; that \; I_{i,j}=X} I_{i,j}$. The figure shows that most $I_{i,j}$s have small values. This allows us to drop $Cooccurring(Keyword_i, Keyword_j)$ if its corresponding value $I_{i,j}$ is less than a pre-set threshold. In our example database containing 15000 images, by dropping $Cooccurring(Keyword_i, Keyword_j)$ if its value is less than 3, we reduce the total number of $Cooccurring(Keyword_i, Keyword_j)$ entries by 92% while retaining 78% of the cooccurring object instances (i.e. $\sum I_{i,j}$).

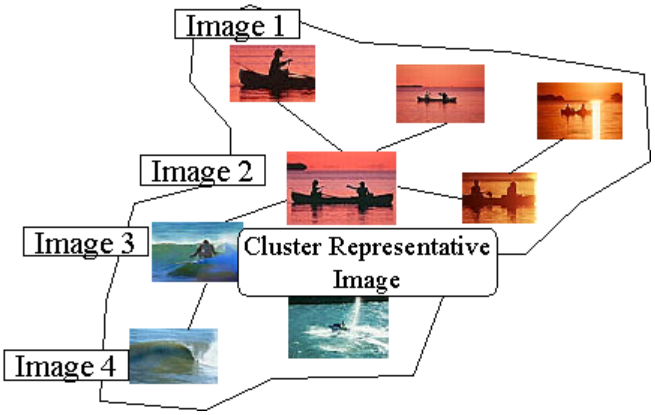

Fig. 9 Difficulty in finding cluster centers

This technique allows us to reduce cooccurring object entries substantially while keeping all necessary instances as indices to provide feedback.

4 Image Indexing and Selectivity Construction

In the previous section, we described the indexing schemes and selectivity construction for the semantic information contained within images. In this section, we present the indexing schemes and selectivity construction mechanisms we use for visual information, i.e., colors and shapes. During the image registration phase, the system extracts the visual characteristics of the images. Based on these visual characteristics, the system finds one or more clusters which matches the given image the best and places it in these clusters. The classification results, cluster centers (i.e. representative images) and the number of members in each cluster, are used as selectivity values. With such an approach, we provide a uniform framework of selec-

tivity for both semantics and visual characteristics for providing feedback or query processing.

4.1 Challenges in Image Clustering

Ideally, image selectivity can be calculated using the following steps (as how it is calculated in traditional text only databases): (1) cluster images based on colors and shapes into categories; say, $Cluster_1 \cdots Cluster_m$; (2) select an image I_i with representative colors and shapes for each cluster $Cluster_i$ where i is between 1 and m. The number of images in $Cluster_i$, S_i, can be viewed as selectivity for colors and shapes represented by I_i; and for a given image I, we can estimate the number of matching images by comparing colors and shapes between I and I_i. S_i is the estimated number of matching images if I matches with I_i above a pre-determined threshold for color and shape similarity.

In reality, however, it is difficult to select representative visual characteristics, especially shape, for clustering images. For example, Figure 9 shows a cluster of similar images. In this case, the images labeled *1* and *2*, and the images labeled *3* and *4* are similar based on colors while the images labeled *2* and *3* are similar based on shapes. In this case, it is hard to choose the representative image since similarity is not transitive in general. We may select the image labeled *2* as the representative image, it is not an ideal one. In many cases, a reasonable representative image for a cluster of images may not exist.

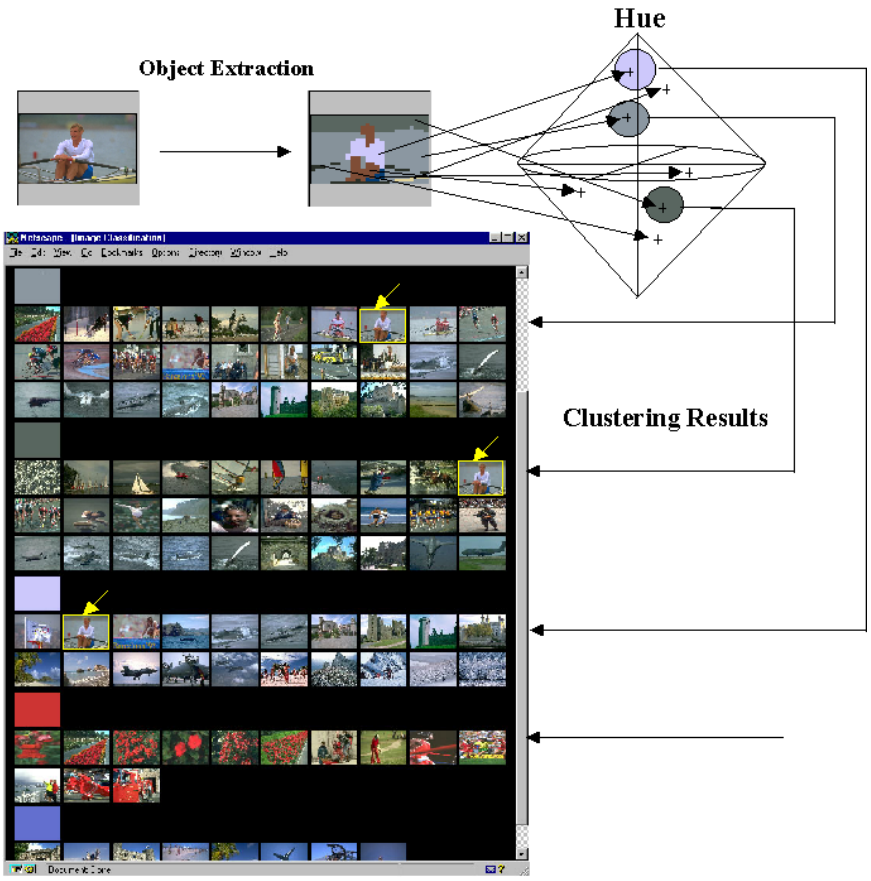

Fig. 10 Region/Color extraction, mapping, and image classification

To overcome these difficulties, we take an alternative approach as follows: (1) classify images based on colors and shapes *separately* while providing a uniform viewpoint in classification; and (2) synthesize, rather than select, representative images (i.e., template colors and shapes) to form clusters. Next, we present the classification schemes based on colors, followed by the schemes based on shapes.

4.2 Color-based Image Classification

Our system first extracts major colors from each image. In the current implementation, up to 8 major colors are extracted from each image although the number of major colors can be an arbitrary number. Our system maps each image color to HLS (Hue, Lightness, and Saturation) space and uses the locations of these points in the HLS space for classification. This is shown at the top of Figure 10. In this example, seven major colors are extracted from the image `man_boat.gif` and are mapped to the HLS space. In the bottom of Figure 10, we show three clusters in which `man_boat.gif` is classified based on color similarity. Note that `man_boat.gif` is highlighted in these clusters. There are total of seven clusters in which `man_boat.gif` is classified into. The number of members in a cluster for a given color is the selectivity value for such color.

For image retrieval, the user specifies colors and a threshold for color similarity. Based on the color clustering hierarchy, our system finds the cluster or clusters whose colors match with the specified colors. Figure 10 shows an example output for image retrieval based on the color clustering. Currently, 128 colors and black are used as templates in our implementation for image classification by color. The number of template colors required depends on the applications, color distributions, and complexity in image collection. The system administrator can control the maximum cluster diameter by adjusting the color similarity threshold.

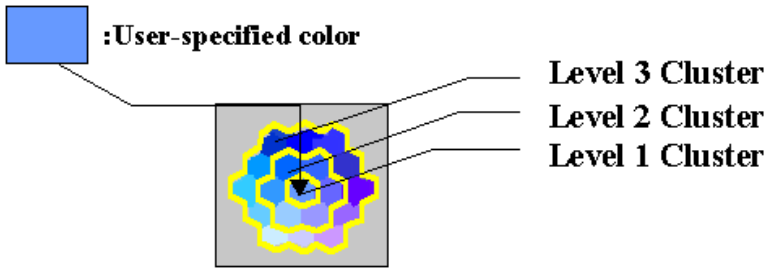

Fig. 11 Color-based cluster hierarchy

In Figure 11, the cluster diameter of the level 1 clustering is smaller than the cluster diameter of the level 2. As the cluster diameter gets larger, the precision of the classification decreases. The clustering scheme brings together those colors that are closer to each other to form higher levels of the hierarchy. This color classification hierarchy assists SEMCOG in the query-relaxation task by providing alternative colors. It also contributes to the query processing by providing color selectivities. Note the similarity between the hierarchical structure for colors in Figure 11 and the hierarchical structure for semantics in Figure 6. In Figure 11, we can view that there are seven *level 1* color clusters with the "*ISA_COLOR_OF level 2 color cluster*" relationship. Such a color clustering hierarchical structure is used for both query relaxation and progressive processing described later in Subsections 5.3 and 5.4.

4.3 Shape-based Image Classification

Shape similarity is rather hard and ambiguous to define. Our approach to shape-based image classification is to identify a set of template shapes and

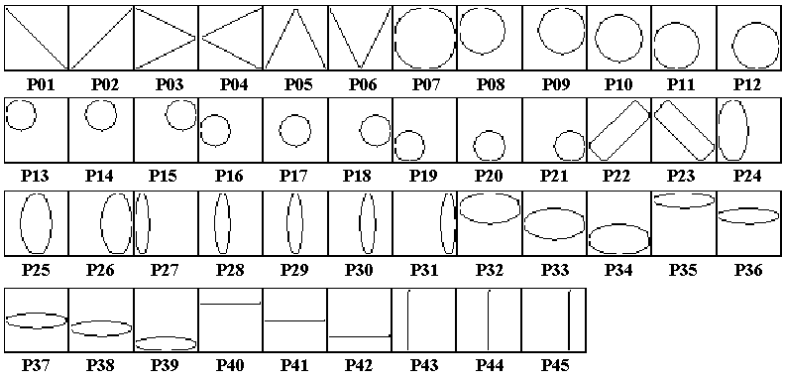

Fig. 12 Shape templates for image clustering

classify images to clusters based on similarity between the major shapes in images and the template shapes provided. In the current implementation, 45 template shapes are defined as shown in Figure 12. These template shapes are are given by the system designers based on applications. Major shapes are extracted from images as follows:

- For an image I, we match it with 45 template shapes,

$$T = \{template\_shape_i | 1 \leq i \leq 45\}.$$

  The similarity between the image I and $template\_shape_i$ is

$$shape\_similarity(I, template\_shape_i).$$

- The major shapes, T', of the image I is a subset of T, such that for every $template\_shape_j \in T'$, the value of $shape\_similarity(I, template\_shape_j)$ is greater than a threshold. The *primary shape* is defined as

$$template\_shape_{pr} \in T'$$

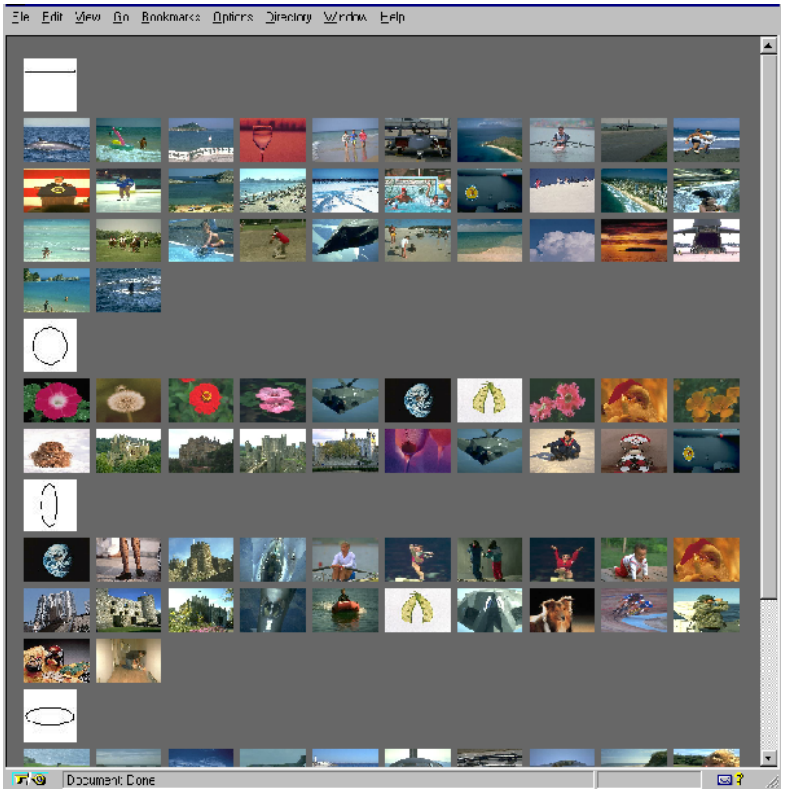

Fig. 13 Image classification based on shapes

such that

$$shape\_similarity(I, template\_shape_{pr}) =$$

$$max(\{shape\_similarity(I, template\_shape_j) | template\_shape_j \in T'\}).$$

Based on these template shapes, images are classified into up to 45 clusters. As was the case in color-based classification, each image may be assigned to multiple clusters. Examples of clustering results are shown in Figure 13.

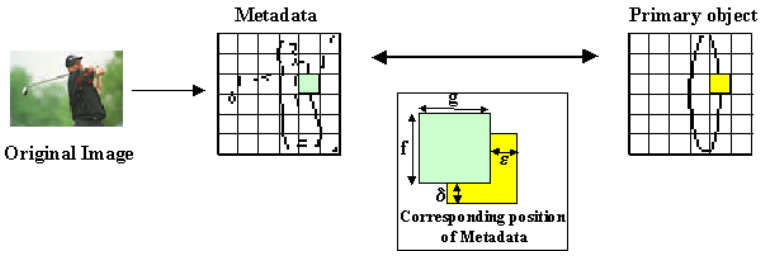

Fig. 14 Matching images with primary objects

4.4 Classification based on Primary Objects

The classification process based on primary objects involves two steps. In the first step, we calculate the similarity between the primary objects and image metadata. The final step is to categorize the images. Since the object in an image may be located in various places within the image, several sets of query images are created by shifting the location of the primary objects. Our boundary comparison algorithms consider shifting factors in matching boundaries of nearby objects. This can be used to reduce image matching time with a larger shifting step for boundary comparison globally. In our current implementation, shifted images are created for every 15classifies images based on the composition of the object, the system uses the primary objects without shifting in the image. We have designed the matching algorithms by considering the local correlation among the blocks of the image. The procedure is illustrated in Figure 14.

In this process, the system divides a target metadata $U_t = \{u_{ij}\}$ and a primary object $V = \{v_{ij}\}$ into several blocks (rectangular partitions or grid). Each local block size is $f \times g$. Each local block is shifted in 2-

dimensional (2D) direction near the same position of the metadata to find the best match position. Because of this 2D direction shifting, the system matches the object located in different positions. Even if the object in the metadata includes some noise and additional lines, or the size of the object is different, these errors may not influence any other local blocks because of the small matching size. We calculate the correlation ${}^{ab}C_{\delta\epsilon}$, between the local blocks, ${}^{ab}U$ and ${}^{ab}V$ by shifting ${}^{ab}V$ by δ vertically and δ horizontally.

$$ {}^{ab}C_{\delta\epsilon} = \sum_{r=fa}^{f(a+1)-1} \sum_{s=gb}^{g(b+1)-1} (\alpha U_{rs} \cdot V_{r+\delta,s+\epsilon} + \beta \overline{U_{rs}} \cdot \overline{V_{r+\delta,s+\epsilon}} \oplus \gamma U_{rs} + V_{r+\delta} s + \epsilon) \tag{1} $$

Here, coefficients α, β, γ are the control parameters used to estimate matching and mismatching patterns. We calculate the similarity values between primary objects and metadata C_t, as follows:

$$ C_t = \sum \sum max({}^{ab}C_{\delta\epsilon}) \tag{2} $$

If there is no boundary lines on the specific pixels in the query images, we can derive one of the two conclusions. Either no boundary line exists, or none have been specified for that region. For the classification based on the primary objects, we apply the second strategy. The system focuses on the basic lines of the primary object and ignores other parts. We normalize the similarity based on the pixel number of the lines in the primary object NL_p. We define the similarity values between the metadata and the primary

object S_t as follows:

$$S_t = \frac{C_t}{NL_p} \tag{3}$$

This algorithm evaluates the similarity using each line element as a unit. It is possible to evaluate the similarity even when the boundary lines are detected partially.

The next step in this process is to determine the category of the image. After the matching procedure, the maximum value of the similarity between every image created by shifting primary objects and the image is used to determine whether such an image contains the primary object. Thus our technique supports location-independent identification of primary object in an image. If the similarity between the primary object and the boundary image is greater than a certain threshold, the image is determined to contain the primary object and is classified into that primary object category. By adjusting the threshold value, the system designer can control the number of images classified under one primary object.

4.5 Creation of Statistics Using Image Classification

To store the statistics for estimating numbers of matching images, SEMCOG uses one table for the upper bounds and the other table for the lower bounds. In these tables, the rows and columns correspond to colors and shapes, respectively. At each slot in the table, there is an integer which denotes the number of images which contain the corresponding color and shape (Figures 15 and 16). Since we employ 129 template colors and 45 template shapes,

these two tables are 129 by 45. Note that in Figure 15, all major colors and shapes are used. On the other hand, in Figure 16, only the primary color and shape are used. Using these two tables, SEMCOG estimates the maximum and minimum number of matches for a given query image as follows:

- Calculation of maximum number of matches: As illustrated in Figure 15, we identify all the major shapes and colors in images. In the example of the Golden Gate Bridge image, `color04`, `color06`, `color34`, `shape03`, and `shape06` are extracted. We sum the values in the slots

  $(color04, shape03)$, $(color06, shape03)$, $(color34, shape03)$,

  $(color04, shape06)$, $(color06, shape06)$, and $(color34, shape06)$

  to find the maximum number of possible matches (i.e. 168). Note that this estimation is based on the assumption that the images in the result will correspond to one and only one shape-color pair in the image; otherwise, we would be over-estimating the number of matches. However, this assumption is valid for the calculation of the maximum number of matches.
- Calculation of minimum number of matches: The system identifies the primary color and shape from the given image and use the corresponding color-shape slot for the minimum number of matches. As shown in Figure 16, the system locates the value in the slot $(color06, shape03)$, 21 as the estimated minimum number of matches for the given image. This approach is based on the assumption that the given image has only one major shape (i.e. its primary shape). This is the lower bound of number

of matching images. Since there may be more then one equally likely color and shape combinations due to the fact that this is a similarity-based computation, the system may locate multiple slots. We use the minimum value among them as the minimum number of matches.

5 Using Statistics for Facilitating Multimedia Database Exploration

In Sections 3 and 4 we present the schemes for constructing selectivity statistics for semantics-based and media-based predicates. In this section, we describe how the two statistics are integrated and used for multimedia database exploration and automated query reformulation. We start with the functionalities of query verification followed by its underlying techniques.

5.1 Providing Feedback for Query Reformulation

We first use an example query, shown in Figure 17, involving both semantics-based and media-based predicates to illustrate the feedback provided by our system. The specific tasks performed by the query verifier are as follows:

- Estimating number of matching images and query predicate selectivity analysis: *Query verifier* provides the maximum, minimum, and estimated numbers of matches for the query, in the small panel located in the middle of the query interface. With this feedback, the user knows that if he/she should relax or refine the query. Users can view the detailed analysis of each query predicate by clicking the *Show* button on the

right side of the panel. The strictness of query criteria depends on its selectivity. This is shown in Figure 17(b). These selectivity values are pre-computed. Note that the statistics of selectivity values are used for the matching image estimation purpose and they are used for image retrieval when fast response is crucial.

- System feedback for semantics-based predicates: For each semantic condition, the user interface provides (1) a set of alternatives; (2) similarity between each alternative condition and initial condition; and (3) selectivity of the alternative condition. Figure 17(c) shows system feedback for the conditions `is(man)`. The system also provides feedback for object co-occurrence relationship. Figure 17(f) shows system feedback on objects which are in the same image as `man`. With this information, the user knows not only the fact some images do contain both `man` and `transportation` (i.e. car and bicycle) but also the distribution and selectivity of objects which are in the same images with a man, such as *car*, *bicycle*, *woman*, and *house*.
- System feedback for media-based predicates: For each visual-based condition, the user is provided with alternative colors and shapes. Figures 17(d) and 17(e) shows the system feedback for alternative colors and shapes. The feedback also includes similarity between the alternative colors and shapes and the colors and shapes of *car.gif* and their corresponding selectivity. This selectivity is pre-computed through image classification by color and shape as described in Section 4.

5.2 Estimating the Number of Matches for Feedback Generation

In Figure 17, we illustrate the query verification functionalities. These functionalities are supported using multimedia content selectivity statistics to avoid expensive query processing. In this subsection, we describe these schemes in detail.

Let us assume that a user specifies a query to retrieve images containing at least two objects: `man` and an object whose visual characteristics is similar to `car.gif`. We assume that the multimedia content statistics are as follows:

- The total number of images is 50.
- The images in the database contain at most two objects. We simplify this example for ease of illustration.
- The number of matches for the conditions

  containing two objects, *is(man,X)*, and *i_like(car.gif,Y)*

  are known to be 66, 27, and 18 respectively.
- The total number of objects is 83.

Since, the maximum number of objects in a given image is two, there is no difference between the number of images which contain at least two object and the number of images which contain exactly two objects. Note, on the other hand, that in our system, there is a difference between the number of images which contains two objects and the number of matches for the query "images containing two objects". The number of matches is twice the number of images since there are two matches for each image.

Therefore, there are 33 (i.e. 66/2) images containing exactly two objects and 17 (i.e. $50 - 33$) images containing exactly one object.

Using the above information, we compute the expected, maximum, and minimum number of matches for the above query as follows:

5.2.1 Expected Number of Matches Assuming that the semantics of image objects are uniformly-distributed, the estimated number of matches, based on probabilities, is

$P(\textit{there are two objects in the image}) \times$

$P(\textit{one object is man and the other object looks like car.gif} \mid$

$\textit{there are two objects in the image}).$

To solve the above probability, we need a conditional probability for the second term. However, we do not have this information in the given set of selectivity values. We calculate its approximated value as $2 \times selectivity(man) \times selectivity(car.gif)$. Hence, the approximate probability, for our example, is $\frac{33}{50} \times 2 \times \frac{27}{83} \times \frac{18}{83} = 0.093$, or the expected number of images satisfying the query is $50 \times 0.093 = 4.65$.

5.2.2 Minimum and Maximum Numbers of Matches We denote the following selectivity functions: (1) the image selectivity function for the condition *containing exact N objects* as $ObjectContainmentImageSelectivity(N)$; (2) the object selectivity function for the semantics-based condition S as $ObjectSemanticsSelectivity(S)$; and (3) the object selectivity function for the

cognition-based condition C as $ObjectCognitionSelectivity(C)$. In this example,

- $ObjectContainmentImageSelectivity(2)$,
- $ObjectContainmentImageSelectivity(1)$,
- $ObjectSemanticsSelectivity(is(man))$, and
- $ObjectCognitionSelectivity(i-like(car.gif))$

are required. The maximum number of matches is the minimum value of the object semantics selectivity; 18 in this example. In Figure 18, three bars are used to represent selectivity for $is(man)$, $i\_like(car.gif)$, and estimated number of matching images for this query. The maximum number of matches is illustrated in Figure 18(a), where the overlap area of the bars for $is(man)$ and $i\_like(car.gif)$ is maximized.

Assuming that the objects with certain semantics are uniformly distributed in images, the minimum number of matching images can be computed as follows:

$\bullet (\sum ObjectSemanticsSelectivity + \sum ObjectCognitionSelectivity)$

$$- \sum_{N=1}^{2} (ObjectContainmentImageSelectivity(N))$$

if $\sum_{N=1}^{2} ObjectContainmentImageSelectivity(N) <$

$$(\sum ObjectSemanticsSelectivity + \sum ObjectCognitionSelectivity).$$

This is illustrated in Figure 18(b) as if the total number of images was 37, instead of 50.

$\bullet$ 0

if $\sum_{N=1}^{2} ObjectContainmentImageSelectivity(N) \geq$

$$(\sum ObjectSemanticsSelectivity + \sum ObjectCognitionSelectivity).$$

This is illustrated in Figure 18(c).

The minimum number of matching images for this example is 0 because

$ObjectContainmentImageSelectivity(2)+$

$ObjectContainmentImageSelectivity(1) = 33 + 17 >$

$ObjectSemanticsSelectivity(is(man))+$

$ObjectCognitionSelectivity(i - like(car.gif)) = 27 + 18$

Note that without the assumption of the objects with certain semantics are uniformly-distributed in images, the minimum number of matching images would be either 1 or 0; which is not meaningful for users. We illustrate this in Figures 18(d) and 18(e). Based on the above formula, the system calculates and provides users with the feedback on the expected, maximum, and minimum numbers of matching images as 4.65, 18, and 0 respectively.

To generalize the formula presented above, for a query associated with M objects and M conditions on databases with images containing at most two objects, the minimum number of matching images can be computed as follows:

- $(\sum ObjectSemanticsSelectivity + \sum ObjectCognitionSelectivity)$

  $- \sum_{N=1}^{M}(ObjectContainmentImageSelectivity(N))$

  if $\sum_{N=1}^{M} ObjectContainmentImageSelectivity(N) <$

  $(\sum ObjectSemanticsSelectivity + \sum ObjectCognitionSelectivity).$

- 0

  if $\sum_{N=1}^{M} ObjectContainmentImageSelectivity(N) \geq$

$$(\sum ObjectSemanticsSelectivity + \sum ObjectCognitionSelectivity).$$

5.3 Usage in Automated Query Relaxation

In Figure 17, we illustrate how the system use statistics and indices of visual and semantic characterisitcs of image database contents to assist users in reforumulate queries in an interactive mode fashion. Alternatively, if users prefer, the system can automatically and incrementally reformulate queries until there are sufficient results. The system chooses the most relevant alternative conditions in respect to the original condition. Such relevancy is calculated during the semantics indexing (Section 3) and image clsutering (Section 4) phases. The relevance of two semantics-based conditions is calculated using the formula presented in Section 3.2. The relevance of two media-based conditions is calculated based on feature set (e.g., color histogram, texture, shape, etc) of two conditions. As summaried in Section 2, our modeling and language are object-based and thus our system can support query reformulation at a finer granularity. The query language used in SEMCOG can be viewed as a logic-based language consisting of a single rule of the form

$$Q(Y_1, \ldots, Y_n) \leftarrow D_1 \wedge D_2 \wedge \ldots \wedge D_d.$$

where $Y_1, \ldots, Y_n$ are free variables and $D_1, .., D_d$ are disjunctions that consist of predicates ,$P_1, \ldots, P_p$, their negations, $\neg P_1, \ldots, \neg P_p$, variables, X_1, X_2,. . ., X_n, and constants. There are d possible query reformulation options

by changing only one condition (i.e. predicate). We denote the query whose m th predicate (i.e. $Predicate\_from$) is reformulated to $Predicate\_to$ as

$$Q_m(Y_1, \ldots, Y_n) \leftarrow D_1 \wedge D_2 \wedge \ldots D_{m-1} \wedge Predicate\_to \wedge D_{m+1} \ldots \wedge D_d.$$

Note that for the results of $Q_m(Y_1, \ldots, Y_n)$ ranked with a score $S(Q_m)$, they should be ranked as

$$similarity(Predicate\_from, Predicate\_to)) * S(Q_m)$$

for the original query Q. Thus, the value of

$$similarity(Predicate\_from, Predicate\_to)$$

can be viewed as *penalty* for the ranking scores of results of reformulated queries. For query reformulation involved k predicates, the *penalty* for the ranking scores are

$$\prod_{i=1 \ldots k} similarity(Predicate\_from_i, Predicate\_to_i).$$

This value can be used to show the users for visualizing how different the alternatives are from the original query criteria, or they can be used for automatic query relaxation as discussed next.

Let us assume that a user submits a query to retrieve images containing a man and a car. If there are no matches or very few matches, SEMCOG can automatically reformulate the initial query. In the given example, the system may alter the query to retrieve images which contain a man with a bus, a woman with a car, or a woman with a bus.

Let us assume that the similarity values among related predicates are as follows:

- Similarity between *is(car)* and *is(bus)* is 0.8
- Similarity between *is(man)* and *is(woman)* is 0.7.

The matching value for the images containing a car and a man is 1 (these images match the original query criterion perfectly). On the other hand, the results for the relaxed queries are subject to a "penalty". The images generated by the reformulated queries and the relevance of the results are as follows:

- Images containing a man and a bus: 0.8
- Images containing a woman and a car: 0.7
- Images containing a woman and a bus: 0.56 (i.e. 0.8×0.7). Note that we multiply the degrees of relevance because these two conditions are associated by an `AND` operator.

5.4 Progressive Processing

In this section, we introduce multi-level classification and apply it for progressive query processing, which can be viewed as a more intelligent way for automated query relaxation presented in Subsection 5.3. Figure 19 shows the multi-level classification based on similarities. In Figure 19, the system defines multiple levels of clustering based on a primary object: *Class1* is more relevant than *Class2* and *Class2* is more relevant than *Class3*. Using this hierarchy, the system can perform a proximity search, which is suitable

for navigation, by join operations on the image IDs of each clusters. In the given example, the user specifies a query image with the two primary objects, *Shape30* and *Color48*. Images in the first class for both *Shape30* and *Color48* clusters have the highest scores (i.e. first rank candidates). Images in the first class for *Shape30* and the second class for *Color48* and the image in the second class *Shape30* and the first class for *Color48* have the second highest scores (i.e. second rank candidates). With this property, for a query to retrieve top K results, our system can generate candidates in the first classes, followed by generating candidates in the second classes progressively until the total number of candidates reach K. Similarly, if the user specifies that the shape is more important than the color in image matching; the system generates results in $(class1, class2)$ before $(class2, class1)$.

5.5 Evaluation of the Image Retrieval by Clustering

We have presented many usage for statistics of multimedia contents. Many require user usability, which is in the field of CHI. Here we present an experimental study to evaluate our classification technique in the aspects of the reduction of search space versus recall ratio. The ideal classification is that the search space can be reduced while the recall is maintained at a satisfactory level.

In this experiment, 15032 photographs are used. We first classify images using 45 shape primary objects and 129 color primary objects. Note that these primary objects are defined for a general-purpose image collection

and are not specialized for this particular image set. The size of clusters can be adjusted by the threshold values for the similarity between images and primary objects. The larger is the similarity threshold value used, the more search space can be reduced. However, there is a higher chance that the recall may drop. In this experiment, we set the threshold value as 0.6.[2] We issued 6928 queries to search (1) all 15032 images, and (2) only the images classified into the same categories as the query image. We then compared the top 5 query results for two schemes. If these two schemes generate the same query results, the scheme which searches only clusters gains substantial search space reduction while maintaining the perfect recall. If the scheme which searches only clusters generates only 4 out of 5 images generated by the scheme which matches with all 15032 images, the recall is 80%.

In this experiment, our approach of image retrieval based on clustering can reduce the search space by 76% while maintaining 73% of initial recall of performing image matching on all 15032 images. We do find 27% loss of recall in our experiments. Since our system performs image matching based on clustering mainly for the purpose of fast response time for navigation, in which fast response time is more important than perfect recall. For image retrieval by query, the system can perform image matching on all 15032 images. Our approach provides more flexible image database exploration.

[2] The similarity values are normalized scores between 0 and 1. The similarity calculation function and normalization procedure are described in [3].

6 Related Work

One approach such as [7,8], is to create feature vectors for representing the visual characteristics. Similarity between two images is determined by calculating the distances between the vectors of a query image and each of the images in the database. The attribute values are assigned to the whole images. Each attribute values has less semantic meanings and it is difficult for users to understand the contribution of the values. It is very hard to integrate these visual characteristics with the semantics tightly. This whole image-based image modeling is not suitable to refine the parameters based on the retrieval

Another approach is to extract regions or blocks from images based on color continuities and shapes. [9–11] Image similarity is calculated using the spatial information of these regions. However, they mainly focus on the image matching capability and not focus on the integration with the semantics. [12] extracts object from images and store them as Semcons. Users can navigate based on both semantic and visual characteristics at the object level. However, this method assumes that the object extraction and semantic assignment are performed completely manually. On the other hand, SEMCOG can perform object extraction automatically using region division methods based on colors and shapes.

Research in [7–12] mainly addresses the matching/retrieving capabilities and does not address the feedback from the query and refinement of the query based on the system feedback. SCORE[13] focuses on the use

of a refined ER model to represent the contents of pictures. It calculates the similarity values based on these ER representations. SCORE manages the similarity based on the semantics, however, does not support image-matching capability.

Many techniques have been proposed for image clustering. [14] focuses on the color information. They extract color values from images and map them onto the HLS color spaces. Based on the resulting clusters, users can access image directories or filter out of the images for searching. [15] focuses on the clustering of shape similarity. Based-on the multi-scale analysis, it extracts the hierarchical structure of shape. According to this hierarchical structure, it attempts to provide more effective search capabilities. Many of the methods described in "Image Retrieval", also propose the classification methods using their matching criteria. QBIC [16] clusters images using feature vectors. [11] extracts the objects from the images based on the color and texture. Using the combination of extracted objects (BLOB) and their attributes (top two colors and texture) they try to categorize the images into several groups.

Most of existing work focuses on image clustering and retrieval tasks. Our work aims at providing more efficient multimedia database exploration in terms of usability, system load, and response time. To support such utility, we present a uniform framework to represent the statistics for both textual data (i.e. semantics) and visual characteristics. Our work is unique and novel in these respect.

7 Conclusions

Retrieval in multimedia databases is different from traditional database query processing since it is based on similarity calculation for media and semantics comparisons. In this paper we present a system SEMCOG which supports a new approach to media retrieval by systematically facilitating users to reformulate queries. Our approach is to use semantic and visual indices and selectivity values to provide various system feedbacks to assist users in reformulating queries. We describe a uniform framework to represent statistics for both semantics and visual characteristics of images. Based on such statistics, our system also performs automated query relaxation by choosing the most relevant colors, shapes, or object semantics to reformulate users' queries incrementally until there is a reasonable number of candidates in the answer set. We also present experimental results of evaluating the efficiency of image retrieval based on clustering. Additional evaluation related to usability in the scope of CHI is left as future work.

The contributions and novelty of this work include: (1) a framework for multimedia database exploration through computer-human interaction based query cycle; (2) techniques for generating and maintaining selectivity values for both semantics- and visual-based predicates and using them for estimating query results without expensive query processing; (3) schemes for interactive and automated incremental query reformulation; and (4) schemes supporting progressive query processing for image retrieval.

References

1. Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Welsey Publishing Company, Inc., 1989.

2. Karen Sparck Jones and Peter Willett (editors). *Readings in Information Retrieval.* Morgan Kaufmann, San Francisco, California, USA, 1997.

3. Wen-Syan Li and K. Selçuk Candan. SEMCOG: A Hybrid Object-based Image Database System and Its Modeling, Language, and Query Processing. In *Proceedings of the 14th International Conference on Data Engineering*, Orlando, Florida, USA, February 1998.

4. Wen-Syan Li, K. Selçuk Candan, Kyoji Hirata, and Yoshinori Hara. Facilitating Multimedia Database Exploration through Visual Interfaces and Perpetual Query Reformulations. In *Proceedings of the 23th International Conference on Very Large Data Bases*, pages 538–547, Athens, Greece, August 1997. VLDB.

5. G. A. Miller. WordNet: A Lexical Databases for English. *Communications of the ACM*, pages 39–41, November 1995.

6. R. Richardson, Alan Smeaton, and John Murphy. Using Wordnet as a Knowledge base for Measuring Conceptual Similarity between Words. In *Proceedings of Artificial Intelligence and Cognitive Science Conference*, Trinity College, Dublin, 1994.

7. J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Jain, and C.-F. Shu. The Virage Image Search Engine: An Open Framework for Image Management. In *Proceedings of the SPIE - The International Society for Op-*

tical Engineering: Storage and Retrieval for Still Image and Video Databases IV, San Jose, CA, USA, February 1996.

8. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, and Q. Huang. *Intelligent Multimedia Information Retrieval, edited by Mark T. Maybury*, chapter I: Query by Image and Video Content: The QBIC System. MIT Press, 1997.
9. John R. Smith and Shin-Fu Chang. VisualSEEk: a Fully Automated Content-based Image Query System. In *Proceedings of the 1996 ACM Multimedia Conference*, pages 87–98, Boston, MA, 1996.
10. W. Ma and B.S. Manjunath. NeTra: A Toolbox for Navigating Large Image Databases. In *Proceedings of the 1997 IEEE ICIP Conference*, pages 568–571, 1996.
11. C. Carson, S. Belongie, H. Greenspan, and J. Malik. Color- and texture-based image segmentation using EM and its application to image querying and classification. Submitted to *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1997.
12. William I. Grosky. Managing Multimedia Information in Database Systems. *Communications of the ACM*, pages 72–80, December 1997.
13. A. Prasad Sistla, Clement Yu, Chengwen Liu, and King Liu. Similarity based Retrieval of Pictures Using indices on Spatial Relationships. In *Proceedings of the 1995 VLDB Conference*, Zurich, Switzerland, September23-25 1995.
14. Kyoji Hirata and Yoshinori Hara. The concept of media-based navigation and its implementataion on hypermedia system 'miyabi'. *NEC Research & Development*, 35(4):410–420, October 1994.
15. A. Del Bimbo and P. Pala. Shape Indexing by Structural Properties. In *Proceedings of the 1997 IEEE Multimedia Computing and Systems Conference*, pages 370–378, Ottawa, Ontario, Canada, June 1997.

16. Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. Query by Image and Video Content: The QBIC System. *IEEE Computer*, 28(9):23–32, September 1995.

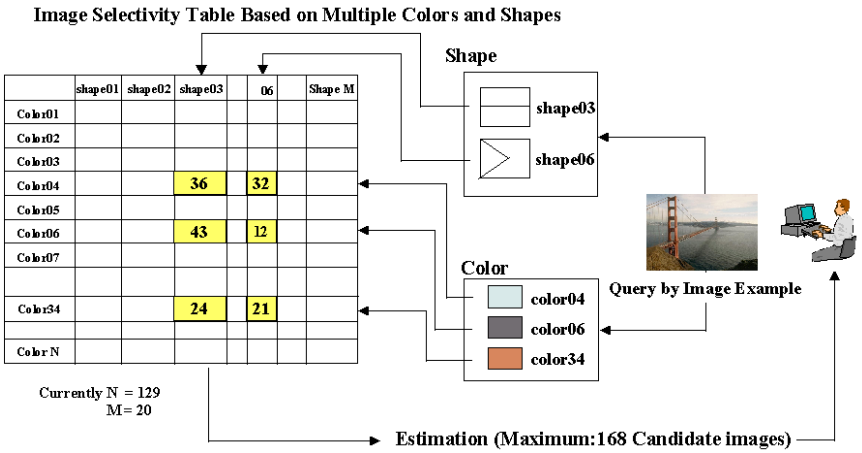

Fig. 15 Upper bound of estimated number of matching images

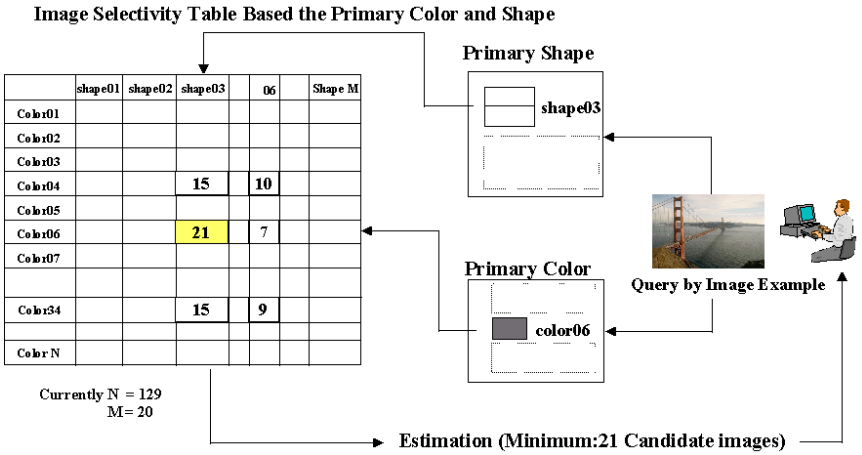

Fig. 16 Lower bound of estimated number of matching images

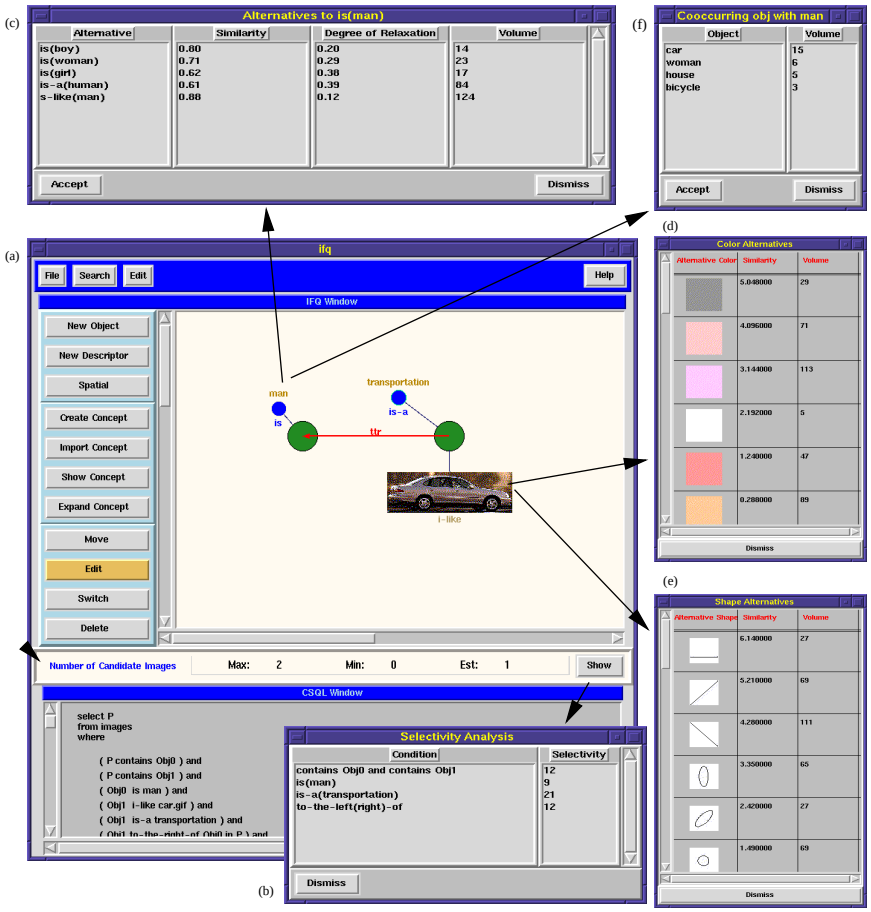

Fig. 17 Multimedia database query specification with system feedback

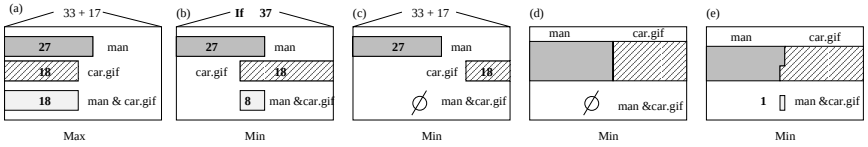

Fig. 18 Maximum and minimum numbers of matches

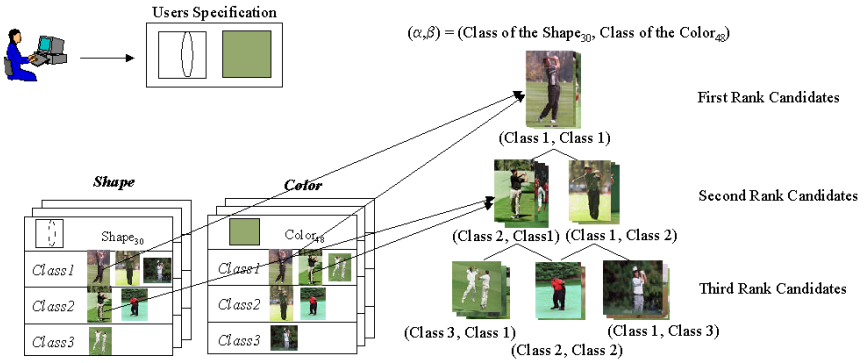

Fig. 19 Progressive Query Processing