SEMCOG: An Object-based Image Retrieval System and Its Visual Query Interface

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

C&C Research Laboratories, NEC USA, Inc.
110 Rio Robles, San Jose, CA 95134
Email: {wen,candan,hirata,hara}@ccrl.sj.nec.com

1 Introduction

Image retrieval is a key issue in many image database applications. Major approaches include queries by image examples (cognition-based approach) and query by semantics (semantics-based approach).

The semantics-based approach is good at image retrieval based on image semantics. However, this approach has low visual expressive capability. Since images are visual, it is hard to describe in detail using text alone.

The cognition-based approach has an advantage of using visual examples. However, one disadvantage of using the cognition-based approach alone is its low precision. This is because users' drawings are usually not precise enough and current image recognition technique still has its limits. For example, if a user wants to retrieve images containing a dog and provides a drawing of a dog, currently systems of cognition-based approach can only return many images with some animals which look like dogs, such as cats and tigers. Another weakness of this approach is that it can not support queries on generalized concepts, such as transportation and appliances. For example, if a user wants to retrieve all images containing any type of *appliances*, the user must provide drawings for all possible appliances, such as TVs, radios, computers, and etc. This is not practical.

As to the query specification flexibility, SQL-based languages provide support for specification of queries using only text. However, it is important to support query specification by visual examples. Existing "visual" interfaces are either GUIs to pose textual queries or drawing windows to provide sketches or examples of *whole* images. These interfaces do not give users flexibility in using combinations of semantics and visual expressions to pose queries. These GUIs do not really visualize target images of users' query specifications.

2 SEMCOG Approach

We argue that image retrieval based on either approach *alone* is not sufficient in terms of modeling and query specification flexibility. We also argue that a visual query interface which is capability of visualizing target images is essential. SEMCOG[1] (SEMantics and COGnition-based image retrieval) aims at integrating semantics and cognition-based approaches to give users a greater flexibility to pose queries. SEMCOG's image matching is based on objects in the images rather than the whole images. In SEMCOG, a query "Retrieve all images in which there is a man to the right of a car and the man looks like this image" can be posed using combinations of semantics and visual expressions. The queries are posed in the way of specifying image objects and their layouts using a visual query interface, IFQ (In Frame Query), rather than complicated multimedia database query languages.

The user's query can be simplified as a mental model shown at the top of Figure 1. She then specifies her mental model using IFQ using combinations of visual examples or semantics. IFQ window shows a query "Retrieve all images in which there is a person who is to the right of an object that looks like this image (a computer)". An actual window dump of IFQ for this query is shown in the middle of Figure 1. This specification is then translated (by IFQ) to a CSQL (Cognition and Semantics-based Query Language) query, a SQL-like query language used in SEMCOG. In this example, three types of queries are involved: semantics-based, cognition-based, and scene-based. The results for this example contain two images. Please note that *person* has been relaxed to *man* and *woman* accordingly and an image contains a computer and a book is also included. Their ranking is calculated by matching objects and their spatial relationships (layout) with users' query.

Through integration of semantics and cognition-based approaches of image retrieval, SEMCOG gives users a greater flexibility to pose queries so that queries in CSQL

†This work was performed when the author visited NEC, CCRL.

SIGMOD '97 AZ,USA

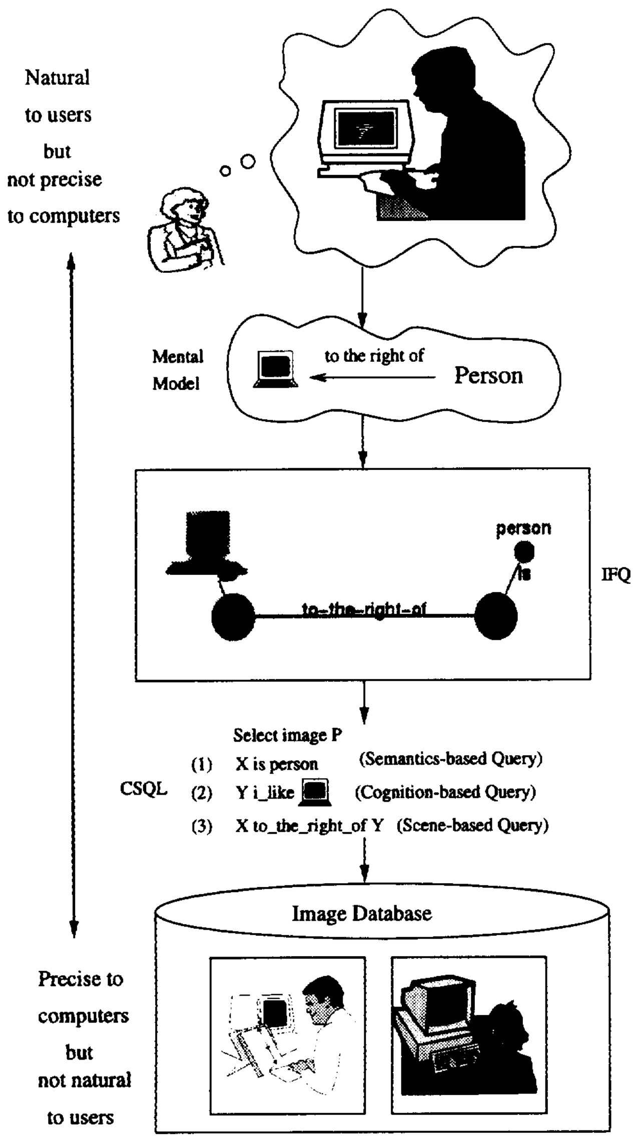

Figure 1: Example Query in SEMCOG

can represent users' candidate images better. As a result, the query specifications are more precise to database systems. Another advantage is that in the query specification process, users do not need to be aware of schema design and implementation of the image database as well as query language syntax since queries are generated by IFQ automatically.

3 System Architecture

SEMCOG architecture contains five components as shown in Figure 2. Their functionalities are as follows:

The *Facilitator* coordinates the interactions between components of SEMCOG. It forwards image matching related tasks to the *Cognition-based Query Processor* and non-image matching tasks to the *Semantics-based Query Processor*. One advantage of assigning these tasks to the *Facilitator* because the *Facilitator* has more complete knowledge of the query execution statistics and it can provide a globally optimum query processing.

COIR (Content-Oriented Image Retrieval)[2] is an objec based image retrieval engine based on colors and shapes. We use it as the *Cognition-based Query Processor* in SEMCOG. The main task of the COIR is to identify image regions based on pre-extracted image metadata, colors and shapes. Since an object may consist of multiple image regions, COIR consults the ***image component catalog*** for matching image objects.

When an image is "registered" at SEMCOG, the *Image Semantics Editor* interacts with COIR to edit the semantics of an image and objects in the image. The *Image Semantics Editor* then stores the image, semantics, and image metadata to the database.

The *Terminology Manager* maintains a terminology base which is used for query relaxation. For example, a user may submit a query as "Retrieve all images containing an *appliance*." Since *appliance* is a generalized concept rather than an atomic term, the *facilitator* consults the *Terminology Manager* to reformulate the query. Existing dictionaries, such as Wordnet, can be employed to build a terminology base.

The *Semantics-based Query Processor* performs queries concerning image semantics. The image semantics required for query processing is generated during the image registration. The semantics-based query processing is the same as traditional query processing on relational DBMSs.

4 Query Language

CSQL, a SQL-like language, is the underlying query language used in SEMCOG. SQL is augmented by adding predicates which are capability of handling multimedia data. These predicates extend the underlying database system to a multimedia database system. These predicates defined in CSQL include: (1) *Semantics-based*: ***is*** (e.g. man vs. man), ***is_a*** (e.g. car vs. transportation, man vs. human), and ***s_like*** for "semantics like" (e.g. car vs. truck); (2) *Cognition-based*: ***i_like*** for "image like" that compares visual signatures of two arguments and ***contains***; (3) *spatial relationship-based*: *to_the_right_of*, *to_the_left_of* and etc.

5 Query Interface

IFQ is an attempt to bridge the gap between precise computer languages and users mental models. IFQ is a visual, rather than "graphical", query interface which allows users to input keywords, concepts, semantics, image examples, sketches, and spatial relationships. IFQ can visualize target images as query specification process progresses.

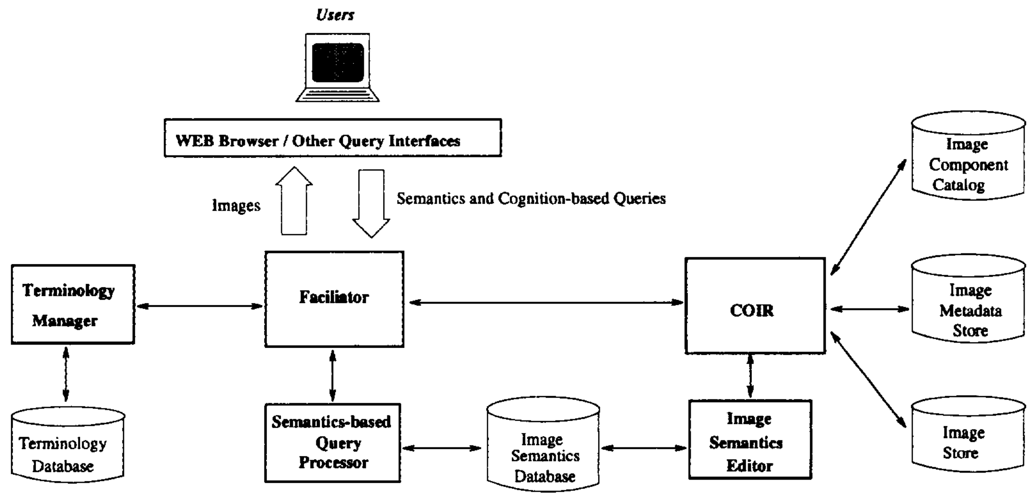

Figure 2: System architecture of SEMCOG

The query specification process in IFQ consists of three steps: introducing image objects, describing them, and specifying their spatial relationships. In IFQ, objects are represented as bullets and *descriptors*, represented as small bullets, attached to these objects describe their properties.

Figure 3 shows a query "Retrieve all images in which there is a man to the right of a car and he looks like this image" posed using IFQ. The IFQ query is posed as follows: The user introduces the first object in the image. and then further describes the object by attaching "i_like < *image* >" and "is *man*" descriptors. After a user specifies an image path or provides a drawing, the interface automatically replaces the descriptor with the thumbnail size image the user specifies. Then, the user introduces another object and describes it using the "is *car*" descriptor. Finally, the user describes the spatial relationship between these two objects by drawing a line, labeled by *to-the-right-of*, from the man object to the car object.

While user is specifying the query using IFQ, the corresponding CSQL query is automatically generated in the CSQL window. Users can pose queries simply by clicking buttons and dragging and dropping icons representing entities and descriptors. Figure 4 shows the result for the query in Figure 3, including a thumbnail size image and its ranking. Users can click on any thumbnail image to see the real image as shown on the right side.

IFQ also provides an *arrange* function. IFQ checks the matching between the layout specifications provided by the user and the actual layout on the screen. If there is a mismatch, IFQ rearranges the query objects on the screen according to the query specifications. Another functionality for increasing the perceptual qualities of IFQ is *iconize*. *Iconize* replaces the semantics terms on the IFQ window with the corresponding icons. As shown in Figure 5, IFQ replaces *IS_A transportation* and *IS man* with corresponding icons.

In many cases, users do not have specific target images in mind. Users want to extract and browse semantic and information about image objects. IFQ also supports interactions through specifying unbound descriptors and relaxing conditions, such as *IS-A* and S-Like predicates. In Figure 5, the user relaxes the condition of being a *car* to "being a kind of transportation". *IS_A transportation* is specified instead. The user can further introduces an unbound descriptor *out4* to check the actual semantics. The result given in Figure 5 shows two candidate images including an image containing a bus. This image has a lower ranking, because the human in the image can not be identified as men.

6 Conclusions

SEMCOG is currently being implemented on top a deductive database and a commercial object-relational DBMS. We have shown the design of SEMCOG and its query interface. The novelty of our work includes: (1) Object-based image retrieval rather than a *whole* image; (2) queries using combinations of *semantics* and *visual* examples; and (3) a visual query interface and query generator.

References

[1] Wen-Syan Li, K. Selçuk Candan, and K. Hirata. SEMCOG: An Integration of SEMantics and COGnition-based Approaches for Image Retrieval. In *Proceedings of 1997 ACM Symposium on Applied Computing Special Track on Database Technology*, San Jose, CA, USA, February 1997.

[2] Kyoji Hirata, Yoshinori Hara, H. Takano, and S. Kawasaki. Content-Oriented Integration in Hypermedia Systems. In *Proceedings of 1996 ACM Conference on Hypertext*, March 1996.

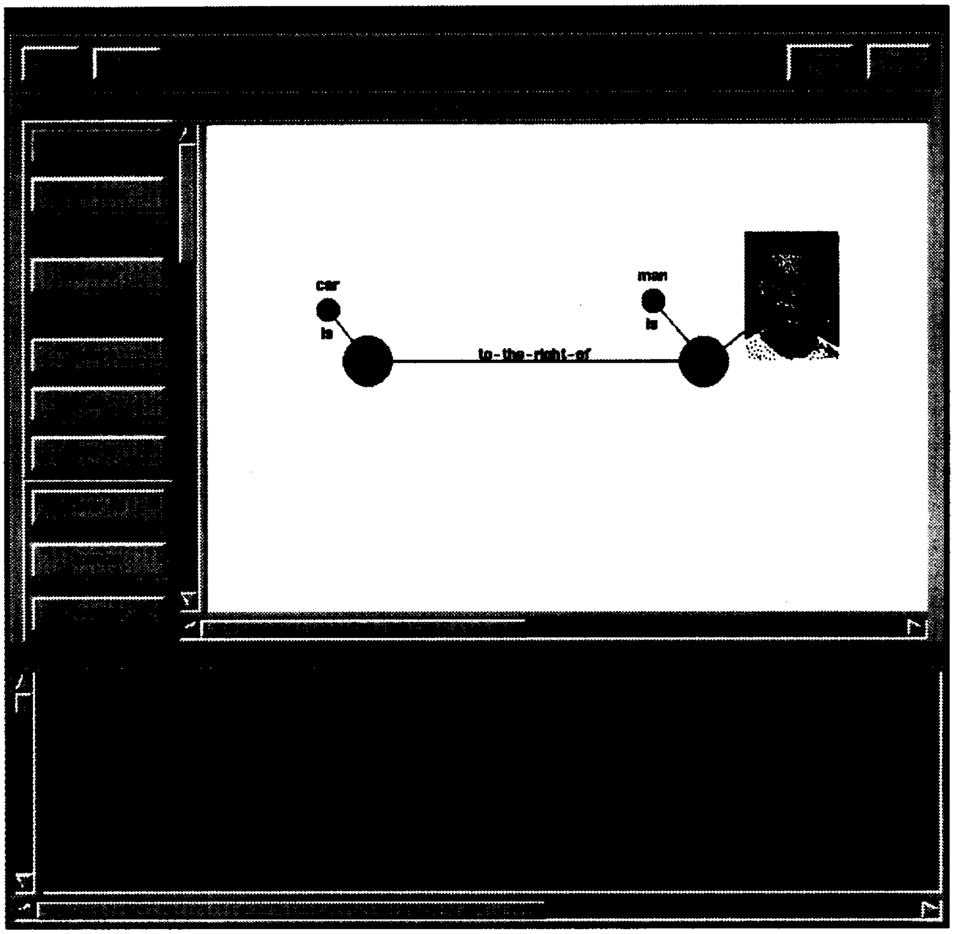

Figure 3: IFQ Specification Window (top) and CSQL Generating Window (button)

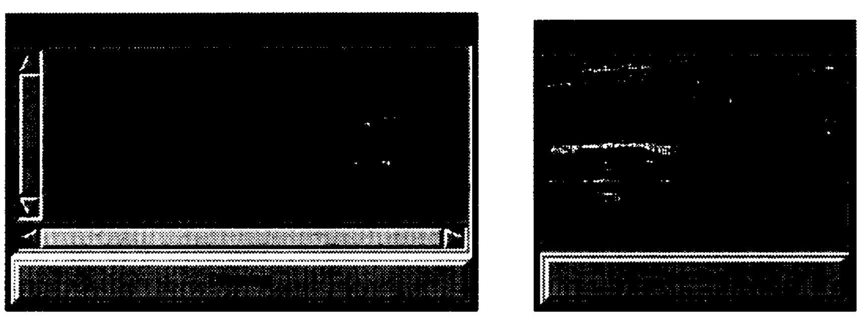

Figure 4: Query Result and Image Retrieved for the Query in Figure 3

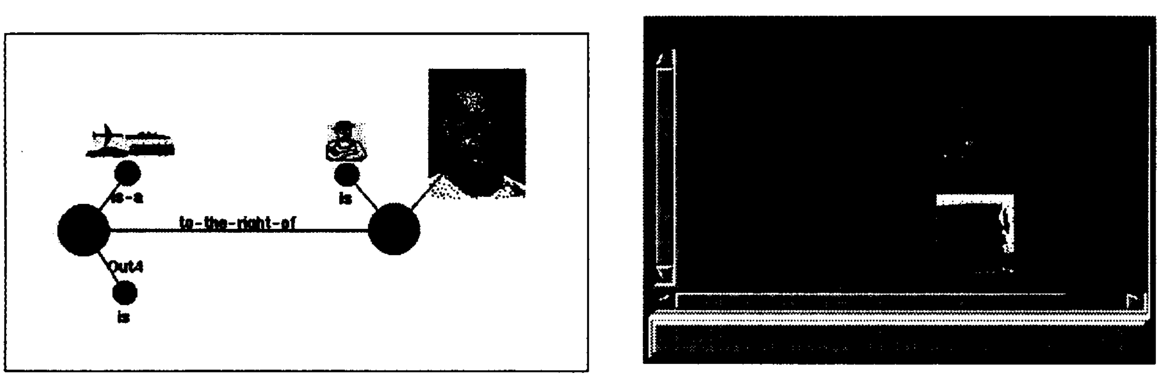

Figure 5: Relaxed Query for Extracting Semantics and its Result