A Three-Dimensional Iconic Environment for Image Database Querying

Alberto Del Bimbo, *Member, IEEE,* Maurizio Campanai, *Member, IEEE,* and Paolo Nesi, *Member, IEEE,*

Abstract— **Retrieval by contents of images from pictorial databases can be effectively performed through visual icon-based systems. In these systems, the representation of pictures with 2D strings, which are derived from symbolic projections, provides an efficient and natural way to construct iconic indexes for pictures and is also an ideal representation for the visual query. With this approach, retrieval is reduced to matching two symbolic strings. However, using 2D-string representations, spatial relationships between the objects represented in the image might not be exactly specified. Ambiguities arise for the retrieval of images of 3D scenes. In order to allow the unambiguous description of object spatial relationships, in this paper, following the symbolic projections approach, images are referred to by considering spatial relationships in the 3D imaged scene. A representation language is introduced that expresses positional and directional relationships between objects in three dimensions, still preserving object spatial extensions after projections. Iconic retrieval from pictorial databases with 3D interfaces is discussed and motivated. A system for querying by example with 3D icons, which supports this language, is also presented. The system fully allows the user to experience the sensation of directly interacting with reality with no intermediate mental processing.**

Index Terms— **Iconic retrieval, image databases, image representation languages, query by example, symbolic projections, 3D visual interfaces, visual languages.**

I. Introduction

RETRIEVAL of images from pictorial databases is typically performed on the basis of image contents, namely object types and attributes, and spatial relationships between objects. Different approaches have been proposed in the literature for supporting this kind of retrieval. These include query by pictorial example (QPE) [1], [2], where the user formulates a query by example using a schema of pictorial data in graphic and tabular form; query by visual example (QVE) [6], where similarity retrieval is supported by allowing the user to draw a rough sketch of image contents to be used as a visual key; relational nonprocedural SQL-like query languages [3], [4]; retrieval based on semantic significance [5]; and visual icon-based systems [7]. In visual icon-based systems, the user can specify target image contents by placing icons in the appropriate positions of the graphic display. With this approach, the expression of the query is greatly simplified, as the user is enabled to graphically reproduce image contents and experience their direct manipulation.

Manuscript received September 1991; revised July 1993. This work was supported in part by MURST 40%. Recommended by T. Ichikawa.

The authors are with the Dipartimento di Sistemi e Informatica, Universitá di Firenze, 50139 Firenze, Italy.

IEEE Log Number 9213124.

Images in the database may be arranged in a variety of representations so as to allow efficient retrieval of pictorial information. Different image representations proposed in the literature include pixel-oriented data structures [8], quadtrees [9] and R-trees [3], pyramidal models [5], and representations based on symbolic projections [10]. In particular, symbolic projections are an efficient and compact way to construct iconic indexes and are also an ideal way to represent visual iconic queries. Objects in the images are projected on the two image coordinate axes, and a *2D symbolic string* is derived as the output of a spatial analyzer which preserves the spatial knowledge embedded in the image. 2D-strings encode *precedence relationships* between object projections and can be used as an index for the image [7], [10], [11]. In the formulation of the iconic query, icons representing object types may also be projected on the two axes. Therefore, the visual query is also expressed as a 2D string. The 2D string associated with the iconic reconstruction is matched with the 2D strings associated with images, and the query is reduced to string–substring matching [7].

This approach recently received much attention and was developed by several authors. An extended set of operators that represent more precise relationships between object projections with respect to [7] was introduced in [12]. A representation for multiresolution symbolic pictures referred to as 2D-Hstring was presented in [13]; a generalized 2D-string representation that subsumes other classes of spatial operators, including 2D-Hstrings, and supports spatial reasoning effectively was expounded in [14]. Uncertainty management was introduced in [15], using fuzzy-set techniques. An extension of 2D strings to deal with image sequences was addressed in [16].

Two-dimensional visual iconic queries and 2D-string based representations are effective for the retrieval of images representing 2D objects or very thin 3D objects, but they might not allow an exact definition of spatial relationships for images representing scenes with 3D objects. In fact, in this case, two sources of difficulty exist that might result in an incorrect representation of the spatial relationships between objects. The first difficulty is related to the impossibility of reproducing the scene depth using 2D icons. 2D icon overlapping can be used only to a limited extent since it impacts on the intelligibility of the query. The second difficulty derives from the fact that, as demonstrated by research in experimental and cognitive psychology [17], the mental processes of human beings simulate physical world processes. Computer-generated line drawings representing 3D objects are regarded by human beings as 3D structures and not as image features, and they

imagine spatial transformations, such as rotations or shifting, directly in the 3D space.

The relevance of considering the extension of the 2D-string representation to three dimensions was perceived in [7] in order to allow more powerful image retrieval and spatial reasoning. Developments of this idea were recently carried out by a few other authors. In [18], the extension of the 2D-string to the representation of 3D scenes was theoretically addressed. In that paper, orthogonal projections on planes of a three-dimensional scene in two or three directions were considered, and three distinct representations were derived. A comparison of such representations was carried out with reference to compactness and ambiguity.

A framework for image retrieval by contents was presented in [19], where images were associated with a tertiary representation of the three-dimensional imaged scenes according to symbolic projections. Intuitively, such association was motivated by the fact that, when considering images of 3D scenes, typically human beings perceive the scene depth and keep in mind a 3D view. A visual query system was presented, in which the user manipulates 3D icons and expresses the query by example. However, in that paper, objects were assumed to coincide with their centroids, so that their extensions in the 3D space were not taken into account. Viewing objects as points has several limiting consequences. Since object extensions are not considered, only rough spatial relationships can be expressed with consequent loss of precision in the description of images; for each pair of objects, the representation language only provides precedence and coincidence operators. This also presents problems in the interaction with the system. Since spatial relationships are evaluated with respect to object centroids (that are not actually visible), relationships used in the system are not the same as those considered by the user when expressing the query visually. The relationships in the system being rougher, nonrequested images may also be given, and the user partially loses the sensation of using the same paradigm of interaction as in reality.

In this paper, a new representation language is introduced, which supports unambiguous description of *positional* and *directional* object relationships in three dimensions, still preserving object spatial extensions after projections. The language extends *interval logic* operators, which have been used to express temporal relationships in concurrent systems as well as in artificial intelligence [22], to deal with the spatial domain. Iconic retrieval from pictorial databases with 3D interfaces is motivated, and a system for querying by example with 3D icons, which supports this representation language, is also presented. The system fully allows the user to experience the sensation of directly interacting with reality with no intermediate mental processing. Retrieval is performed by placing 3D icons into a 3D virtual space to reproduce the original real scene and by selecting the appropriate viewpoint from which the image of interest is taken.

The system is designed according to the object-oriented paradigm [20] and is implemented in C++ on a PC/AT platform. It is interfaced with the Ontos Data Base Management System [21], running on a separate system in the same LAN. The virtual world visualization is obtained by using the Matrox PG-SM-1281 board. To add realism to the user's interaction, a direct manipulation interface (a manipulation glove) has been used.

The paper is organized as follows. In Section II, the subject of iconic retrieval by contents from pictorial databases is analyzed with reference to the relationships between images and imaged scenes, and the boundary between the use of 2D, 2.5D, and 3D icons is explored. In Section III, the language for the representation of spatial relationships between objects in three dimensions is introduced. In Section IV, several problems raised by the use of three-dimensional scene-based descriptions of images are discussed. In Section V, the framework for iconic retrieval by contents of images, supporting the description language and using 3D icons, is presented. In Section VI, implementation of the system is briefly explained. Finally, conclusions and future developments are expounded in Section VII.

II. Iconic Retrieval from Pictorial Databases

In this section, the boundary between the use of 2D, 2.5D, and 3D icons for visual querying by contents of pictorial databases is analyzed. To this end, first the characteristics of images and icons are briefly reviewed.

A. Images and Icons

Image contents can be described in terms either of the image subparts (*image objects*) and their spatial relationships (as derived through a segmentation task and a spatial analyzer) or of the objects in the original scene (*scene objects*) and their spatial relationships (as derived by a scene understanding system). In the first case, 2D objects are involved, and 2D spatial relationships are evaluated directly on the image plane. In the second, scenes associated with images involve objects that differ greatly in their structural properties from one application to the other. Specifically, scenes involve 2D objects (*2D scenes*) if objects have prevalently a 2D structure (such as land boundaries, roads, rivers, or very thin objects) or if drawings, maps, or aerial pictures are considered. Scenes involve 3D objects (*3D scenes*) if they are common real-world scenes. Spatial relationships for the two cases are 2D and 3D, respectively.

Icons for querying image databases are special images defined through a computer system by the user. They maintain a symbolic relationship with objects in the real world. According to the type of application, either a 2D or 3D structure can be associated with each icon to build virtual scenes with 2D or 3D objects, respectively.

B. 2D or 3D Icons

Following the previous discussion, a basic point in visual iconic querying of pictorial databases is the understanding of conditions under which the iconic query represents unambiguously image contents.

Provided that a language which supports unambiguous representation of object spatial relationships is given, as a general statement, we affirm that *an unambiguous correspondence is established between the iconic query and image contents if*

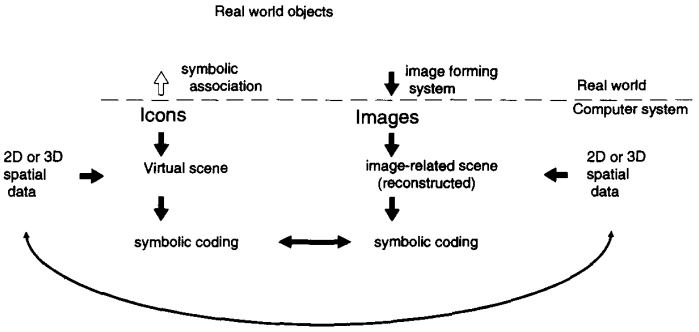


Fig. 1. Image querying through icons: the "dimensionality" of the icons must follow that of the objects in the scene represented in the image.

spatial relationships referred to are those between the objects in the scene that is represented in the image, rather than those between the objects in the image. Therefore, images should be described in terms of the original scene rather than of their subparts, and combinations of icons must represent virtual scenes rather than virtual images.

This leads to the statement that *dimensionality of data structures associated with icons must follow the dimensionality of the objects in the scene represented in the image* (see Fig. 1). A 2D structure must be associated with each icon to describe a 2D scene. A 3D structure should be employed for each icon to reproduce 3D scenes. To better understand this statement, it will be necessary to consider in more detail the alternatives for the two cases of images of 2D or 3D scenes. In the following, we will refer to 2D and 3D icons for the cases of icons with 2D and 3D structures, respectively.

Case a—Images of 2D Scenes: For the case of 2D scenes, it should be noticed that there is no practical difference in considering relationships in the image or in the scene: relationships between image objects are exactly the same as those between scene objects. Using 2D icons, an exact correspondence exists between spatial positions of icons and positions of objects in the image. A 2D iconic environment is able to express the query unambiguously. The use of 3D icons is not justified by the nature of the problem.

Case b—Images of 3D Scenes: In this case, a difference exists as to whether relationships in the images or the scenes are considered. Relationships between image objects are not the same as those between scene objects, the former being evaluated in the 2D image plane and the latter in the 3D scene space.

Using 2D icons, a single virtual scene may correspond to several arrangements of objects in a real-world scene and thus to several distinct images. This can be observed by considering the example in Fig. 2. In the image in Fig. 2(a), a simple 3D scene of a room is represented with a table, a chair, a pot of flowers, and a light fixture. If 2D icons are used, the horizontal direction in the virtual plane can be made to correspond to the X-direction in the real scene. The vertical direction can be made to correspond to either the Y- or Z-direction in the real scene. They both cannot be represented at the same time. Therefore, in the 2D iconic representation of Fig. 2(b), there is an ambiguity about the interpretation of the position of the flower pot, whether it is on the table or behind the table, as well as the position of the lamp.

These drawbacks could be overcome using icon overlapping (2.5D icons) to reproduce the missing third dimension (see Fig. 2(c) as an example). Unfortunately, overlapping can be used only for simple scenes. With more complex scenes such as that depicted in Fig. 3 (where a bridge and a highway with several vehicles are shown), the use of 2D icons may render expressing meaningful spatial relationships impossible.

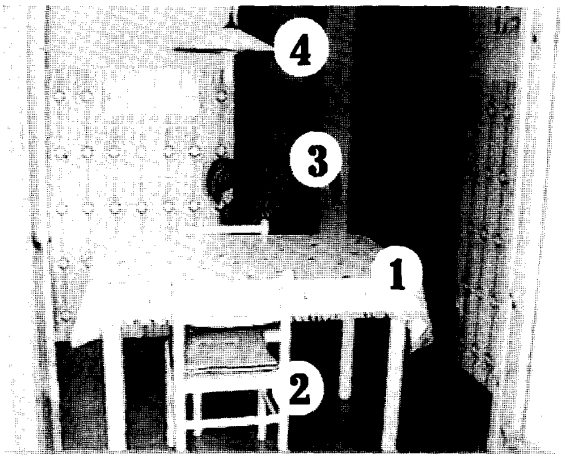
Using 3D icons, spatial relationships defined in the virtual scene exactly correspond to spatial relationships between objects in the real-world scene (see Fig. 2(d)). Therefore, once the viewpoint has been set, a single image representing the scene from that viewpoint is identified, provided that images in the database have associated their scene-based descriptions. As pointed out previously, using 3D icons in place of 2D icons to formulate the query is also in accordance with the typical behavior of the user, who expresses the query on the basis of the view of the scene he has in his mind, which is 3D. Therefore, in this approach, the user is relieved of the burden of performing a translation of his scene view into a 2D view, such as that represented in the image.

III. Symbolic Representation of 3D Scenes

In the following, according to what was discussed in the previous section, we assume that image contents are represented referring to the spatial relationships of objects in the imaged scene. Images of three-dimensional scenes are associated with a representation of their contents in three dimensions, and a three-dimensional iconic environment is used for expressing the query.

A generic scene S is thus defined as a set of objects O in a three-dimensional Euclidian space R^3, where a mapping function F from O to the power set of R^3 associates each object $o \in O$ with the set of points that it occupies in R^3. In our approach, objects are represented through their minimum enclosing parallelepiped (*mep*) and their *axial plane*, defined as the plane that contains the centroid and is orthogonal to the minimum elongation axis of the *mep*.

A symbolic description of the scene is a set of formulas expressing mutual relationships between pairs of objects with reference to a coordinate system $E = \{e_i\}$, $i = 1, 2, 3$. Two



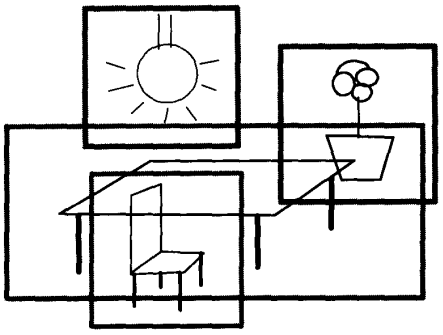
1: table

2: chair

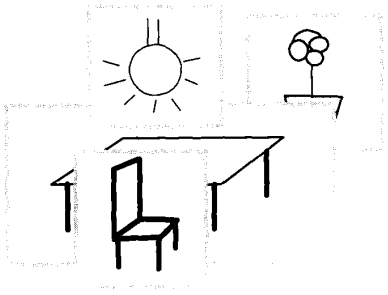
3: flower pot

4: light fixture

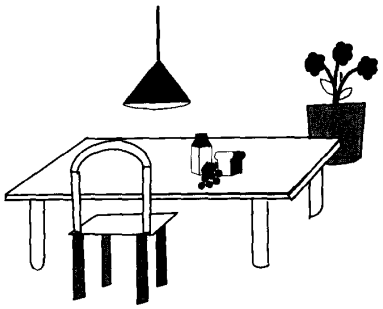
(a)



(b)



(c)



(d)

Fig. 2. (a) Sample image representing a simple real world scene with 3D objects. (b) Iconic representation using 2D icons. (c) Iconic representation using 2D icons with icon overlapping (2.5D icons). (d) Iconic representation using 3D icons.

types of formulas are considered, dealing with either *positional* or *directional* relationships between objects. *Positional formulas* take into account relationships between *mep* orthogonal projections over one axis of the reference system (see Fig. 4). *Directional formulas* consider relationships between axial planes of the objects.

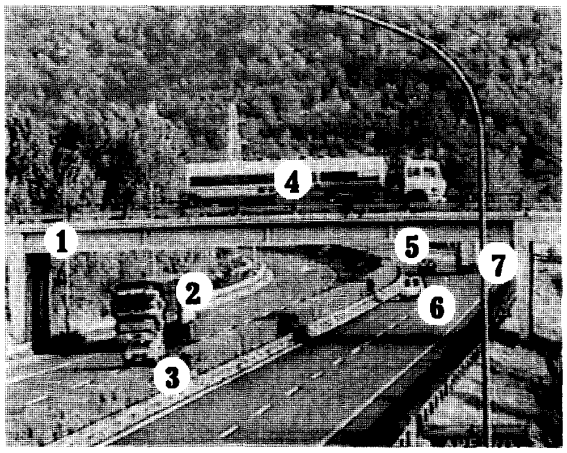
Positional formulas ϕ_i consider intervals originated by *mep* projections on the i axis and are expressed as

$$\phi_i ::= p_i \,|\, p_i < O_p > q_i$$

where p_i and q_i are the projections over the i axis of generic objects p and q, and O_p is a spatial operator that relates the two intervals originated by the projections of p and q on the same axis i. Due to the nature of the model assumed for space, spatial operators have been derived extending operators used for the specification of temporal relationships between time intervals in *interval logic* [22], with an obvious shift of interpretation [12]. All 13 possible relationships between a pair of intervals are expressed, preserving object spatial extension after projection.

Truth values of spatial formulas are determined according to the semantic rules given below for the case of the x axis. For the sake of notation, projections of p and q are referred to as $p ::=< x_{p1}, x_{p2} >$, and $q ::=< x_{q1}, x_{q2} >$, respectively, x_k being points of the x axis of the reference system with $x_{p1} < x_{p2}$ and $x_{q1} < x_{q2}$:

- p *strictly-after* q iff $x_{q2} < x_{p1}$;
- p *after with right adjacency* q iff $x_{q2} = x_{p1}$ and $x_{q2} < x_{p2}$;
- p *after* q iff $x_{q1} < x_{p1} < x_{q2}$ and $x_{q2} < x_{p2}$;
- p *is_included_by with left adjacency* q iff $x_{q1} < x_{p1} < x_{q2}$ and $x_{q2} = x_{p2}$;
- p *spatial coincidence* q iff $x_{q1} = x_{p1}$ and $x_{q2} = x_{p2}$;



1: bridge

2: truck_1

3: car_1

4: tank-truck

5: car_2

6: car_3

7: truck_2

Fig. 3. Sample image representing a complex real world scene with 3D objects.

- p *is_ included_by* q iff
  $x_{q1} < x_{p1} < x_{q2}$ and $x_{q1} < x_{p2} < x_{q2}$;
- p *is_ included_by with right adjacency* q iff
  $x_{q1} = x_{p1}$ and $x_{q1} < x_{p2} < x_{q2}$;

and those for the dual operators:

- p *strictly-before* q;
- p *before with left adjacency* q;
- p *before* q;
- p *includes* q;
- p *includes with right adjacency* q;
- p *includes with left adjacency* q,

that may be easily derived from the previous definitions. The visual sketch of the meaning of positional relationships and the symbols used are given in Fig. 5.

Directional formulas ψ consider *collinearity* and *parallelism* relationships between axial planes of object *meps*. These formulas are expressed as

$$\psi ::= < p|q > | < p \| q > | < p \nparallel q >$$

where | is the symbol of collinearity, || the symbol of parallelism, and $\nparallel$ the symbol expressing the absence of parallelism, collinearity being a stronger relationship than parallelism.

Truth values of directional formulas ψ are determined by the semantic rules given below:

- $p \mid q$ iff objects p and q share a common axial plane;
- $p \| q$ iff objects p and q are not collinear but their axial planes are parallel;
- $p \nparallel q$ iff objects p and q are not parallel.

Since for each pair of objects, we can express only one positional formula for each axis and one directional formula for relationships between axial planes, respectively, the spatial relationship between a pair of objects p and q may be more synthetically expressed as

$$p \quad R \quad q$$

where

$$R = (O_{px}, O_{py}, O_{pz}, O_d),$$

with O_{pk} any operator from the set of positional operators evaluated along the k axis of the coordinate reference system, and O_d any operator taken from the set of directional operators.

The scene depicted in Fig. 3 is thus described as in the following, with reference to the Cartesian coordinate system of the camera with the Z axis coinciding with the optical axis (symmetric relationships are not reported for the sake of simplicity):

1 - 2 : $(\diamond, |\diamond, \times, \nparallel)$ 2 - 3 : $(\diamond, >, \gg, \|)$
1 - 3 : $(\diamond, >, \gg, \nparallel)$ 2 - 4 : $(\ll, \ll, \diamond, \nparallel)$
1 - 4 : $(\diamond, \lhd, \diamond, \|)$ 2 - 5 : $(\ll, <, \ll, \nparallel)$
1 - 5 : $(\diamond, \diamond, \ll, \nparallel)$ 2 - 6 : $(\ll, \diamond, \ll, \|)$
1 - 6 : $(\diamond, \diamond, \ll, \nparallel)$ 2 - 7 : $(\ll, \ll, \ll, \nparallel)$
1 - 7 : $(\diamond, \diamond, \ll, \nparallel)$

3 - 4 : $(\ll, \ll, \ll, \nparallel)$ 4 - 5 : $(\diamond, \gg, \ll, \nparallel)$
3 - 5 : $(\ll, \ll, \ll, \nparallel)$ 4 - 6 : $(\diamond, \gg, \ll, \nparallel)$
3 - 6 : $(\ll, \ll, \ll, \|)$ 4 - 7 : $(\ll, \gg, \ll, \nparallel)$
3 - 7 : $(\ll, \ll, \ll, \nparallel)$

5 - 6 : $(>, >, \gg, \nparallel)$ 6 - 7 : $(\ll, \ll, \ll, \nparallel)$
5 - 7 : $(\ll, \ltimes, \times, \|)$

Three-dimensional virtual scenes can also be represented with the same language, considering projections of icons on the axes of a reference system.

IV. Issues with 3D Scene-Based Descriptions

Introduction of representations of 3D scenes with extended objects for image retrieval raises new problems to be solved with respect to the 2D domain. These include the selection of the reference coordinate system, the automatic derivation of 3D descriptions of the images and the expression of relationships with complex 3D objects.

A. Selection of the Reference Coordinate System

With the symbolic projections approach, it is required that scene descriptions are made with respect to a reference coordinate system. Two distinct approaches may be followed using *object-centered* and *observer-centered* scene descriptions, respectively.

Object-centered scene descriptions are independent of the observer's viewpoint. Each object is provided with its individual reference coordinate system, and the overall description

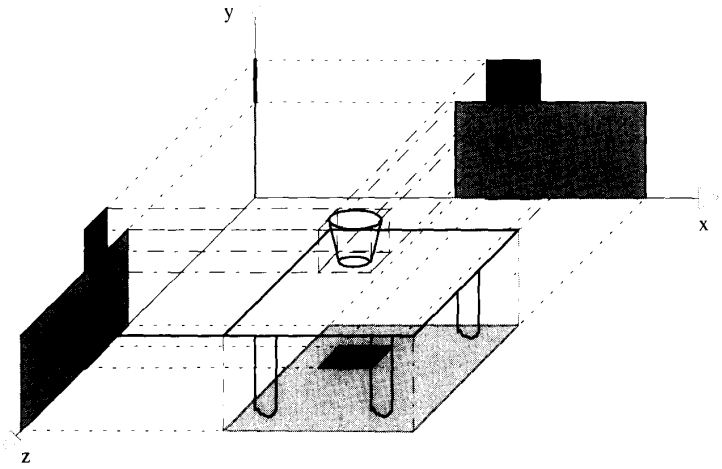


Fig. 4. Orthogonal projections of a three-dimensional scene.

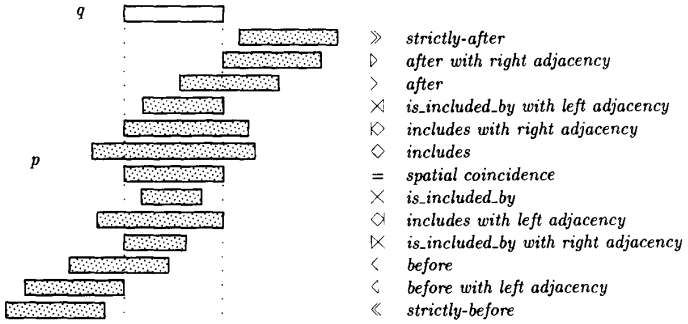


Fig. 5. Mutual spatial relationships and operators between two intervals.

of the scene is given by a set of descriptions, each capturing how one object *sees* the other objects with respect to its own reference coordinate system. Some criteria are required to establish the orientation of the private coordinate system of the object with respect to the object structure. With object-centered descriptions, images of one scene taken from different viewpoints all have the same scene description associated.

Observer-centered scene descriptions are based on the position of the observer's viewpoint. Positional relationships are referred to the Cartesian coordinate system of the observing camera with the Z axis coinciding with the optical axis. With observer-centered descriptions, images of one scene taken from different viewpoints have distinct scene descriptions associated.

Generally speaking, the selection of the appropriate description to be associated with the scene depends on the objectives of the retrieval task. In the presence of no prior knowledge or interest about the observer's viewpoint, an object-centered description can be assumed. Otherwise, observer-centered descriptions are preferred. In the system presented in the following, observer-centered descriptions are used both for the original imaged scene and the virtual scene reconstructed by the user in the visual query. The description of the query coincides with that of the target image only if the viewpoint that is set in the visual query is the same as that from which the image was taken.

B. Automatic Determination of Scene Descriptions

Automatic determination of scene descriptions to be associated with images is not as straightforward as the association of 2D image descriptions with images, since model-based object recognition and image understanding processes [23]–[26] must be applied to the original image to emulate human capabilities to recover the 3D scene structure from 2D images and its description based on the observer's point of view. Typically multiple views of the same scene are needed to obtain a complete scene description. In the system expounded in the following, scene descriptions have been directly defined by the user through an interactive facility of the system.

C. Relationships with Complex 3D Objects

Use of *meps* to model objects and icons allows object extensions in the derivation of spatial relationships to be taken into account. However, in the presence of scenes with complex object structures that encompass smaller objects (such as bridges, arcs, L- or U-shaped buildings, etc.) ambiguous descriptions are possible. An example is provided in Fig. 6(a) and (b), where, as the complex object is regarded as a whole,

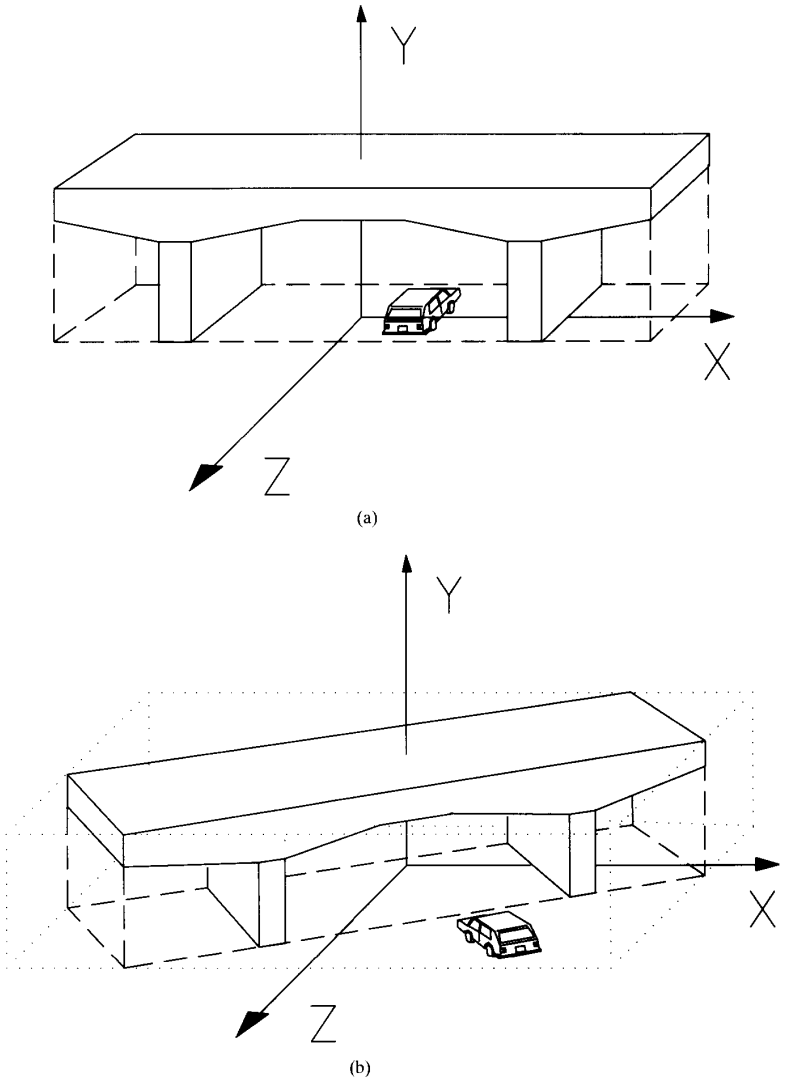


Fig. 6. A full inclusion relationship between *mep* projections for the cases: (a) car under a bridge, and (b) car in front of a bridge.

a relationship of full inclusion between *mep*s is detected in both situations.

In order to avoid such ambiguities, we defined complex objects as *composed objects* [27] in our retrieval systems, and relationships in the imaged scene were evaluated with reference to the *mep*s of the component objects. Therefore, the bridge in Fig. 6(a) is regarded as composed of a road on two piers, and relationships are derived between the individual bridge components and the car.

Correspondingly, *3D composed icons* have been introduced. 3D composed icons are special icons representing composed objects. Similar to the objects that they represent, composed icons are made up of simpler icons that individually represent a component object. They retain a description of their structure in terms of spatial relationships between component icons and express more precisely relationships between icons. In order to analyze relationships between composed icons, as the system detects an intertwining relationship between icon *mep*s, this is further developed by considering the relationships with the *mep*s of the individual components.

V. A 3D Iconic Environment for Image Retrieval

In the following, a 3D iconic environment for querying a database of images of real-world scenes is presented. The system is designed according to three basic principles:

- to introduce a description of real world images in the database in terms of the 3D description of the scene that they represent;
- to perform querying through a virtual scene that is defined with a direct manipulation of 3D icons;
- to represent spatial relationships between objects both in

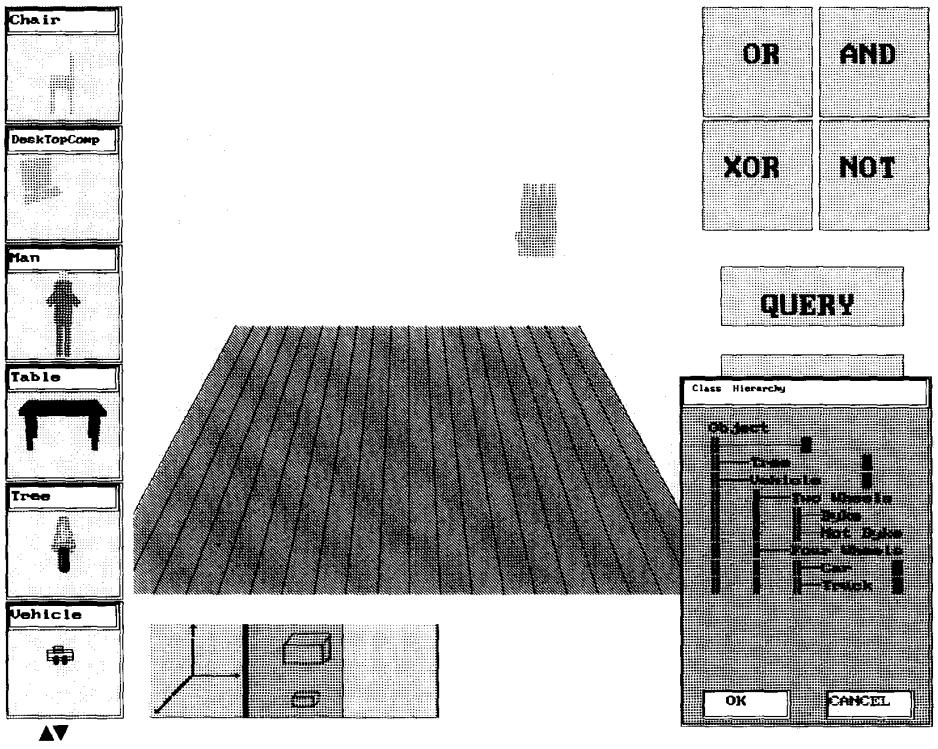


Fig. 7. The system user interface. From left to right, IES, FS, CI, DC, and LOS windows of the Dialogue_Interface. The Query_Window is in the central part of the screen.

the original and virtual scenes according to the representation language expounded previously.

Images and their descriptions are stored in an object-oriented database [21]. Icons used for the query specification are also stored in the database. These represent categories of objects (*category_icons*) or object instances (*object_icons*). Icons are comprised of a graphical form and a text body [28], the latter specifying the attributes of the entity represented by the icon. Attributes of the classes associated with icons are the same as those in the classes of the corresponding objects detected in the images. Icon classes are arranged into an inheritance hierarchy. The inheritance hierarchy can be inspected for the selection of icons during the query construction or for the definition of new icon classes.

A. Interaction with the System

The visual interface of the system (see Fig. 7) is comprised of a *Dialogue_Interface*, which supports the selection of icons from the database and other facilities, as well as a *Query_Window*, for the expression of the 3D query and the visualization of the query results. To add realism to the user's interaction, a *manipulating glove* was introduced to support direct interaction with 3D icons (see Fig. 8). The glove used provides no force feedback to the user. An audio feedback is given when icons are grasped and when collisions between icons are caused by the user in the virtual space. Human factors are improved since the user interacts with the system through a 3D gesture and has the feeling of grasping and moving objects as in the real world. The absence of physical constraints (such as the box for some 3D mice) further enhances the normality of operation.

When operations are made in the Query_Window, a 3D cursor follows the movements of the operator's hand and allows rotation, translation, and grasping operations. When used in the Dialogue_Interface, the glove simply operates as a 2D pointing device and an arrow cursor is visualized. A visual language supports interpretation of operations with the glove.

B. Operations in the Dialogue_ Interface

The Dialogue_Interface section is organized into several distinct windows (see Fig. 7):

- An *Iconic_Entity-type_Selection (IES)* window (in the left part of the screen) which includes a list of available 3D icons.
- Three *Facilities_Selection (FS)* windows (in the lower part of the screen), which support the user interaction with the system. In particular, the user can change the point from which the virtual scene is viewed (i.e., the orientation of the virtual camera), by defining the appropriate rotations of the scene along the three axes (the left window). Zooming of the scene is supported in the middle window. In the right window, the possibility of rotating the hand in the virtual space is provided, which is a function that is not directly supported by the glove.
- A *Class-hierarchy_Inspection (CI)* window (in the lower right part of the screen), where the user inspects the data-

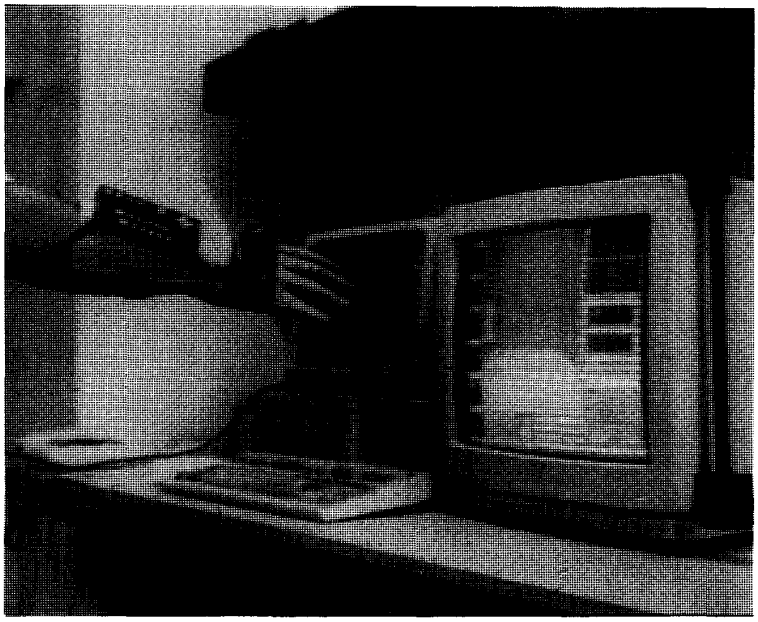


Fig. 8. Interaction with the system using the manipulating glove.

base icon class hierarchy to search for the desired icon. Icon class-hierarchy navigation is performed by pointing at the hierarchy branch of interest and looking at the class names that appear at the different levels. When the desired level of representation is reached, the user can either ask for available object_icons (representing instances of the selected category) or select the category_icon for the definition of the visual query.

- A *Database_Command (DC)* window, in which two buttons are present for issuing commands to process the query and to display the icon class hierarchy.
- A *Logic_Operation_Selection (LOS)* window (in the upper right part of the screen), which includes icons of logical operators AND, OR, XOR, and NOT to be used to define complex visual queries.

C. Operations in the Query_Window

Queries on the pictorial database are expressed by example. The user can put the selected icon in any place in the virtual space to reconstruct the real scene represented in the images to be extracted from the database. Spatial operators between two icons such as *Right, Behind, In_front_of, Above* ... are naturally defined by the user, by locating the 3D icons at the appropriate positions in the space. When the icon of interest is selected, the text body of the corresponding class is presented to the user for a possible specification of logical or numerical constraints on the attribute values, in order to restrict the query scope. An example of visual query is reported in Fig. 9, where office images are searched, including a desk, a desk-top computer, a person, and a picture, with their relative positions derived with respect to a selected observer's viewpoint. Once the position and orientation of the virtual camera are chosen, spatial relationships between pairs of icons are evaluated considering *mep* projections with reference to the Cartesian coordinate system of the camera. The requested images are presented on the screen after the query parsing. The result of the query is shown in Fig. 10, where the image which corresponds to the scene depicted has been retrieved. The user can change his point of observation of the scene during the query construction, using the point-of-view-change facility of the system. It follows that the user can issue a query for an image of the same scene from a different point of view by simply using this facility and thus avoiding the rewriting of the query. In Fig. 11, a different query is issued by rotating the previous virtual scene. From the new viewpoint, mutual relationships of icons have been changed and the query answer (see Fig. 12) presents different images representing a person seating at the desk.

The rotation of the original scene requires the automatic updating by the system of the spatial relationships according to the new viewpoint. In this operation, the possibility of full occlusion of icons must be taken into consideration in order to avoid issuing a wrong query to the database. Icons that are no longer visible from the new viewpoint must not be considered and the relationships with these icons must be updated in the new scene. A full occlusion condition is detected between the two icons p and q (i.e., q is occluded by p) if the following relationships between *mep* projections are satisfied:

$$O_{px} = \diamond || \diamond | \diamond |$$
$$O_{py} = \diamond || \diamond | \diamond |$$
$$O_{pz} = \ll | < | \lhd$$
$$O_d = any.$$

In this case, the occluded object is discarded from the object list.

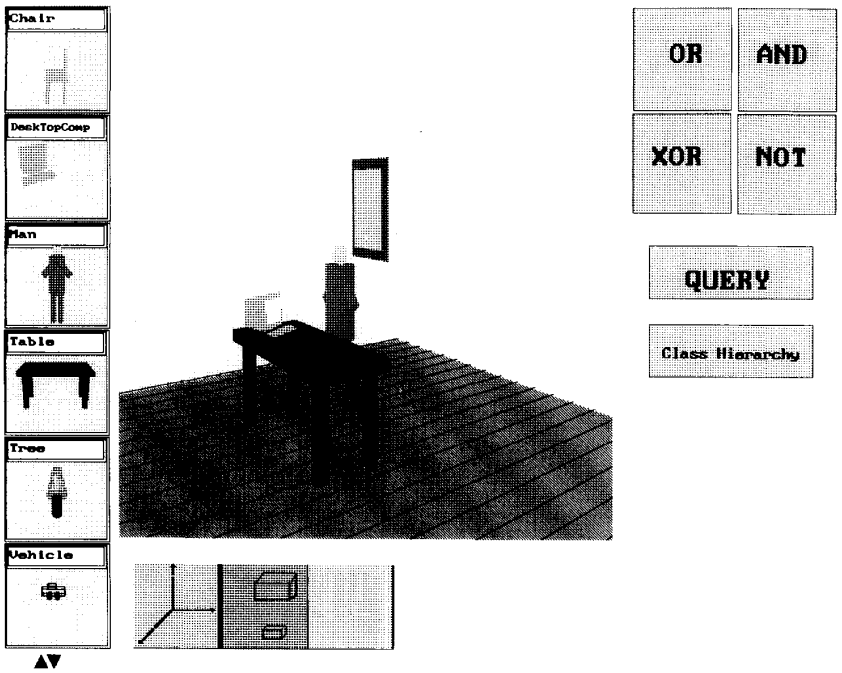


Fig. 9. Definition of a virtual scene by placing selected icons in the virtual space of the Query_Window.

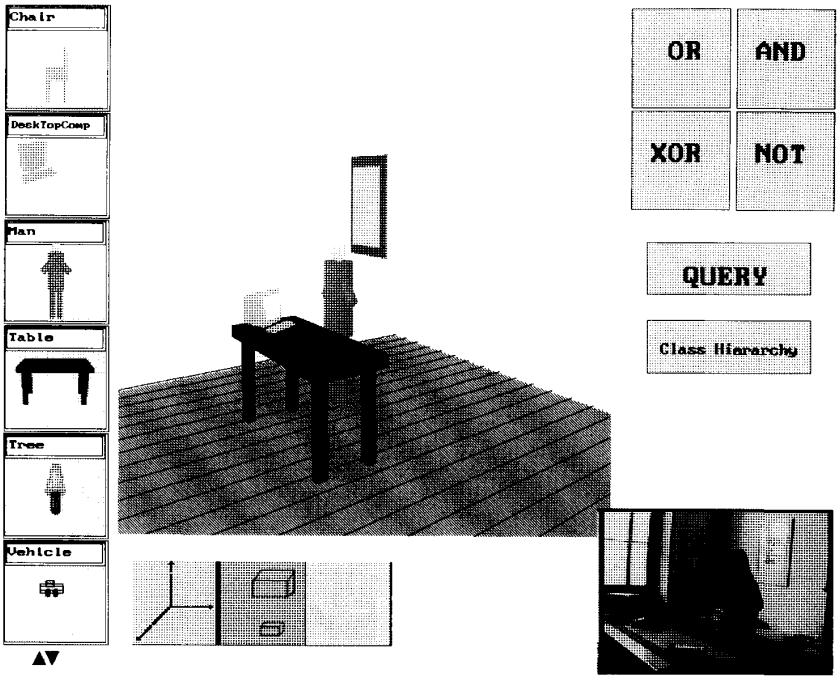


Fig. 10. Results of the query in Fig. 9.

Boolean connectives can be used to form a more complex query from two separate queries, as shown in Fig. 13, where the query of Fig. 11 is ORed with a query for a person, a desk, a pot of flowers, and a picture as represented in the smaller Query_Window. Results are shown in Fig. 14. Complex queries can be further combined with normal queries by logical operators to express even more complex queries.

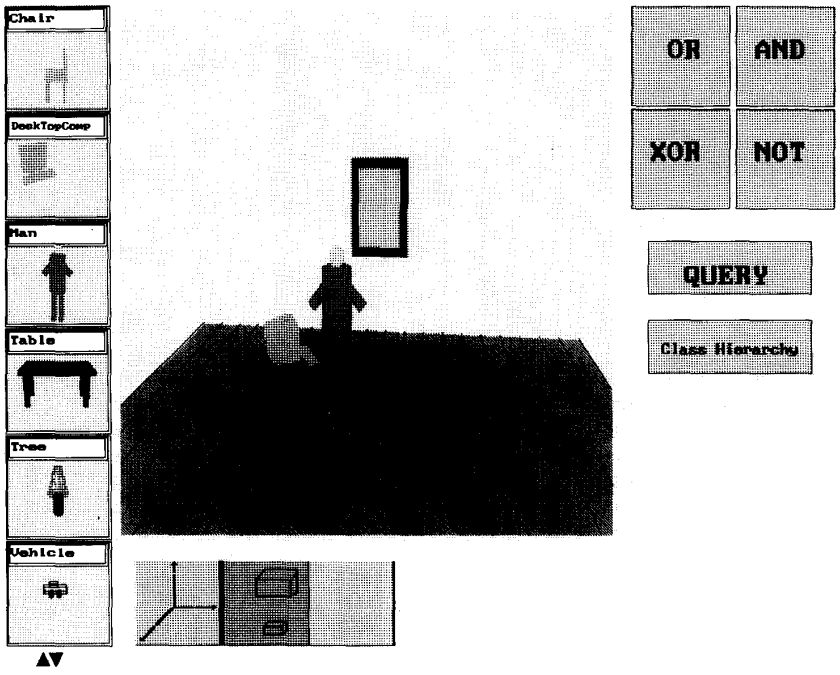


Fig. 11. A new query can be issued by simply changing the point of view of the scene previously defined (see Fig. 9).

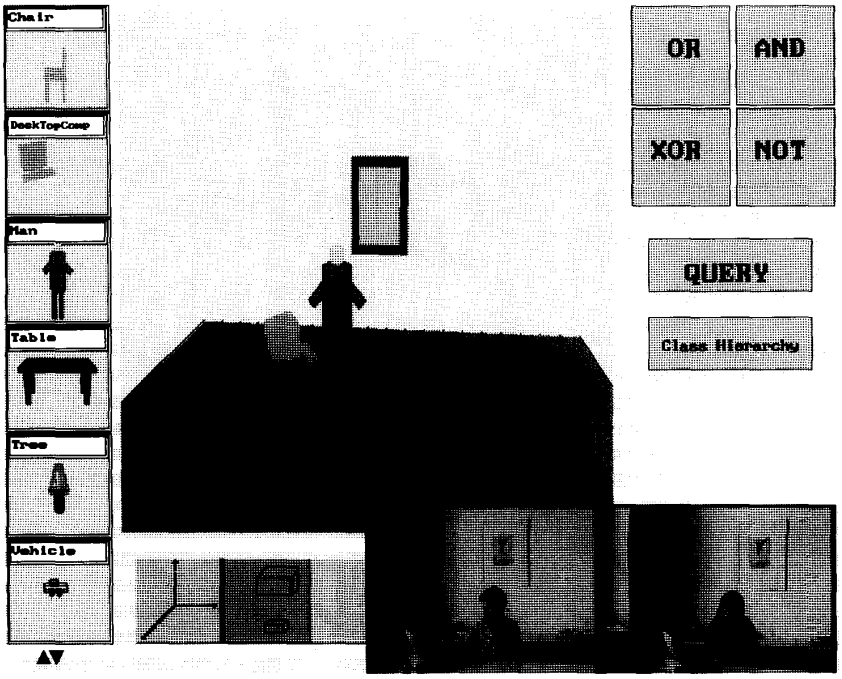


Fig. 12. Results of the query in Fig. 11.

VI. System Implementation

The system was designed according to the object-oriented paradigm. Operations are supported through several subsystems:

- *Screen*, which supports visualization and interaction with the user.

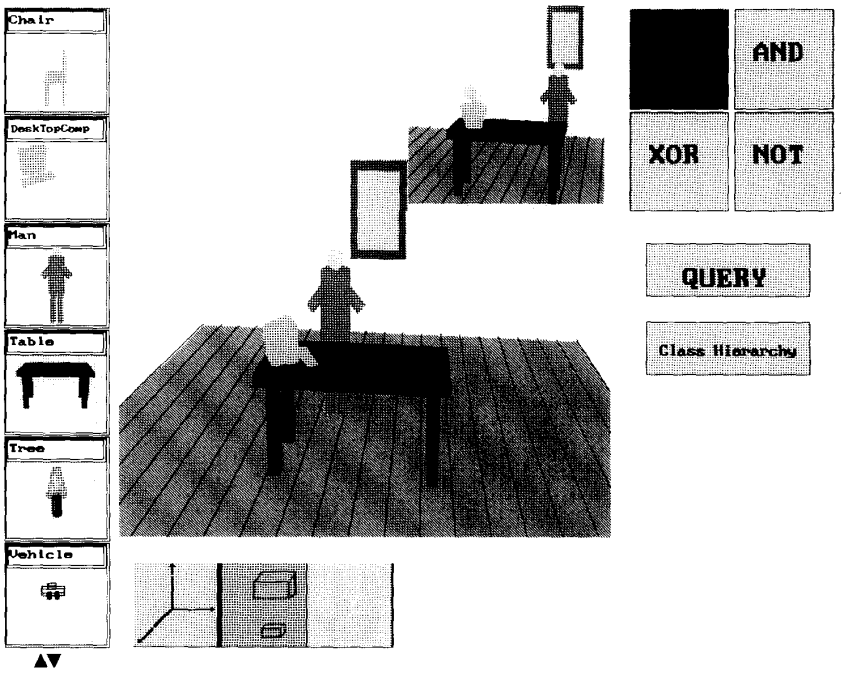


Fig. 13. Simple queries can be combined to form more complex queries using logical operators. The query in Fig. 11 is ORed with the new query in the smaller Query_Window.

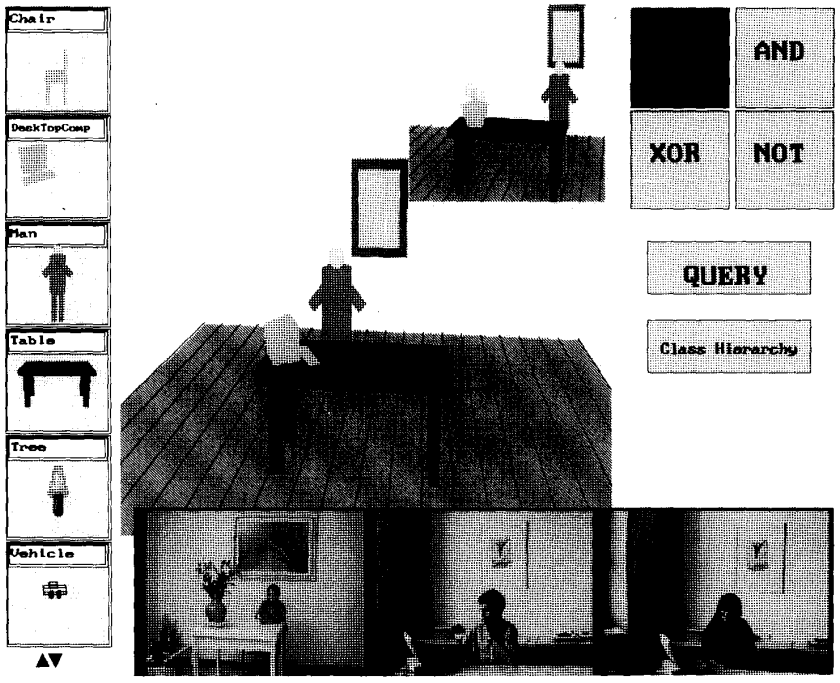


Fig. 14. Results of the query of Fig. 13.

- *Scanner* and *Parser*, which cooperate in the translation of the spatial relationships between objects into a symbolic sentence.
- *Object_Interface*, which translates the symbolic sentence into the database language.
- *Database*, which stores permanent objects.

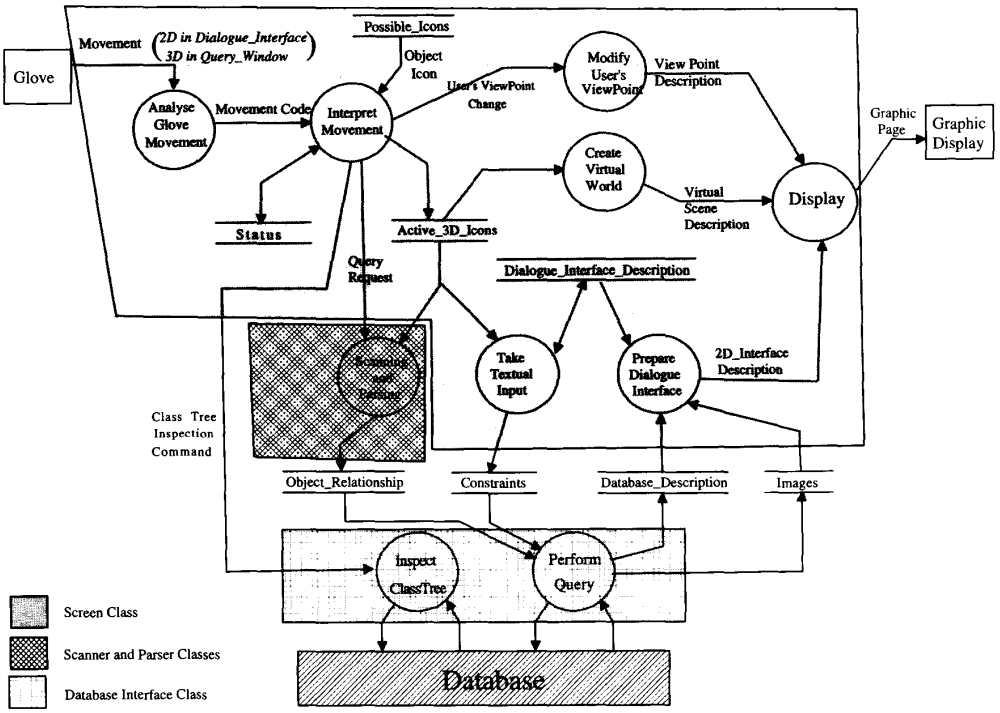


Fig. 15. Flow diagram of the system operation.

These elements and their principles of operation are briefly presented in the following. In Fig. 15, a flow diagram of the system operation is sketched.

A. Screen

Screen is composed of a 2D windowing facility, a 3D virtual world editor, and a multimodal interaction support (glove and keyboard). Screen keeps a state which is the set of the positions of the icons in the virtual space. State transitions occur when messages are received from the *Hand* object (which is the interface to the manipulating glove) or from the human-interface objects (windows, buttons). In the *Cursor_2D* state, cursor movements are allowed only in the Dialogue_Interface part of the screen (e.g., for the navigation of the icon class-hierarchy). In the *Cursor_3D* state, the cursor can only be used to point to an icon to be selected, but icons cannot be moved by the user. When the icon is selected, the state *Cursor_obj* is entered, and the icon can be moved within the virtual space. Positions of the icons are stored with reference to the absolute Cartesian coordinate system of the graphic board. Since the user can change his point of view, when the query is completed, the final position and orientation of the virtual camera are stored into a *Point_of_View* object, which is used in order to determine the camera-relative positions of the icons. The *Obj3D_list* object keeps the references to the icons in the virtual space.

B. Scanner, Parser and Object_Interface

As each icon is placed or moved in the virtual space, the scanner creates a set of *Double_pos* objects that capture the relative positions of a pair of icons. When the query is committed, the parser translates these relative positions into the symbolic representation. Relationships are evaluated with reference to the coordinate system of the virtual camera, so as to take into account the observer's viewpoint and obtain an observer-centered description of the scene. The Object_Interface subsystem translates the query (including parsed spatial relationships, object constraints, and logical expressions) into the database language (Object SQL). The syntax of the query is briefly expounded in the Appendix with an extended BNF notation.

C. Database

According to the database architecture, data are stored in the form of objects rather than records or tables. For each stored image, two basic objects are instantiated of the classes *Image* and *Imaged_scene*. Image instances keep raw image data, while *Imaged_scene* instances keep symbolic descriptions of the imaged scenes. Each Imaged_scene instance has a direct reference to its corresponding Image instance and to a list of binary spatial relationships between the objects of the original scene (*Spatial_relationships* instance). Binary relationships between objects are of the types: *Object-to-Object*, *Object-to-Subobject* or *Subobject-to-Subobject*. References are also kept between each Imaged_scene instance and Scene_object instances which represent the objects detected in the scene. Imaged_scenes and their Scene_object and Spatial_relationships instances are clustered together on the disk so as to be retrieved in a

single database operation. Matching is recognized if the scene description issued is at least contained in that defined in the Imaged_Scene and object attribute values are in the allowed range. Since images are not yet read in memory when the query is resolved, the pointer is initially set to the value *inactive*. Only if matching is recognized is the Image instance activated.

VII. Conclusions

In this paper, the subject of retrieval by contents of pictures depicting 3D scenes through three-dimensional iconic interfaces has been addressed. In this approach, images are referred to by considering spatial relationships between objects in the 3D imaged scene. A representation language has been introduced that expresses unambiguous positional and directional relationships, preserving object spatial extension after projection. The use of 3D interfaces provides a significant shift of quality in the user interaction with respect to 2D interfaces. In fact, in this case, the intermediate step is avoided between the real scene and its graphical description as required with the paradigm of 2D icon manipulation. Using 3D icons, the user experiences a mental shift from the sensation of interacting with a schematic representation of reality, to the sensation of directly interacting with reality. If object spatial extensions are preserved, solid objects visually satisfy relationships as seen in reality. Principles of operation and organization of a visual environment for the retrieval of images using 3D icons based on this language were also expounded. Scene-based tertiary descriptions of images and 3D iconic interfaces appear even more interesting for the symbolic representation of image sequence contents, where 3D scenes and movements are typically involved [29], [30].

Appendix

The syntax of the query is briefly reported in the following through an extended BNF notation:

⟨query_expression⟩ ::= ⟨single_query⟩ — ⟨complex_query⟩
⟨complex_query⟩ ::= **[not]** ⟨complex_query⟩ |
⟨single_query⟩ ⟨boolean_op⟩ ⟨single_query⟩ —
⟨complex_query⟩ ⟨boolean_op⟩ ⟨single_query⟩
⟨single_query⟩::= {⟨icon⟩⟨icon_unary_spatial_op⟩}|
{ ⟨icon⟩⟨icon_binary_spatial_op⟩ ⟨icon⟩}
⟨boolean_op⟩ ::= **and** | **or** | **xor**
⟨icon⟩ ::= ⟨object_icon⟩ |
⟨category_icon⟩ [⟨category_icon_op⟩]
⟨icon_unary_spatial_op⟩ ::= ⟨icon_position⟩ |
everywhere
⟨icon_position⟩ :: = **xyz-mep projections**
⟨icon_binary_spatial_op⟩ ::= $\{\langle O\_p \rangle\}^3 < O\_d \rangle$
$\langle O\_p \rangle$ ::= $> \mid < \mid \gg \mid \ll \mid \rhd \mid \lhd \mid =$ |
$\diamond \mid \times \mid |\diamond \mid \diamond| \mid \ltimes \mid \rtimes$
$\langle O\_d \rangle$ ::= $\mid \mid \parallel \mid \not\parallel$
⟨category_icon_op⟩ ::= {⟨attribute ⟩⟨operation⟩}
⟨attribute⟩ ::= **attribute_from** (⟨class⟩)
⟨operation⟩ ::= ⟨operator⟩⟨value⟩
[{ ⟨boolean_op⟩⟨operator⟩ ⟨value⟩ }]
⟨operator⟩ ::= ⟨numeric_operator⟩ ⟨symbolic_operator⟩
⟨numeric_operator⟩ ::= = | ≠ | > | < | >= | <=
⟨symbolic_operator⟩ ::= **eq** | **ne** | **gt** | **lt** | **ge** | **le**
⟨value⟩ ::= **constant**

References

[1] N. Chang and K. Fu, "Query by pictorial example," *IEEE Trans. Software Eng.*, vol. SE-6, pp. 519–524, Nov. 1980.
[2] T. Joseph and A. Cardenas, "PICQUERY: A high level query language for pictorial database management," *IEEE Trans. Software Eng.*, vol. 14, pp. 630–638, May 1988.
[3] N. Roussopoulos, C. Faloutsos, and T. Sellis, "An efficient pictorial database system for PSQL," *IEEE Trans. Software Eng.*, vol. 14, pp. 639–650, May 1988.
[4] J. Orenstein and F. Manola, "PROBE: Spatial data modeling and query processing in an image database application," *IEEE Trans. Software Eng.*, vol. 14, pp. 611–629, May 1988.
[5] M. Tanaka and T. Ichikawa, "A visual user interface for map information retrieval based on semantic significance," *IEEE Trans. Software Eng.*, vol. 14, pp. 666–670, May 1988.
[6] T. Kato, "Database architecture for content-based image retrieval," *Image Storage and Retrieval Systems, SPIE,* vol. 1662, pp. 611–629, 1992.
[7] S. Chang, C. Yan, D. Dimitroff, and T. Arndt, "An intelligent image database system," *IEEE Trans. Software Eng.*, vol. 14, pp. 681–688, May 1988.
[8] M. Chock, A. Cardenas, and A. Klinger, "Database structure and manipulation capabilities of a picture database management system (PICDMS)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, pp. 484–492, July 1984.
[9] H. Samet, "The quad-tree and related data structures," *ACM Comput. Surveys*, vol. 16, June 1988.
[10] S. Chang and C. Yan, "Iconic indexing by 2-D strings," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, pp. 413–428, May 1987.
[11] S. Chang and S. Liu, "Picture indexing and abstraction techniques for pictorial database," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 475–484, July 1984.
[12] P. Holmes and E. Jungert, "Symbolic and geometric connectivity graph methods for route planning in digitized maps," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 549–565, May 1992.
[13] S. Chang and Y. Li, "Representation of multi-resolution symbolic and binary pictures using 2DH strings," in *Proc. IEEE Workshop Languages for Automation*, 1988.
[14] S. Chang and E. Jungert, "Pictorial data management based upon the theory of symbolic projections," *J. Visual Languages Comput.*, vol. 2, pp. 195–215, June 1991.
[15] S. Lee, M. Shan, and W. Pang, "Similarity retrieval of iconic image database," *Pattern Recognition*, vol. 22, pp. 675–682, June 1989.
[16] T. Arndt and S. Chang, "Image sequence compression by iconic indexing," in *Proc. IEEE VL'89 Workshop Visual Languages*, Oct. 1989, pp. 177–182.
[17] L. Cooper and R. Shepard, "Turning something over in the mind," *Sci. Amer.*, pp. 114–120, Dec. 1984.
[18] G. Costagliola, G. Tortora, and T. Arndt, "A unifying approach to iconic indexing for 2-D and 3-D scenes," *IEEE Trans. Knowledge Data Eng.*, vol. 4, pp. 205–222, June 1992.
[19] A. Del Bimbo, M. Campanai, and P. Nesi, "3D visual query language for image databases," *J. Visual Languages Comput.*, vol. 3, pp. 257–271, Sept. 1992.
[20] M. Stefik and D. Bobrow, "Object-oriented programming: Themes and variations," *AI Mag.*, vol. 6, no. 4, pp. 40–62, 1986.
[21] ONTOS, "Object-oriented database documentation, rel. 1.42, operating system: Os/2," Ontologic Inc., MA, Tech. Rep., 1989.
[22] J. Halpern and Y. Shoham, "A propositional modal logic of time intervals," *J. Ass. Comput. Mach.*, vol. 38, pp. 935–962, Oct. 1991.
[23] P. Besl and R. Jain, "Three-dimensional object recognition," *Comput. Surveys*, vol. 17, pp. 75–145, Mar. 1985.
[24] R. Brooks, "Model-based three-dimensional interpretation of two-dimensional images," in *Readings in Computer Vision*, M.A. Fischler and O. Firschein, Eds. Los Altos, CA: Morgan Kauffman, 1987, pp. 360–371.
[25] M. Herman and T. Kanade, "The 3-D MOSAIC scene understanding system: Incremental reconstruction of 3D scenes for complex images," in *Readings in Computer Vision*, M. A. Fischler and O. Firschein Eds. Los Altos, CA: Morgan Kauffman, 1987, pp. 471–483.

[26] G. Adiv, "Inherent ambiguities in recovering 3D motion and structure from a noisy flow field," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 477–489, May 1989.
[27] L. Mohan and R. Kashyap, "An object-oriented knowledge representation for spatial information," *IEEE Trans. Software Eng.*, vol. 14, no. 5, pp. 675–688, 1988.
[28] S. K. Chang, "Principles of visual languages," in *Visual Programming Syst.*, S. K. Chang, Ed. Englewood Cliffs, NJ: Prentice Hall, 1990, pp. 1–59.
[29] A. Del Bimbo, E. Vicario, and D. Zingoni, "A spatio-temporal logic for image sequence coding and retrieval," in *Proc. IEEE VL92 Workshop on Visual Languages*, Sept. 1992.
[30] A. Del Bimbo, E. Vicario, and D. Zingoni, "Symbolic indexing of image sequences with spatio-temporal logic," Univ. Firenze, Firenze, Italy, Tech. Rep. 8 93; also submitted to *IEEE Trans. Knowledge Data Eng.*, Feb. 1993.

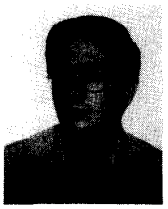
Alberto Del Bimbo (M'90) was born in Florence, Italy, in 1952. He received the doctoral degree in electronic engineering from the University of Florence, Italy.

He was with IBM Italy from 1978 to 1988. He is currently Professor of "Reti Logiche" (Digital Systems) with the Department of Sistemi e Informatica at the University of Florence, Italy. His research interests and activities are in the field of distributed systems, computer vision, and visual languages.

Dr. Del Bimbo is a member of the IAPR (the International Association for Pattern Recognition) and is in the board of IAPR Technical Committee 8 (Industrial Applications). He is also a reviewer for CEC (the Commission of the European Community) at the Directorate of Telecommunications, Information Industries and Innovation, Office and Business System Division.

Maurizio Campanai was born in Arezzo, Italy, in 1963. He received his degree in electronic engineering in 1991 from the University of Florence, Italy.

He is currently the technical responsible for CESVIT CQ_ware (Center for Software Quality) at the Department of Sistemi e Informatica, University of Florence, Italy. His main interests and research areas include software engineering, visual languages, and software quality assurance.



Paolo Nesi (M'92) was born in Florence, Italy, in 1959. He received the degree in electronic engineering from the University of Florence, Italy. In 1992, he received the Ph.D. degree in electronic and informatic engineering from the University of Padoa, Italy. Since 1987 he has been active on different research topics, including robot vision, motion analysis, parallel architectures, and visual man–machine interaction. In 1991 he was a visitor at the IBM Almaden Research Center, CA. Since November 1991 he has been with the Department of Sistemi e Informatica at the University of Florence, Italy, as a researcher and Assistant Professor of Computer Science. Dr. Nesi is a member of IAPR.