T.L. Kunii (Editor)
Elsevier Science Publishers B.V. (North-Holland)
IFIP, 1989

GRIM_DBMS: a GRaphical IMage DataBase Management System

Fausto RABITTI (*) and Peter STANCHEV (**)

(*) Istituto di Elaborazione della Informazione
National Research Council
Via Santa Maria 46
56100 Pisa, Italy

(**) Institute of Mathematics
Bulgarian Academy of Sciences
G. Bonchev Str., 8
1113 Sofia, Bulgaria

Because of the diffusion of graphical editors in different application areas (e.g. business graphics, CAD, multimedia documents, etc.), graphical images are nowadays produced in large amounts. In this paper we address the problem of building a database system for graphical images (GRIM_DBMS), in which the images can be accessed by a partial description of the image content. The approach is based on a limited automatic analysis for images belonging to a domain described in advance to the system. The semantic objects, recognized in the graphical images during the analysis process, are interpreted according to the theory of evidence. The image query processing is based on special access structures generated from the image analysis process. An example demonstrates the main functions of the GRIM_DBMS in a specific application domain.

1. INTRODUCTION

When large amounts of images have to be managed in a computer system, the need to apply the database technology naturally arises. In the last decade, much of the work in the field of "image databases" appeared in the proceedings of the "IEEE Workshops on Pictorial Data Description and Management", starting from 1977, and in some other series such as "Computer Graphics and Image Processing", "Computer Vision, Graphics, and Image Processing", "Image and Vision Computing", etc.

Other sources are the work edited by Chang and Kunii [Chan81a] and the collections of papers on pictorial applications and information systems in [Blas80], [Chan80b], [Goos80]. The valuable survey of Tamura and Yokoya [Tamu84] includes insights into many actual approaches, as well as descriptions of several systems, such as the Graphics-oriented Relational Algebraic Interpreter (GRAIN), the Relational Database system for Images (REDI), the Database system of Microscopic Cell Images (IDB), etc. Another work [Chan85] presents a survey of seven commercial systems, currently available and their software capabilities: Xerox's 8010 Information System, TERA's Automated Records Management Systems, Teknotron's Systems, Scitex's Response 250, Toshiba's Document Image Filing System DF2100, CCA's Spatial Data Management System, IIS's System 600 series of software products. Different query approaches are described in works such as [Choc84], for low level image retrieval, [Rose84], for image retrieval in CAD/CAM systems, [Tang81], for alphanumeric and image data retrieval. Image query languages are described in [Chan80a], [Chan81b]. A review of approaches to machine interpretation of remotely sensed images is presented in [Tail86].

However, most of the systems appearing under the heading of image databases are often image systems without full database functionalities or database systems not directly dealing with images [Nagy85]. In fact, most of the existing systems are application specific, that is, the way in which images are stored, organized and retrieved is specific of a certain application and cannot be generalized to different applications.

The main conceptual problem in dealing with images derives from the difficulty to exactly define and interpret the content of images. Images can be very rich in semantics, but are subject to different interpretations according to the human perception of the application domain. On one hand, it is difficult to recognize the objects (with the associated interpretation) contained in an image, on the other hand is difficult to determine and represent the mutual relationships among these objects, since they form structures which vary greatly from image to image.

For the problem of image retrieval by content, one could think to apply Data Base Management System (DBMS) or Information Retrieval System (IRS) techniques. However, with respect to DBMS's, it is difficult to recognize regular structures of objects contained in images, and then organize image instances into a limited number of types, to which the interpretation is associated. This is the approach required by the strictly typed data models adopted in database systems [Tsic82]. In IRS, instead, a free formatting of text is allowed, usually respecting some loose hierarchical structuring in sections, sub-sections, paragraphs and sentences. These systems do not attempt to understand the text (unless some expert system approach is adopted), but still allow an effective retrieval on text. In fact, as opposed to image objects, they can

exactly recognize words (as ASCII patterns), on which they base their retrieval capabilities with the possible help of a thesaurus to support synonyms [Salt83]. This is possible, in case of text, because a common semantic is associated to the words used in the natural language. Hence, both DBMS and IRS approaches cannot be directly applied to image retrieval.

In addressing the problem of image retrieval of stored images, if we want to think of a system trying to do for images what DBMS and IRS do for formatted data and text, we must accept some indeterminantness, characteristic of images, and then deal with the inaccuracy introduced by this fact. In [Rabi87a] an approach based on fuzzy set theory has been applied to the analysis and description of pictorial images. Certainty factors for the recognition of objects inside the images are computed using fuzzy logic rules.

A major practical problem for pictorial images is element recognition. The first step in the image analysis and recognition process is the decomposition of the images into relevant and identifiable elements, which are the basic components which will constitute the building blocks of the image structure.

In this step, often called segmentation in pictorial image processing [Ball82], the image space is partitioned into meaningful regions, corresponding to image elements. After the segmentation, the system must recognize the tentative elements in the image, matching them with the pictorial representation of the elements to be searched. In this process, different variations as changes in size, rotation, translation (eg. using discrete Fourier transforms), can be attempted. An additional problem arises when partial element overlapping occurs in the image.

The result of this process should be the set of basic elements recognized and their relative positions. However, this is a highly computing intensive process which often requires special hardware, such as array processors, exploiting the inherent parallelism of the algorithms in order to have acceptable response times. In the end, the system might even not be able to exactly identify the single elements.

Instead, in dealing with graphical images we have the advantage that the images are not entered into the system through a scanning device but are generated by some interactive graphical editor. Therefore, the segmentation process is not necessary. Basic elements are recognized without uncertainty since they can be described in terms of the graphical primitives of the graphical editors.

In this paper, a database system for graphical images, GRIM_DBMS, is presented. This system supports the analysis and retrieval of graphical images. It is required that the images belong to a specific domain which must be described in advance to the system. A limited automatic analysis of the images is performed before storing the images in the database. This process is accomplished using a rule-based system. The interpretation of the content of the images is based on the Dempster-Shafer theory of evidence [Gord84]. An example for implementation of the GRIM_DBMS to the area of House – furnising design is also presented.

2. THE GRIM_DBMS SYSTEM

The main functions and the correspondent tools of GRIM_DBMS are illustrated in Fig. 1.

Domain Description	Image Creation, Analysis & Storage	Image Retrieval
a) Multi-functional graphical editor /function "elements & relation definition"/ b) Dialog system for filling in rules for object definition c) Multi-functional graphical editor /function "definition of class representative images"/	a) General purpose graphical editor b) Image analyzer	Query processor

Fig. 1. *Main Phases and Correspondent Tools in GRIM_DBMS*

2.1. Domain Description in GRIM_DBMS

The purpose of this phase is to describe the characteristics of the application domain of the images to be classified and retrieved. The domain description function supplies the initial information necessary for the various phases of the image analysis. It comprises: a) definition of basic elements, relations and corresponding attributes (this information will be used in the ARG-based element recognition); b) definition of production rules (this information will be used in the semantic object recognition and image interpretation); c) definition of class representation images (this information will be used in image clustering).

2.2. Image Creation, Analysis and Storage in GRIM_DBMS

A) Image Creation

For image creation, it is possible to use any graphical editor producing a representation of the image in terms of graphic primitives according to

a chosen standard (ACM-CORE, in the actual implementation). After the desired image is developed, it is stored as a file specified by the name of the image and containing the correspondent editor primitives for the graphical image. In this way, the following phase of Image Analysis is independent from the editors used in the image creation.

The system also provides a multi-functional graphical image editor, which is specialized for the specific application domain. From a table containing all the basic elements of the GRIM_DBMS application domain it is possible to select basic graphical elements and to transfer them to the drawing area of the editor using a "mouse" as pointing device. After an element is moved to the right place, the user can apply some functions allowing modifications on the element as scaling, rotation, translation and deletion. These operations can be repeated as many times as necessary for all the elements of the image. This specialized editor generates also an ARG representation of the image (see following section) and so the first phase of the Image Analysis process (i.e. Image Element Recognition) is not necessary.

B) Image Analysis

The image analysis includes four steps:

* **Element Recognition.** Since the number of the basic graphical elements (eg. polilines, curves, etc.) can be very large in a single image (in the order, of thousands) a very efficient approach is required for recognizing the basic elements which are meaningful in the application domain. They constitutes the basic symbols which compose the semantic objects in the image. In this phase, it is not possible to adopt a rule system, based on a generalized inference mechanism with back-tracking, because of its computational complexity. We need instead more efficient and specialized algorithms (with polynomial computation complexity) even if we have to pay this with a description system less rich in semantic content. For this reason we have adopted an approach based on the Attributed Relational Graphs [Eshe86].

The ARG graph is a relational structure which consists of a set of nodes and a set of branches representing the relations between the nodes, as both nodes and branches may have some attributes assigned to them.

During the Image Element Analysis, an ARG representation of the image (in terms of basic elements of the application domain, thier relationships and attributes) is obtained and stored in the same image description file.

* **Object Recognition.** The purpose of this phase is to recognize more semantically meaningful objects from the basic elements derived and organized in the previous phase. This task is accomplished by

recursively applying the production rules defined for the chosen application domain. An inference mechanism based on backward chaining tries to derive from the basic elements more general objects and to give a recognition degree to the object recognized. In this phase a generalized inference mechanism is used. Its computational complexity is acceptable now, since fewer objects (in the order of hundreds) are present in the image.

The inference process starts from production rules obtained from the ARG image representation. After this step, a sequence in the form (1) is obtained:

$$(1) \quad \{O_{1\ 1}(\mu_{1\ 1}, l_{1\ 1}), \ldots, O_{1\ s_1}(\mu_{1\ s_1}, l_{1\ s_1})\}, \ldots,$$
$$\{O_{n\ 1}(\mu_{n\ 1}, l_{n\ 1}), \ldots, O_{n\ s_n}(\mu_{n\ s_n}, l_{n\ s_n})\}$$

Such a sequence describes an image with n distinct physical objects. The unit $O_{i\ j}(\mu_{i\ j}, l_{i\ j})$ is a semantical representation of the physical object $i(i = 1, 2, \ldots, n)$ in the image in the j–th $(j = 1, 2, \ldots, s_j)$ recognition (i.e. a semantic object). $\mu_{i\ j}$ and $l_{i\ j}$ are respectively the recognition degre (RD) and the list of attributes of the i–th physical object in the j–th recognition.

* **Image Interpretation.** Using a procedure similar to Barnett's scheme [Barn81], based on the Dempster–Shafer theory of evidence [Gord84], and fully described in [Rabi87b], we convert the results obtained from the previous phase into a list of new structures containing information for each object:

$$(2) \quad \Big\{O_{1\ 1}\big([Bel(O_{1\ 1}), 1 - Bel(\overline{O}_{1\ 1})], l_{1\ 1}\big), \ldots,$$
$$O_{1\ q_1}\big([Bel(O_{1\ q_1}), 1 - Bel(\overline{O}_{1\ q_1})], l_{1\ q_1}\big)\Big\}, \ldots,$$
$$\Big\{O_{n\ 1}\big([Bel(O_{n\ 1}), 1 - Bel(\overline{O}_{n\ 1})], l_{n\ 1}\big), \ldots,$$
$$O_{n\ q_n}\big([Bel(O_{n\ q_n}), 1 - Bel(\overline{O}_{n\ q_n})], l_{n\ q_n}\big)\Big\}$$

Here $q_i \leq s_i(i = 1, 2, \ldots, n)$. The function $Bel(O_{i\ j})$ $(i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, q_i)$ is a belief function.

The belief function $Bel(O_{i\ j})$ gives the total amount of belief committed to the object $O_{i\ j}$ after all evidence bearing on $O_{i\ j}$ has been pooled. The function Bel provides additional information about $O_{i\ j}$, namely $Bel(\overline{O}_{i\ j})$, the extent to which the evidence supports the negation of $O_{i\ j}$, i.e. $\overline{O}_{i\ j}$. The quantity $1 - Bel(\overline{O}_{i\ j})$ expresses the plausibility of $O_{i\ j}$, i.e., the extent to which the evidence allows one to fail to doubt $O_{i\ j}$. The interval $[Bel(O_{i\ j}), 1 - Bel(\overline{O}_{i\ j})]$ is called belief interval.

In the expression (2), object interpretations with "low" belief (e.g., in the sense of interval mean value less than a chosen one) could be omitted.

* **Image Clustering.** The image clustering process is in principle similar to the document clustering of text documents used in Information Retrieval Systems [Salt83]. The most significant classes of images in the application domain are defined in terms of representative images, one for each class. The image interpretations are clustered by comparing them with the class representative images. After this computation, the clustering description of the image is expressed as a sequence:

$$(3) \quad \mu_1, \mu_2, \ldots, \mu_p,$$

where μ_i is the membership degree of the image to the i-th class.

C) Image Storage and Indexing

The image representation is stored in a file containing the graphical image as the sequence of the graphical primitives used for the composition by the graphical editor.

The derived image information, resulting from the analysis phase (expressed in terms of the probabilistic model as composition of objects, at different level of complexity, with the associated interval of belief) is stored in an "image header", associated to the image file. In this header, it is stored:

– A sequence, containing the image clustering description. Each term of these sequence contains the membership degree of the complete image to one of the image classes of the application. This kind of information is more synthetical, since it refers to the image as a whole.

– A sequence, containig the objects description. One and the same object may appear more times in the sequence, one for each appearance of that object in the image interpretation. This kind of information is more analytical, since it refers to the composition of the image.

Access structures (that is, the image indices) can be built for a fast access to image headers. Two type of indices are constructed:

– **Object index.** Each entry of the index is associated to a distinct object. For each object, a list is maintained. Each element of the list is constituted by a list of elements (BI, IMH), where IMH is a pointer to an image header, meaning that the object is present in that image. BI is the associated belief interval. For query processing, it is very important to maintain the list in decreasing order of BI. The order is computed using the mean values of the belief intervals.

– **Cluster index.** Each entry of the index is associated to a distinct image cluster. For each class defined in the application, a list of elements (MD, IMH) is maintained. IMH is a pointer to an image header, corresponding to an image with a non-null degree of membership to this cluster, and MD is the value of the membership degree. For query processing, it is very important to maintain the list in decreasing order of MD.

2.3. Query Processing in GRIM_DBMS

According to our query language, the user specifies a query statement of the form: RETRIEVE IMAGES <image_clause> .

The <image_clause> contains a <cluster_clause> and/or an <object_clause>.

The <cluster_clause> is a boolean combination of <cluster_predicate>s, each of the form: <class_name> <cluster_degree> .

The <cluster_predecate> indicates that the images in the database with a similarity to the named class higher than the <membership_value> should be retrieved (as requested in the boolean expression). The <cluster_clause> may be missing if the <object_clause> is present.

The <object_clause> is a boolean combination of <object_predicate>s, each of the form: <object_name> <degree_of_recognition>.

The <object_clause> must be evaluated, according to the boolean expression, taking into account only the images in the database containing those objects, named as <object_name>, with the left value of the belief interval higher than the <degree_of_recognition>. In the <object_predicate> WITH operator is envisaged, which serves the purpose of adding conditions to the attributes associated to the object.

All the stored images "not very distant" from the query statement (in a chosen sense) constitute the query answer set. With this approach, the query answers can be ordered by decreasing similarity to the query specification, so a user may limit the size of the answer and can receive a ranked output of the retrieved images. (These advantages are typical for the information retrieval techniques [Salt83]).

Since the image retrieval is not an exact process (there is no exact way

of defining the image content) and even the user may forget essential characteristics of the sought images, not one, but several non pertinent images are usually retrieved as a result of a query. The existance of relevance feed-back and query reformulation [Salt83] become essential since at any moment the user can go back to the query formulation step, if dissatisfied by the results which he is getting, and change some aspects of the query specification (usually, the values of the belief intervals).

3. GRIM_DBMS APPLIED IN A SPECIFIC APPLICATION DOMAIN

As an example of application, we choose the area of House–furnishing design. We now briefly explain the main functions of GRIM_DBMS, as applied in this field of application.

3.1. Domain Description

a) We limit the demonstration to the basic elements shown in Fig.2, with the associated attributes, and the relations shown in Fig. 3.

union (none)

intersection (none)

Fig. 3. *Relations*

b) Now, we must define the rules for object recognition. We want to make provision for the recognition of the following objects : **double_bed, table, chair, window, door, wall, room, sitting_room, bedroom, double_bedroom, livingroom, bathroom, kitchen.** We must define production rules for each of these objects. The production rules for image **room**, using Prolog syntax, are expressed in Fig. 4. The last six objects are the most complex ones to recognize. Cross, wall_inside and codify are productions which calculate the cross point of two lines, existance of a wall inside the room and the number of objects (rooms) in the image.

c) Suppose that the following images are chosen as class representative images: **block_of_flats, commercial_house, hospital_building, concert_hall.** For all these images, correspondent representations are to be obtained

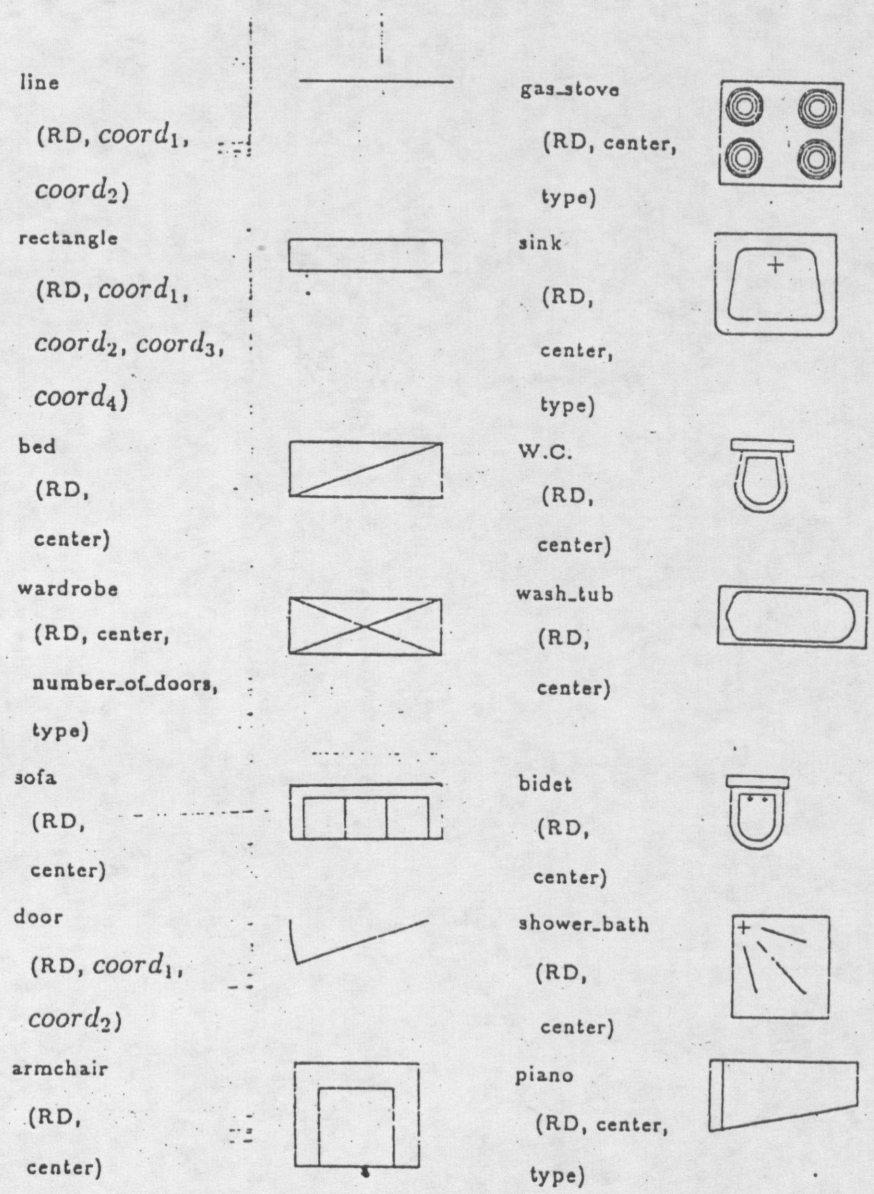

Fig. 2. *Basic Elements*

3.2. Image Creation and Storage

A) Image Creation

Let the graphical image shown in Fig. 5 be designed using the multi-functional graphical editor (specific of this application domain).


```
room (RD, V1, V2, V3, V4, N) :-
    wall (RD1, P1, P2, N1),
    wall (RD2, P3, P4, N2),
    N1 < N2,
    cross (P1, P2, P3, P4, V1),
    wall (RD3, P5, P6, N3),
    N3 > N1, N1 <> N2,
    cross (P3, P4, P5, P6, V2),
    wall (RD4, P7, P8, N4),
    N4 > N1, N4 > N2, N4 <> N3,
    cross (P5, P6, P7, P8, V3),
    cross (P1, P2, P7, P8, V4),
    not (wall_inside (V1, V2, V3, V4, N1, N2, N3, N4)),
    RD = (RD1 + RD2 + RD3 + RD4) / 4,
    codify (N1, N2, N3, N4, N).
```

Fig. 4. *Prolog rules for "room"*

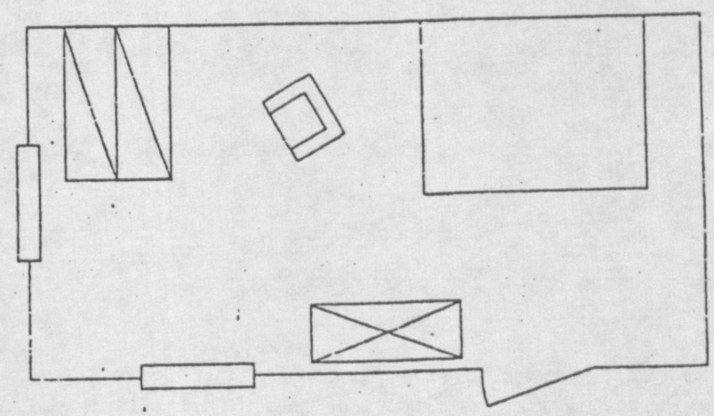

Fig. 5. *Example Image*

The corresponding ARG representation of the image, is expressed in Fig. 6.

B) Image Analysis

N (node set) $= \{n_1, n_2, \ldots, n_{15}\}$;

B (branch set) $= \{b_1, b_2, b_3, b_4\}$;

A (node attribute alphabet):

element	attributes
line	RD, $coord_1$, $coord_2$
rectangle	RD, $coord_1$, $coord_2$, $coord_3$, $coord_4$
bed	RD, center
wardrobe	RD, center, number_of_doors, type
sofa	RD, center
door	RD, $coord_1$, $coord_2$
armchair	RD, center
gas_stove	RD, center, type
sink	RD, center, type
W.C.	RD, center
wash_tub	RD, center
bidet	RD.center
shower_both	RD, center
piano	RD, center, type

E (branch attribute alphabet):

relation	attributes
union	none
intersection	none

G_N : $n_1 \longrightarrow$ line (1, [0.0, 0.0], [2.0, 0.0])
$n_2 \longrightarrow$ line (1, [0.0, 0.0], [0.0, 2.0])
$n_3 \longrightarrow$ line (1, [0.0, 6.0], [12.0, 6.0])
$n_4 \longrightarrow$ line (1, [4.0, 0.0], [8.0, 0.0])
$n_5 \longrightarrow$ line (1, [0.0, 4.0], [0.0, 6.0])
$n_6 \longrightarrow$ line (1, [12.0, 0.0], [12.0, 6.0])
$n_7 \longrightarrow$ line (1, [10.0, 0.0], [12.0, 0.0])
$n_8 \longrightarrow$ rectangle (1, [7.0, 6.0], [11.0, 6.0] , [11.0, 3.0], [7.0, 3.0])
$n_9 \longrightarrow$ rectangle (1, [-0.2, 4.0], [0.2, 4.0], [0.2, 2.0], [-0.2, 2.0])
$n_{10} \longrightarrow$ rectangle (1, [2.0, 0.2], [4.0, 0.2], [4.0, -0.2], [2.0, -0.2])
$n_{11} \longrightarrow$ door (1, [8.0, 0.0], [10.0, 0.0])
$n_{12} \longrightarrow$ bed (1, [1.2, 4.6])
$n_{13} \longrightarrow$ bed (1, [2.1, 4.6])
$n_{14} \longrightarrow$ wardrobe (1, [6.3, 0.7], 2, wood)
$n_{15} \longrightarrow$ armchair (1, [4.0, 4.3])

$G_B : b_1 \rightarrow (n_{12}, n_{13}).\ b_2 \rightarrow (n_{12}, n_3).\ b_3 \rightarrow (n_{13}, n_3).\ b_4 \rightarrow (n_8, n_3).$ - union

Fig.6. *The ARG representation of the example image*

* Object Recognition

Using the production rule base, the following sequence is obtained: sequence(1): { double_bedroom (RD = 0.9, window(RD = 1), window(RD = 1), double_bed(RD = 1), wardrobe(RD = 1, number_of_doors = 2, type = wood), armchair(RD = 1), (table(RD = 0.9), sofa(RD = 0.5)), door(RD = 1)), living_room (RD = 0.7, window(RD = 1), window(RD = 1), double_bed(RD = 1), wardrobe(RD = 1, number_of_ doors =2, type = wood), armchair(RD = 1), (table(RD = 0.9), sofa(RD = 0.5)), door(RD = 1)) }.

* Image Interpretation

After the object interpretation phase, the following sequence is obtained: sequence(2): { double_bedroom (BI = [0.73, 0.81], double_bed(BI = [1, 1]), wardrobe(BI = [1, 1], number_of_doors = 2, type = wood), armchair(BI = [1, 1]), table(BI =[0.81 0.91]), window(BI = [1, 1]), window(BI = [1, 1]), door(BI = [1, 1])), living_room (BI = [0.19 0.27], double_bed(BI = [1,1]), wardrobe(BI = [1, 1], number_of_doors = 2, type = wood), armchair(BI = [1, 1]), table(BI =[0.81 0.91]), window(BI = [1, 1]), window(BI = [1, 1]), door(BI = [1, 1])) }.

* Image Clustering

By comparing the image interpretation (sequence(2)) with all class representative descriptions, the membership degrees of the image to each class are obtained. The clustering description of the image is the followinge:

sequence(3): { block_of_flats(0.9), commercial_house(0.0), hospital_building(0.5), concert_hall(0.0) }.

C) Image Storage and Indexing

From the sequences (2) and (3), we obtain the new information to enter in the object and cluster indexes.

3.3. Query Processing

The query: "find the draughts for a room in a block_of_flats or in hospital_building, which present double_bedroom with a table and at least one wordrobe (with at least two doors, and of type "wood")", could be expressed in our query language as follows:

```
RETRIVE IMAGES (block_of_flats/0.9) OR
(hospital_building/0.7)
CONTAINING
double_bedroom/1.0 AND
```

```
(number_of_doors > 2 AND type MATCH "wood"))
```

Then the query processing is very fast since only indices are used and no information is to be extraced, in this phase, from the images.

4. CONCLUSION AND FUTURE WORK

GRIM_DBMS runs on SUN/3 workstations, under Unix 4.2. It is written in C and Quintus Prolog and uses the SUNCORE graphical package [SunC86].

The design of GRIM_DBMS is based on the experiences of a previuos prototype, described in [Rabi87b], which was based on fuzzy set techniques. This prototype was implemented on an IBM PC/AT computer and was intended for the management of business graphical images, generated by a commercial business graphical editor (IBM Graph Assistant).

We plan to apply the results obtained with GRIM_DBMS in the project MULTOS, which is part of the ESPRIT (European Strategic PRogramme for Information Technology) [Bert85]. In this project, a first prototype for the storage (based on optical media) and retrieval of multimedia office documents has already been implemented. However, in this prototype images are treated as passive components in the multimedia documents, that is, components which are retrieved as part of the document but cannot actively contribute in the retrieval process (no condition on images can be part of the query, only conditions on attributes, text and the document structure) [Bert88]. In the second MULTOS prototype, we plan to build a specialized subsystem, functionally similar to GRIM_DBMS, which will allow a higher integration of images in the document retrieval process.

REFERENCES

[Ball82] Ballard D., Brown C.,*"Computer Vision"*, Prentice Hall (1982)

[Barn81] Barnett J.,*"Computational Methods for a Mathematical Theory of Evidence"*, in Proc. 7-th Inter. Joint Conf. on Artificial Intelligence , Vancouver, B.C., pp.868-875 (1981)

[Bert85] Bertino E., Gibbs S., Rabitti F., Thanos C., Tsichritzis D.,*"Architecture of a Multimedia Document Server"*, Proc. 2nd ESPRIT Technical Week, Brussels (Sept. 1985)

[Bert88] Bertino E., Rabitti F., Gibbs S., *"Query Processing in a Multimedia Document System"*, ACM Trans. on Office Information Systems. Vol.6, N.1, pp.1-41 (1988).

[Blas80] Blaser A., (edt.) *"Database Techniques for Pictorial Applications"*, Springer Verlag, Berlin (1980)

[Chan80a] Chang N., Fu K.,*"Query-by-Pictorial-Example"*, IEEE Transactions on Software Engineering, Vol. SE-6, No. 6, pp.519-524 (Nov. 1980)

[Chan80b] Chang S., Fu K., (eds.) *"Pictorial Information Systems"*, Springer Verlag (1980)

[Chan81a] Chang S., Kunii T.,*"Pictorial Data-Base Systems"*, IEEE Computer Vol.14 (Nov. 1981)

[Chan81b] Chang N., Fu K.,*"Picture Query Languages for Pictorial Data-Base System"*, IEEE Computer Vol.14 (Nov. 1981)

[Chan85] Chang S.,*"Image Information Systems"*, Proc. of the IEEE, Vol. 73, No. 4, pp.754-764 (Apr. 1985)

[Choc84] Chock M., Cardenas A., Klinger A.,*"Database Structure and Manipulation Capabilities of a Picture Database Management System"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No. 4, pp. 484-492 (Jul. 1984)

[Eshe86] Eshera M., Fu K., *"An Image Understanding Systems Using Attributed Symbolic Representation and Inexact Graph-Matching"*, IEEE Trans. Pattern Anal. Macth. Intell., Vol. PAMI-8, No.5, pp.604-618 (Sept. 1986)

[Goos80] Goos G., Hartmanis J., (edt.),*"Pictorial Information System"*, Springer Verlag (1980)

[Gord84] Gordon J., Shortliffe E.,*"The Dempster-Shafer Theory of Evidence in Rule-Based Expert Systems"*, in *"Mycin Experiments of the Stanford Heuristic Programming Project"*, Buchanan B., Shortliffe E., (edt.), Addison - Wesley Publishing Company, pp.272-292 (1984)

[Nagy85] Nagy G.,*"Image Database"*, Image and Vision Computing, Vol. 3, No. 3, pp. 111-117, (Aug. 1985)

[Rabi87a] Rabitti F., Stanchev P.,*"An Approach to Image Retrieval from Large Image Databases"*, Proc. ACM-SIGIR Conf., New Orleans, pp.284-295 (1987)

[Rabi87b] Rabitti F., Stanchev P.,*"Graphical Image Retrieval from Large Image Databases"* , Proc. AICA Conf., Vol. 2, Trento, pp.69-89 (1987)

[Rose84] Rosenthal A., Heiler S., Manola F.,*"An Example of Knowledge - Based Query Processing in a CAD/CAM DBMS"*, Proc. Tenth Int. Conf. on VLDB, Singapore, pp. 363-370 (1984)

[Salt83] Salton G., McGill M.,*"Introduction to Modern Information Retrieval"*, McGraw-Hill (1983)

[SunC86] *"SunCore graphical package"*, SUN 3 Reference Manuale (1986)

[Tail86] Tailor A., Cross A., Hogg D., Mason D.,*"Knowledge – based interpretation of remotely sensed images"*, Image and Vision Computing, Vol. 4, No. 2, pp. 67-83 (May 1986)

[Tamu84] Tamura H., Yokoya N., *"Image Database Systems: A Survey"* , Pattern Recognition, Vol.17, No.1, pp. 29-43 (1984)

[Tang81] Tang G.,*"A Management System for an Integrated Database of Pictures and Alphanumerical Data"*, Computer Graphics and Image Processing 16, pp. 270-286 (1981)

[Tsic82] Tsichritzis D., Lochovsky F.,*"Data Models"*, Prentice-Hall, Englewood Cliffs, N.J. (1982)

(68.) * Rabitti F., Stanchev P., "GRIM_DBMS - a GRaphical IMage DataBase System", in "***Visual Database Systems***", T. Kunii (edt.) North-Holland 1989, 415-430.