Associazione Italiana per l'Informatica
ed il Calcolo Automatico
A.I.C.A.

ATTI DEL CONGRESSO ANNUALE

Volume 2

Trento
28 Settembre - 2 Ottobre 1987

GRAPHICAL IMAGE RETRIEVAL FROM LARGE IMAGE DATABASES

F. RABITTI (*), P. STANCHEV (**)

(*) Istituto di Elaborazione dell'Informazione - National Research Council - Via S. Maria 46 - 56100 Pisa (Italy)
(**) Institute of Mathematics - Bulgarian Academy of Sciences - G. Bonchev Str. 8 - 1113 Sofia (Bulgaria)

Abstract

Given the diffusion of several graphical editors, in different application areas (eg. business graphic editors, CAD editors, etc.), graphical images are nowadays produced in large amounts. In this paper we address the problem of retrieving images from large image databases, giving a partial description of the image content. This approach allows a limited automatic analysis, using a rule-based system, for images belonging to a domain described in advance. The semantic objects contained in the graphical images are interpreted according to the theory of evidence. The image query processing is based on special access structures generated from the image analysis process.

KEYWORDS: *Image retrieval, Graphical image management, Attributed relational graph, Theory of evidence, Business graphics.*

1. Introduction: The Problem of Image Retrieval

There is a growing number of application areas where digital image processing is a primary concern. Huge amounts of images in electronic form are produced in environments such as: interactive computer-aided design, geographic data processing, remote sensing of earth resources, regional economic and health data processing, cartographic and mapping applications [Chan85].

Therefore, the problem of retrieving images from large image databases is receiving increasing attention. Noticeable examples are:

- The database EIDES (ETL Image Database for Experimental Studies) contains various commonly used images stored as disk files. Its management system is designed to perform modifications, insertions, and editing of both image data and their associated secondary information and to prepare the requested image region for user's reference [Tamu80].
- Query-by-Pictorial-Example is a relational language for querying pictorial relations, like in conventional relational Data Base Management System (DBMS). In addition to the capabilities of the conventional query languages, queries can also be expressed in terms of pictorial examples through a display terminal [Chan80].
- HIPS is a system for image processing which is well known for its ever-growing library of image transformation tools implemented as UNIX "filters". It includes a small set of subroutines which primarily deal with a standardized descriptive image sequence header [Land84].
- A system for the management of an integrated database of pictures and alphanumerical data, whose major thrust is in image retrieval, is proposed in [Tang81]. The language SEQUEL is extended to serve as an interface between the users and the system: for example, data migration and data compression are accomplished by the data definition facilities in the extended SEQUEL.

Most of these systems are application specific, that is, the manner in which images are stored, organized and retrieved is specific of a certain application and cannot be generalized to different applications. In many systems, images are stored in files which are linked to other image files into structures for image retrieval and presentation according to the logic of the application [Hero80].

Other systems are based on an underlying database whose schema describes the image content and composition [Econ83]. These systems can exploit the flexibility of DBMS. In particular, the query language and the access structures implemented in the DBMS are

very powerful for retrieval operations. But they are limited by the fact that the content of the images must be described using data models which were developed for database systems rather than image systems, and so they lack the expressive power needed for images. In fact, images are inherently different from database records: these records can be divided in different classes according to their interpretation. The record structure (i.e. the schema) can be described at class (i.e. type) level. Since the ratio of instances per type is very high for database systems, the resulting storage structures and access methods are very efficient. On the contrary, each image may have its own particular structure, and a whole semantic network [Tsic82] may be necessary to completly describe each image instance.

A new application area where the problem of image storing and retrieval has been addressed is office automation. A new information object is defined, the multimedia document, where images are combined with attribute data, text, and voice [Tsic83]. Systems for the storage and retrieval of large volumes of these documents are under study [Bert85]. One of the main functions of these systems is the access to multimedia documents based on their content. However, while these systems incorporate efficient access methods for attribute data and text, they can do very little for images. Their approach is often to link images to other more structured components of the multimedia document, and then exploit the combined access to them [Tsic83].

For the retrieval of documents by content, multimedia document systems can exploit very efficient techniques borrowed from DBMS [Ullm82], when formatted data is concerned, or from Information Retrieval Systems (IRS) [Salt83], when text is concerned. Comparable techniques are not available for images.

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 perspective or 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 DBMS or 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 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 the image retrieval.

In addressing the problem of image retrieval on large volumes of stored images, if we want to think of a system doing for images what DBMS and IRS do for formatted data and text, we must accept some indeterminateness, characteristic of images, and then deal with the inaccuracy introduced by this fact. In [Rabi87] an approach based on the fuzzy set theory has been applied to the analysis and description of pictorial images. Certainty factors for the recognition of object insides the images are computed using fuzzy logic rules.

However, a major practical problem for pictorial images is the **element recognition.** The first step in the pictorial 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 (eg. icons stored in a catalog) of the elements to be searched (i.e. template elements). 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, as array processors, exploiting the inherent parallelism of the algorithms in order to have acceptable response times. At 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 graphics primitives of the graphical editors.

In spite of these advantages, the problem of retrieval has received more attention in the context of pictorial image systems rather than graphical image systems. Computer graphics systems focused almost exclusively on the problems of creation/editing and of standardization (see the GKS graphics standard [ANSI84]).

In this paper, an approach is presented for 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].

The real goal of this image analysis process is not to attempt any deep image understanding, but is to support the image retrieval process. In fact, the system allows the user to query the images, already analyzed and stored, giving some specification of their content. The image query processing is based on special access structures (i.e. image indices) which are generated when the image analysis is performed. The query specification, expressed in a linear query language, indicates the essential features of the images to be retrieved.

In the following, the three fundamental phases of image analysis, image storing/indexing, and image retrieval will be presented in detail. An example dealing with business graphics is presented.

2. Functions of the Graphical Image Management System

In a system intended for the management, in the sense of filing and retrieval, of large databases of graphical images we can identify four inter-related functions:

1) **Domain Description:** The purpose of this function is to describe the characteristics of the application domains of the images to be managed. This description is in terms of structures of the objects which should be found in the images of the application domain. This is translated in terms of rules that will be exploited in the image analysis phase (Function 2) to correctly interpret the images.

2) **Image Analysis:** The purpose of this function is to analyze the images to be managed by the system using the definitional information (particular of the application domain) specified in Function 1. This funcion is organized in the following sequential phases:

   A) **Basic Element Organization:** The basic elements, described in terms of the editor graphics primitives, are organized in a graph structure constituted by the basic objects of the application and the mutual relationships (i.e. their relative positions and the associated distinguishing attributes). This graph structure hides the peculiarities of the graphical representation of the basic objects, generated by the graphical editors, and can be understood by the body of rules of the next phase.

B) **Recursive Object Recognition:** In this phase the system applies a body of rules describing the objects, which are semantically relevant in the application domain, in terms of their composition by other objects. The rules are recursively applied: the application of a rule transforms the object graph into another in which a set of objects has been substituted by a more complex object (described in the rule). This process is applied as long as possible, leading to the recognition of all the semantic objects in the image. Since different rules can be applied to the same structure of objects, these rules can recognize different semantic objects in the same object of the image. Moreover, the rules describe the recognition degree of each object recognized, according to how well the object structure investigated matches with the rule specification.

C) **Objects Interpretation:** The result of the previous phase is a set of objects recognized in the image with associated recognition degrees. Then the theory of evidence is used to generate all the different object interpretations, for each object identified in the image, with the associated values of belief and plausibility. Attributes to object interpretations can also be generated.

D) **Image Interpretation and Clustering:** The image interpretation is constituted by the sequence of all the interpretations of the objects found in the image. It is also possible to define the distance between two images in terms of their interpretations. 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 a representative image for each class. Then images interpretations are clustered comparing them with class representative images. Image clusters can then be used in the image retrieval process.

3) **Image Storage and Indexing:** The graphical image is inserted into the database with the added information obtained by Function 2 (i.e. the image interpretation in terms of the constituting objects and the image clustering information). This added information is then used to generate access structures on image content, wich can be used for efficient image retrieval (Function 4). Access structures are constituted by:

- Indices of the semantic objects, qualified with belief and plausibility, with the associated attributes;
- Clusters of the image interpretations.

4) **Image Retrieval:** The purpose of this function is to accept queries on image content and to retrieve the images, stored in the system, which satisfy in some degree the query specification. For the image query processing, the access structures generated by Function 3 are used.

3. Image Analysis

We now describe the four phases of image analysis: basic element organization, recursive object recognition, object interpretation, image interpretation and clustering. The domain definition function is not described in detail in this paper, however in this section we will explicitely indicate the definitions, used in the various phases of image analysis, which are expected as result of the domain definition.

3.1. Basic Element Organization

In this step the initial representation of the image in terms of a graph of basic objects is generated from the graphics primitives of the whole image. The formalism which we have adopted for this purpose is the Attributed Relational Graphs (ARG) [Eshe84] [Eshe86].

An ARG is a relational structure which consists of a set of nodes and a set of branches representing the relations between the nodes. Both nodes and branches may have some attributes assigned to them. Nodes are used to represent the basic objects in the image, while their properties are assigned as attributes to the respective nodes. The relations between two basic objects are represented by attributed branches between the corresponding nodes.

In this phase, since the number of basic graphics elements present inside the image is very large, we have decided not to use a rule system, with a generalized inference mechanism, in order to avoid computational complexity involved in this approach. Instead, for the organization of basic graphic elements into basic image objects of the application, we have adopted the approach based on ARG techniques, which allow the use of object recognition algorithms with polynomial computational complexity. We pay this choice with less flexibility in the description formalism for the basic objects. In the following phase, for the recognition of more complex and semantically more rich objects, we have used a generalized inferential system, since the number of objects (and rules) in the play is a lot more limited.

The image analysis process is explained using an example dealing with business graphics (see the appendices). This example gives only an idea of the steps involved in the image analysis and interpretation process, without really going into the technical details of a more realistic case of graphical images (in this example the basic element organization phase, using ARG techniques, becomes very simple).

In Fig.1 a graphical image obtained by the business graphics editor is shown. The node alphabet used to generated the ARG representation, is given in Table 1, the branch alphabet in Table 2 and the object alphabet in Table 3. The ARG representation of this image is given in Fig.2. Notice that, because of overlapping, the third **dot_** in the **graph_line** is recognized twice with different recognition degrees.

3.2. Recursive Object Recognition

The purpose of this phase is to compose more complex objects from the basic objects recognized and organized in the previous phase. This task is accomplished by recursively applying the production rules defined for the particular application domain. We have used an inference mechanism based on backward chaining: the system tryes to recognize in the ARG graph any complex objects, using their recursive definitions.

The rules define also the degree of recognition of an object as the distance between the object implied in the rule and the object found in the image.

To define the distance, we call element either an object or a link between two objects. The distance between two elements (either objects or branches), one in the structure contained in the image graph and the other in the rule structure, is given as follows:

$$Dist = \begin{cases} \frac{|Weight(graph\ element) - Weight(rule\ element)|}{Weight(rule\ element)}, & if\ Weight(rule\ element) \neq 0; \\ Weight(graph\ element), & otherwise, \end{cases}$$

where the weigth represents the parameters, associated to the objects and it is to be given in the domain description phase.

The recognition degree (RD) of an element is given as:

$$RD = \begin{cases} 1 - Dist, & if\ Dist \leq CUTOFF; \\ 0, & otherwise, \end{cases}$$

where $CUTOFF$ is the minimal value (eg. 0.5) below which an element is not recognized at all. In this way elements with low recognition propabilities are eliminated. If we replace

one graph part with an element, using a rule, the RD of the new element is obtained as average value of RD of all the elements and branches in those graph part.

After this step a sequence in the form (1) is obtained:

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

Such a sequence denotes 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 RD and the list of attributes of the i-th physical object in the j-th recognition.

In our approach we chose a logic programming language, namely Prolog, to express the object recognition rules. Thus, we used the Prolog's inference mechanism to perform the object recognition, as part of the image analysis function.

In our example about business graphics we use the following basic objects: **frame, text_block, dot_, line_, block** and the relations: **over_text, left_text, below_text, below_dot, below_block, in_dot, in_line, in_block.**

The following objects are recognized: **poly_line, poly_block, hom_poly_line, graph_text, graph_line, graph_bar, hom_graph_line.**

The last four objects are the most complex objects which can be recognized in such business graphics images. The object **graph_text** can have the attributes **title, y_title, x_title.** The objects **graph_line, hom_graph_line** and **graph_bar** can have the attributes **legend, y_value, x_value,** etc.

The production rules for poly_line in this example are given in Appendix 1, using the syntax of Turbo Prolog.

3.3. Object Interpretation

Let us suppose that a graph portion, corresponding to a physical object in the image, has been recognized through several rules as different semantic objects each with sertain recognition degree.

Other representations of the physical object are obtained. They include the semantic objects, constructed from identical semantic objects by converting the different recognition degrees into a believ interval. For this purpose the Dempster-Shafer theory of Evidence [Gord84] is applied.

The belief function $Bel(O_i)$ $(i = 1, 2, \ldots, n)$ gives the total amount of belief committed to the object O_i after all evidence bearing on O_i has been pooled. The function Bel provides additional information about O_i, namely $Bel(\overline{O}_i)$, the extent to which the evidence supports the negation of O_i, i.e. $\overline{O}_i$. The quantity $1 - Bel(\overline{O}_i)$ expresses the plausibility of O_i, i.e., the extent to which the evidence allows one to fail to doubt O_i.The interval

$$[Bel(O_i), 1 - Bel(\overline{O}_i)]$$

is called belief interval.

Furthermore, Barnett's scheme [Barn81] is used to compute the belief interval for every object interpretation in the image.

First, all recognition degrees of the identical interpretations of a physical object $O_i (i = 1, 2, \ldots, n)$ in the image are combined. If $\mu_1, \cdots, \mu_l$ represent different degrees of object recognition, the combined support to the object is

$$p_i = 1 - (1 - \mu_1)(1 - \mu_2) \ldots (1 - \mu_l).$$

Then $Bel(O_i)$ and $Bel(\overline{O}_i)$ are calculated:

$$Bel(O_i) = K \times [p_i \prod_{j \neq i} d_j]$$

and

$$Bel(\overline{O}_i) = K \times \left([\prod_j d_j][\sum_{j \neq i} \frac{p_j}{d_j}] \right),$$

where:

$$K \times K^{-1} = 1$$

and

$$K^{-1} = [\prod_j d_j][1 + \sum_j \frac{p_j}{d_j}],$$

$$d_i = 1 - p_i$$

and $j = 1, 2, \ldots, n$.

After this step, the sequence (1) is reduced to the sequence:

(2)

$$\left\{O_{1\,1}([Bel(O_{1\,1}), 1 - Bel(\overline{O}_{1\,1})], l_{1\,1}), \ldots, O_{1\,q_1}([Bel(O_{1\,q_1}), 1 - Bel(\overline{O}_{1\,q_1})], l_{1\,q_1})\right\}, \ldots,$$

$$\left\{O_{n\,1}([Bel(O_{n\,1}), 1 - Bel(\overline{O}_{n\,1})], l_{n\,1}), \ldots, O_{n\,q_n}([Bel(O_{n\,q_n}), 1 - Bel(\overline{O}_{n\,q_n})], l_{n\,q_n})\right\}$$

where $q_i \leq s_i (i = 1, 2, \ldots, n)$. In this sequence, objects with small belief are omitted.

In Appendix 2, we give the computations of the object interpretations for the example about business graphics.

3.4. Image Interpretation and Clustering

The interpretation of an image, as result of the analysis, is in terms of the interpretations of the composing objects. Therefore, the sequence (2) is an image interpretation.

The clustering process consists in adding to the obtained image description information about the membership degrees of the image to every defined class. This is made by comparing the image interpretation with all the given class descriptions (i.e. the class centroid images). This process is similar to the clustering process in Information Retrieval Systems [Salt83].

The membership degree of an image to a class is assumed to be the inverse of the distance of the image interpretation to the class representative image. The distance between two images is defined as the vectorial distance between their interpretations. According to this, an image is represented by a vector whose elements are the objects, and the value of each element is the mean value of the beleif interval of the corresponding object in the image interpretation. The vector elements should be ordered according to some global ordering of all the objects in the domain: if some object is not present in the image the element value is zero.

After this computation, the clustering description of the image is expressed as a sequence:

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

where μ_i is the membership degree of the image to the class K_i .

In the example for business graphics the clusters could be **bar graph, line graph, pie graph, line-bar graph.**

4. Image Storage and Indexing

We will now discuss the two phases of image storage and image indexing.

4.1. Image Storage

In our approach, along with the file containing the image presentation we must also store the added information resulting from image analysis.

The image file will contain the graphical image as the sequence of the graphics primitives used for the image composition by the graphics editor. In general, they will depend on the particular editor used. In a system based on the GKS graphic standard [ANSI84], the image file will be constituted by the GKS metafile, stored as a set of segments, each containing graphic elements with the associated attributes.

4.2. Image Indexing

The complete image description, resulting from the analysis phase, is considered in terms of the probabilistic model as composition of objects, at different level of complexity, with the associted interval of belief. However, it is essential to find a suitable storage representation for this added information since the efficiency of the image retrieval process is based on it. For this purpose, it is useful to define some type of indexing on objects and associated belief intervals.

The image access information is stored in an "image header", associated to the image file. In this header we store:

- One sequence (3), containing the image clustering description. Each term of these sequence contains the membership degree of the complete image in one of the image classes of the application. This kind of information is more synthetical, since it refers to the image as a whole.
- One sequence (2). 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.** 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 vaues of the belief intervals.

- **Cluster index.** 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 maitain the list in decreasing order of MD.

5. Image Retrieval

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

The **<image_clause>** contains the **<cluster_clause>** and/or the **<object_clause>**.

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

This predicate 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 he boolean expression, taking into account only the images in the database containing those objects, named as **<object_name>**, with the lower value of the belief interval higher than the **<degree_of_recognition>**. The WITH operator, in a **<object_predicate>**, serves the purpose of adding conditions to the attributes associated to the object named as **<object_name>**.

Referring to the previous example of a business graphics, a possible query could be:

```
RETRIVE IMAGES (line graph/0.9 ) OR (line_bar graph/0.7 )
CONTAINING
(graph_text/1.0 WITH y_title MATCH "rainfall") AND
((AT LEAST 1 graph_bar/0.8 WITH
(y_value ≥ 70 AND legend MATCH "Italy")) OR
(AT LEAST 1 hom_graph_line/0.8 WITH
(y_value ≥ 70 AND legend MATCH "Italy"))
)
```

With this query the user is looking for graphical image, of type line graph or line_bar graph, containing information about rainfall, and with at least one rainfall measure for Italy higher than 70.

All the stored images having a distance lower than the required accuracy will 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 of the information retrieval techniques [Salt83]).

A browsing facility becomes essential in this approach. That is, the user should have the possibility of browsing through the retrieved images. Since the image retrieval is not an exact process (since there is no exact way of defining the image content) and even the user may forget essential characteristics of the sought images, several non pertinent images can be retrieved as a result of a query. Moreover, relevance feed-back and query reformulation [Salt83] are emphasized, 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.

6. Conclusions and Future work

A small demo-prototype has been implemented on a IBM PC/AT computer dealing with business graphics image, generated by a commercial business graphics editor (i.e. the Graph Assistant). The image analysis process has been implemented using Turbo-Prolog, under MS/DOS.

A more powerful prototype is under development. It will run on a SUN/3 workstation, under Unix 4.2. It will be written in C and Quintus Prolog and will use the SUNCORE graphic package. The enphasis will be a new multi-function editor which will allow:

- the editing of the basic objects, in order to create and edit the basic objects of the application domain;
- the editing of complex object structures, in order to generate automatically the production rules for object recognition in the application domain;
- the editing of user's defined graphical images, for the creation of new images to be analyzed by the system and then inserted into the database.

We will try to enbed in this editor the capability to define interactively the structures and rules specific of the application domain: this is right now the main challenge of our approach in graphical image analysis and retrieval.

REFERENCES

[ANSI84] ANSI X3H3/83-25r3, *"Graphical Kernel System"*, Special issue, Computer graphics (Febr. 1984)

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

[Barn81] Barnett J.,*"Computational Methods for a Mathematical Theory of Evidence"*, in Proceedings of the 7-th International Joint Conference 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)

[Chan80] Chang N., Fu K.,*"Query-by-Pictorial-Example"*, IEEE Transactions on Software Engineering, Vol. SE-6, No. 6, pp.519-524 (Nov. 1980)

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

[Econ83] Economopoulos P., Lochovsky F.,*"A System for Managing Image Data"*, Proc. IFIP Congress (1983)

[Eshe84] Eshera M., Fu K., *"A Graph Distance Measure for Image Analysis"*, IEEE Trans. Syst., Man, Cyber., Vol. SMC.-14, No.3, pp.398-408 (May/June 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)

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

[Hero80] Herot C.F.,*"Spatial Management of Data"* , ACM Trans. Database Syst. Vol. 5(4), pp.493-513 (Dec. 1980)

[Land84] Landy M., Cohen Y., Sperling G., "*HIPS: A Unix - Based Image Processing System*", Computer Vision, Graphics, and Image Processing 25, pp.331-347 (1984)

[Rabi87] Rabitti F., Stanchev P.,"*An Approach to Image Retrieval from Large Image Databases*", Proc. ACM-SIGIR Conf., New Orleans, pp.284-295 (1987)

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

[Tamu80] Tamura H.,"*Image Database Management for Pattern Information Processing Studies*", in Pictorial Information System, Goos G., Hartmanis J.,(edt.), Springer Verlag, pp.198-227 (1980)

[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)

[Tsic83] Tsichritzis D., Christodoulakis S., Economopoulos, P., Faloutsos, C., Lee, A., Lee, D., Vandenbroek, J., Woo, C.,"*A Multimedia Office Filing System,*" Proc. Ninth Int. Conf. on Very Large Data Bases (1983)

[Ullm82] Ullman J.,"*Principles of Database Systems*", Computer Science Press (1982)

element	attributes
frame	RD, x1 - the first devision on O_x, xd - next O_x devisions , xv1 - value of 1-st x-cordiuate, xvs - the x step value, ys - (the y step value)/(the first devision on O_y)
text_block	RD, number, text
dot_	RD, number, type
line_	RD, number, number_1dot, number_2dot
block_	RD, number, type

Table 1. Node alphabet

relation	attributes
over_text	RD, number
left_text	RD, number
below_text	RD, number, x_dis, y_dis
below_dot	RD, number, x_dis, y_dis
below_block	RD, number, x_dis, y_dis
in_line	RD, number
in_block	RD, number, x_dis, y_dis

Table 2. Branch alphabet

object	attributes
poly_line	RD, type, x_value, y_value, number
hom_poly_line	RD, type, x_value, y_value, number
poly_block	RD, type, x_value, y_value, number
graph_line	RD, symbol, x_value, y_value
hom_graph_line	RD, symbol, x_value, y_value
graph_bar	RD, symbol, x_value, y_value
graph_text	RD, symbol, symbol, symbol

Table 3. Object alphabet

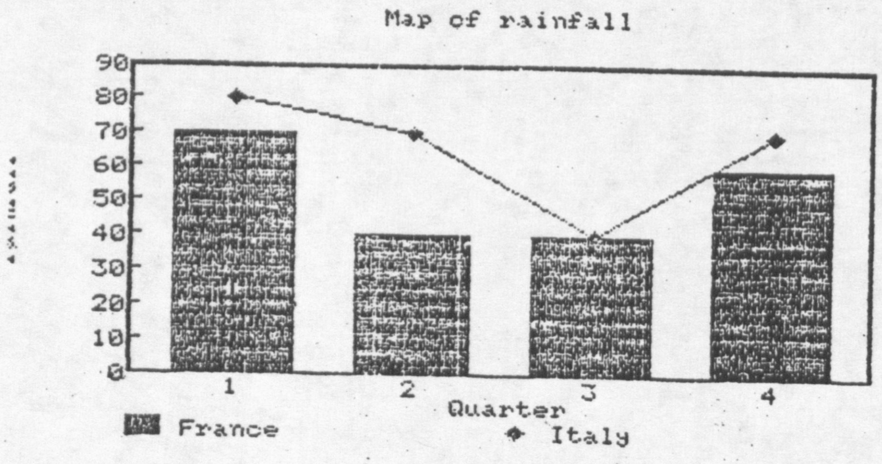

Fig. 1. Example image

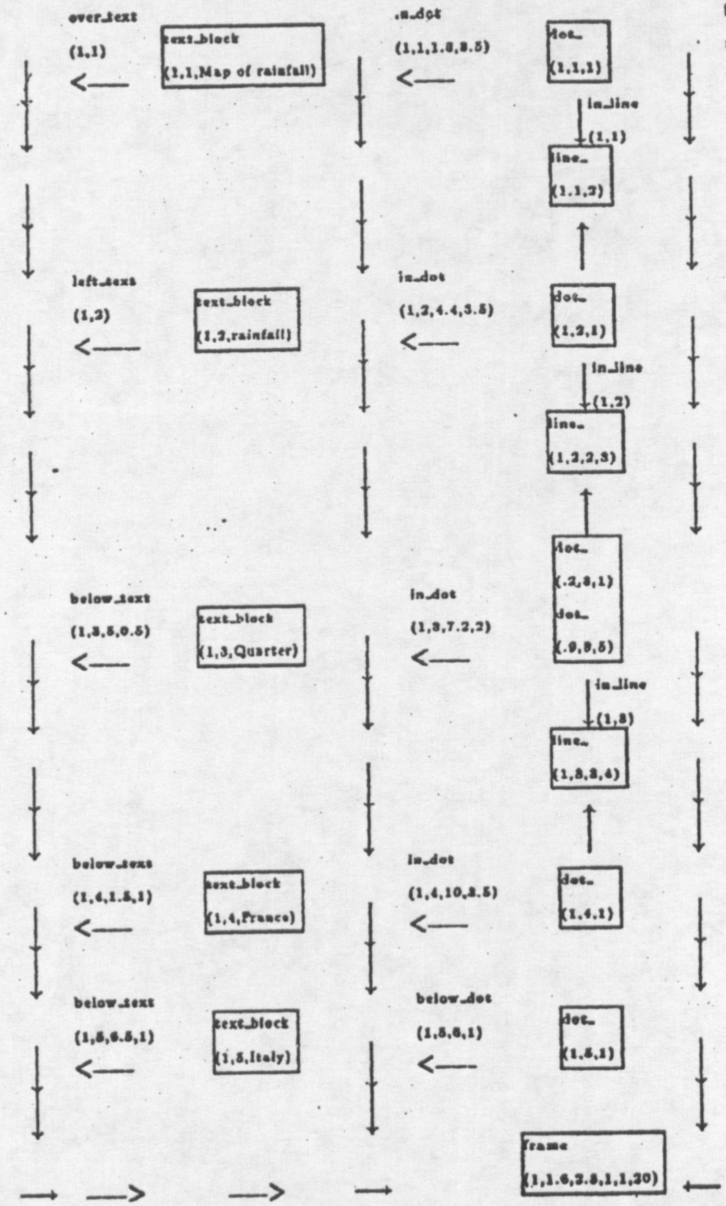

Fig. 2. ARG representation of the example image

APPENDIX 1

```
poly_line(RD, T, [A|[]], [B|[]], 1) :-
  frame(RD1, X1, XD, XV1, XVS, YS),
  dot_(RD2, 1, T), in_dot(RD3, 1, XA, YA),
  A = XVS * (XA - X1)/ XD + XV1,
  B = YA * YS,
  RD = (RD1 + RD2 + RD3)/3.
poly_line(RD, T1, [A|XV], [B|YV], N ) :-
  dot_( RD1, N, T1), not(N = 1), N1 = N - 1,
  poly_line(RD2, T, XV, YV, N1),
  frame(RD3, X1, XD, XV1, XVS, YS),
  NN1 = N -1,
  in_dot(RD4, N, XA, YA),
  line_(RD5, N3, NN1, N), in_line(RD6, N3),
  A = XVS * (XA - X1)/ XD + XV1,
  B = YA * YS,
  RD7 = (RD2 + ( RD1 + RD3 + RD4 + RD5 + RD6) / 5) / 2,
  distance(RD, RD7, T, T1).
```

APPENDIX 2

sequence(1):{ graph_text (RD = 1, title = Map of rainfall, y_title = rainfall, x_title = Quarter)}, { graph_line (RD = 0.7391, legend = Italy, x_value = [4,3,2,1], y_value = [70, 40, 70, 80]), hom_graph_line (RD = 0.98, legend = Italy, x_value = [4,3,2,1], y_value = [70, 40, 70, 80]) }, { graph_bar (RD = 1, legend = France, x_value = [4,3,2,1], y_value = [60, 40, 40, 70]) }

Belief intervals calculation:

$$p_1 = 0.7391 \qquad p_2 = 0.98$$

$$d_1 = 0.2609 \qquad d_2 = 0.02$$

$$K^{-1} = d_1 \times d_2 \times (1 + \frac{p_1}{d_1} + \frac{p_2}{d_2})$$

$$= 0.0052 \times (1 + 2.8329 + 49) = 0.2747$$

$$K = 3.64$$

$$Bel(graph\_line) = K \times p_1 \times d_2$$

$$= 3.64 \times 0.7391 \times 0.02 = 0.0538$$

$$Bel(\overline{graph\_line}) = K \times d_1 \times d_2 \times \frac{p_2}{d_2}$$

$$= 3.64 \times 0.0052 \times 49 = 0.9275$$

$$Bel(hom\_graph\_line) = K \times p_2 \times d_1$$

$$= 3.64 \times 0.98 \times 0.2609 = 0.9307$$

$$Bel(\overline{hom\_graph\_line}) = K \times d_1 \times d_2 \times \frac{p_1}{d_1}$$

$$= 3.64 \times 0.0052 \times 2.8329 = 0.0536$$

graph_line - BI = [0.0538 0.0725], hom_graph_line - BI = [0.9307 0.9464]

sequence(2): { graph_text (BI = [1 1], title = Map of rainfall, y_title = rainfall, x_title = Quarter) }, { hom_graph_line (BI = [0.9307 0.9464], legend = Italy, x_value = [4,3,2,1], y_value = [70, 40, 70, 80])}, { graph_bar (BI = [1 1], legend = France, x_value = [4,3,2,1], y_value = [60, 40, 40, 70]) }

(75.) Rabitti F., Stanchev P., "Graphical Image Retrieval from Large Image Databases", Proc. ***Annual Conf. of the Italian Computer Society (AICA)***, Trento, Italy 1987, 69-89.