A Cognitive Vision Platform for Automatic Recognition of Natural Complex Objects

Céline Hudelot and Monique Thonnat
INRIA
Orion Team
Sophia Antipolis, France
{Celine.Hudelot, Monique.Thonnat}@sophia.inria.fr

Abstract

This paper presents a generic cognitive vision platform for the automatic recognition of natural complex objects. The recognition consists of three steps : image processing for numerical object description, mapping of numerical data into symbolic data and semantic interpretation for object recognition. The focus of this paper is the distributed platform architecture composed of three highly specialized Knowledge Based Systems (KBS). The first KBS is dedicated to semantic interpretation. The second one has to deal with the anchoring of symbolic data into image data. The last KBS is dedicated to intelligent image processing. After a brief overview of the natural object recognition problem, this paper describes the three subcomponents of the platform. ***Keywords** : Cognitive Vision, Natural Object Recognition, Knowledge Based System*

1. Introduction

Our aim is to provide a generic cognitive vision platform for the automatic recognition of natural complex objects in their environment. Unlike man-made objects, natural objects have complex shapes and we have no simple and defined geometric models to describe them. Several systems have been proposed for the recognition of natural objects from images [16], [10]. They all require the cooperation of computer vision techniques for image description and data interpretation techniques for object recognition. In most of these systems, the hypothesis of isolated non overlapping objects was made. We aim at recognizing objects in their natural environment. It implies that images can contain several objects of interest, including a complex background. So, we have to use not only *a priori* knowledge of the objects but also the knowledge of the scene, i.e expected objects and relations among them.

As it was demonstrated in [17], artificial intelligence, with Knowledge Based Systems (KBS) is useful to achieve the task of image understanding. KBS have the advantage of reflecting expert knowledge. Moreover, they are easy to extend and to maintain. Automation of natural complex object recognition requires a great amount of knowledge: (1) application domain knowledge, (2) knowledge about the mapping between the scene and the image, and (3) image processing knowledge. Due to this diversity of knowledge sources and to separate them, Ossola in [14] proposed to use a distributed approach based on two KBS. The first KBS was dedicated to image processing and the second to classification. It was applied to the recognition of galaxies and zooplanctons. We decided to deepen this approach with a cognitive vision platform composed of three specialized knowledge based systems to better reflect the three types of knowledge involved in image interpretation:

- the first KBS is dedicated to semantic interpretation,
- the second one is dedicated to anchoring symbols into image data,
- the last one is dedicated to intelligent image processing.

This architecture reflects the three well known Marr's abstraction levels of computer vision [12]. But whereas the Marr's paradigm made only the distinction between the different level data types (pixel, image primitives, symbolic data), our architecture aims at separating and formalizing the specific knowledge and the specific reasoning for each level. This approach is interesting because of its independence of any particular application domain and of any image processing library.

To evaluate and validate our platform, we choose a hard image interpretation problem : the early detection of plant diseases, in particular rose diseases. The data we are working on are 2D microcopic and macroscopic images of rose parts. Objects of interest can be fungi or insects on rose leaves. Rose fungi (biological muhsrooms) are filamentous species which result in a wide range of morphologies depending on their development (figure 1): colony of ungerminated spores, germinating spores with one or more germ tubes of different lengths, filaments with various degrees of branching, entanglements of one or more filaments, pellets. Rose insects are various and complex (figure2). So, we have to deal with two hard problems : the segmentation of the different objects from their vegetal support and the semantic interpretation of data for an accurate diagnosis.

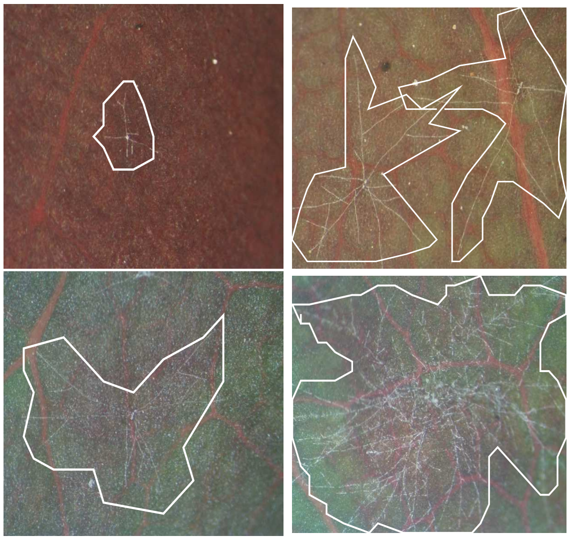

Figure 1. Different states of infection (surrounded manually in white on pictures) and different vegetal supports for a powdery mildew infection (magnification x65).

The focus of this paper is the architecture of the cognitive vision platform with detailed description of its three components. Some preliminary results concerning our application are given to illustrate the platform and to show the potential value of our approach.

2. Cognitive Vision Platform Overview

Our aim is to provide a generic cognitive vision platform for the recognition of natural complex objects in their natural environment. The problem of image interpretation is a complex problem which can be divided into more tractable sub-problems : (1)The image processing problem, i.e. segmentation and numerical description; (2) The problem of the mapping between the segmented image data and the physical objects in the scene; (3) The semantic interpretation problem. We propose a distributed architecture based on three highly specialized knowledge based systems. Each KBS is specialized for the corresponding sub-problem of computer vision:

Figure 2. Diversity and multiplicity of insects : Left. An acarid (x 65) Center. A colony of aphids (x 50), Right. Aleurodes and their eggs (x 50)

- The Interpretration KBS (see section 3) is dedicated to the semantic interpretation of data in the same way the application domain expert do.
- The Anchoring KBS is dedicated to the establishment of the correspondence between high level representations of physical objects and image data (see section 4).
- The Image Processing KBS is dedicated to the intelligent management of the image processing programs used to extract the numerical description of objects (see section 5).

Contrary to well known blackboard system for a centralized communication and control [8], our distributed architecture allows not only independence and modularity of internal data representation but also adaptated reasoning strategies for each sub-problem. Figure 3 shows the platform architecture with its different components.

3. Interpretation Knowledge Based System

This system contains all the knowledge of the application domain. Application domain experts, in our case pathologists, are the best persons to recognize objects of their domain. So our aim is to perform the interpretation in the same way experts do, using their usual taxonomy.

3.1. Domain Knowledge Formalization

As explained in [5], the design of knowledge bases is very time consuming. In order to cope with this problem,

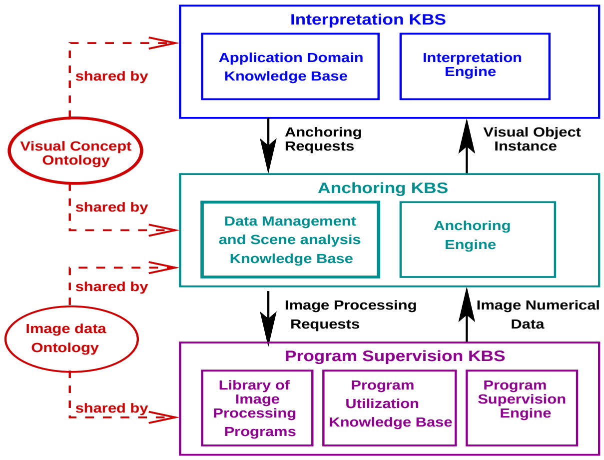

Figure 3. Global architecture of our cognitive vision platform

we can benefit from recent progress in artificial intelligence concerning ontological engineering [7]. As in the community of image retrieval where domain-dependent ontology is used to annotate images [19], we had to make good use of the ontology a domain expert uses to extract the semantics of images. In our system, the domain knowledge is formalized in a declarative model by a hierarchical structure of domain concepts associated with their subparts. Models of known objects are described by concepts organized in a tree reflecting the specialization relation between the different concepts. We choose frames as representation formalism. A domain concept is implemented by a frame with specific attributes and predefined slots. Frames are a well adapted representation scheme to describe structured concepts because they allow to describe internal properties (shape, color, ...), structural properties (subparts), relations between objects and even roles. Procedural knowledge can be attached to frames by specific facets in slots.

Moreover, in [11] the benefits of the use of an ontology of visual concepts to build the domain knowledge base are shown. Some interesting thoughts can also be found in [1]. In particular, an ontology of visual concepts is useful to reduce the gap between domain concepts and low level vision concepts. Taking advantages of these ideas, an existing visual ontology is used to build the domain knowledge base. As a result, each domain concept is described by a list of visual concepts. The used ontology [11] is structured in three main parts: spatio-temporal related concepts, texture related concepts and colorimetric concepts.

3.2. Interpretation Engine

The Interpretation engine has two main functions. It has to:

- traverse the tree of domain concepts until reaching a physical object expressed in terms of visual concepts and translate the physical object into visual object hypothesis request.
- find the class the different objects belong to by matching the visual object description sent by the Anchoring component with domain concepts of the tree.

The reasoning of the Interpretation KBS is based on the domain concept tree traversal (depth-first tree traversal). During the interpretation process, the current object to be classified is compared to each node in the concept domain tree, from the current node to the leaves. The interpretation algorithm can be seen in table 1.

From the root to the leaves (using the domain tree)
For current node
*-1-Physical object hypotheses by sending anchoring request to the Anchoring KBS (**using domain concepts**)*
-2- Wait for visual object instances from Anchoring KBS
*-3- Physical object hypotheses verification by matching between current node and visual object instances (**using domain concepts**)*
If matching is accepted
If the current node is a leaf
-4- Generate interpretation report
-5- Selection of nodes to be processed among current node brothers
Else
-6- Selection of nodes to be processed among current node sons
Else
-7-Selection of nodes to be processed among current node brothers
-8-For each selected node, go to 1
Generate interpretation report

Table 1. The Interpretation engine algorithm

4. Anchoring Knowledge Based System

4.1. Anchoring Problem

This knowledge based system is dedicated to the establishment of the correspondence between high level representation of objects and image data that correspond to the same physical objects. We call this module Anchoring KBS to make a reference to the artificial intelligence problem of anchoring. As well explained in [3], anchoring is the problem of connecting, inside an artificial system, symbols (abstract representations) and sensor data that refer to the same physical object in the external world. Anchoring is a recent research topic and only few general solutions have begun to appear in the litterature for autonomous robotics. A good

While not all anchoring requests have been processed
For each anchoring request
If anchoring request type = primitive visual object hypothesis
*-1- Image processing request generation (**using object extraction criteria**)*
-2- Wait from Image Processing KBS response in term of image features
*-3-Image processing response evaluation (**using evaluation criteria**)*
*-4-Visual object hypothesis verification and instanciation (**using data management criteria**)*
If verification-status = success
-5- Sending of visual object instance to the Interpretation KBS
Else If verification-status = incomplete
*-6- Visual Object Characterization (**using characterization criteria**)*
Else If verification-status = failure
-7- Generation of failure report
-8- Sending of failure report to the Interpretation KBS
Else If anchoring request type = scene description (compound object hypothesis
*-9- Derive spatial dependencies and schedule the order in which visual objects are analysed (**using scene analysis constraint criteria**)*
For each visual object hypothesis
Do 1 to 8
-10-Instanciation of a scene description
-11- Sending of scene description instance to the Interpretation KBS

Table 2. The Anchoring engine algorithm

introduction can be found in [4]. We believe that the problem of anchoring has a real place in the community of automatic image interpretation and scene understanding. Indeed, image interpretation can be defined as the problem of interpreting data provided by images by assigning them to a predefined semantics. A parallelism can be made between the generic anchoring module architecture for robotics described in [3] and the image interpretation architecture we have proposed. Our anchoring module is composed of two parts : a knowledge base dedicated to visual concepts as well as data management and an anchoring engine.

4.2. Anchoring Knowledge Formalization

Because of its role of intermediary between the domain concepts and the image data, this KBS shares : (1) a visual concept ontology with the Interpretation KBS and (2) an image data ontology with the Image Processing KBS. The main components of the knowledge base are implemented by frames for declarative knowledge and rules for inferential knowledge. The main concepts of this knowledge base are:

- **Visual concepts** : They aim at reducing the gap between application domain concepts and image data concepts. Moreover, they are generic concepts, independent of any application domain. Contrary to [11] where the link between symbolic visual concepts and image features is statically defined (for example, the *elongation* visual concept linked with the ratio length/height computed for the **region** image concept), we have decided to distinguish the two different types of concepts. For instance, the *Line* visual concept can be linked with the concept **ridge** image concept but also with the **ribbon** image concept. In our case, the link between visual concepts and image data concepts is dynamically defined by extraction criteria attached to visual concepts. In our representation scheme, each visual concept has two independant parts: the first part is the concept representation from a high level point of view and the second part describes it from a numerical point of view (numerical description in images). Visual concepts are divided into spatio-temporal related concepts, color related concepts and texture concepts.
- **Spatial relations** : Another important point is the representation of spatial relations. Interesting philosophical work about qualitative spatial relations can be found in [2] [6]. As well explained in [9] the particularity of relations is that they are at the same time objects with some properties and links between concepts. As in [9] we decided to represent spatial relations as specific classes. They are organized within a hierarchy of three classes: topological relations, distance relations and orientation relations. As visual concepts, they are shared with the Interpretation KBS. In our frame representation scheme, they also have two parts: a high level description using linguistic words and the translation of this relational vocabulary in constraints on images.
- **Anchoring requests** : they are hypotheses of high level visual objects in terms of visual concepts provided by the Interpretation KBS. Two types of anchoring requests exist : primitive visual object request and scene description request. A primitive object request corresponds to a simple visual object hypothesis with no parts. A scene description request corresponds to compound object hypothesis. We distinguish two types of compound visual objects (CVO): homogeneous CVO composed of a group of the same primitive VO constrained by a defined spatial arrangement (for example a network of lines) and heterogeneous CVO composed of a set of visual objects (primitive or coumpound) and a set of relations constraining them.
- **Various criteria** implemented by rules:
  - Object extraction criteria : to initiate the search for missing information by generating and send-

ing image processing request to the Image Processing KBS
  - Object evaluation criteria to diagnose the results coming from the Image Processing KBS
  - Data management criteria to manage spatial data in a bottom up point of view
  - Scene analysis criteria to manage data in the case of multi-object hypotheses in a top down point of view

4.3. Anchoring Engine

In this system, the anchoring engine performs several tasks depending on the state of the interpretation process. It has to:

- Build an image processing request with the specified constraints according to the high level description given by the Interpretation KBS.
- Select and manage image data to make the correspondence between numerical data coming from the Image Processing KBS and the current visual object in analysis (data-driven reasoning).
- Perform spatial reasoning in the case of multiple objects. This reasoning is useful to put in evidence specific geometric arrangements as network, row, circle and to constrain and guide the image information extraction
- Build a symbolic scene description and send it to the Interpretation KBS.

The algorithm of the anchoring engine is described in table 2.

5. Image Processing Program Knowledge Based System

The role of this module is to extract image features and numerical parameters describing the different objects of interest from images. It is well known that it is an hard task, especially when images to process are natural scene images. Using a specialized program is not sufficient and does not answer to our aim of a generic cognitive vision platform. Indeed, this image processing system has to process images in an intelligent way, i.e. to be able to adapt itself to different image contexts. Based on good experience in our team [13], we decided to use program supervision techniques. Several program supervision KBS have ever been done. As described in [18], they are good techniques for the semantical integration of image processing programs independently of any domains or image processing programs. Program supervision means the automation of the management of an image processing library by choosing, ordering, executing, verifying and if needed repairing programs to perform a given task. This module is composed of three parts: a library of programs, a knowledge base dedicated to image processing and a program supervision engine.

5.1. Image Processing Knowledge Formalization

Image processing knowledge structures are frames for descriptive knowledge and rules for inferential knowledge. The main concepts are described below:

- Data contain all necessary information on the problem of the end-user.
- Goals are image processing functionalities which can be processed by an algorithm or a complex treatment. They express constraints on the expected final state.
- Requests are instanciations of goals on particular data, under particular constraints. In our case, these requests are sent by the Anchoring module according to high level object hypotheses.
- Operators contain the specific knowledge to solve a goal. There are two types of operators : primitive and complex ones. A primitive operator represents a particular program and a complex operator represents a particular combination of programs. They are described by a functionality, arguments (data and parameters), rules and respectively, a decomposition in a request tree for complex operators, and a calling syntax of a program for elementary operators.
- Various criteria implemented by rules, play an important role during the reasoning, e.g. to choose between different alternatives (*choice criteria*), to tune program execution (*initialisation criteria*), to diagnose the quality of the results (*evaluation criteria*) and to repair a bad execution (*adjustement and repair criteria*).

5.2. Program Supervision Engine

The different phases of the program supervision engine are described on figure 4. The initial *planning* phase determines the best strategy to reach the end user goal (step 1). Then the *execution* phase launches the individual programs in the plan (step 2). An *evaluation* phase assesses the quality and contents of the resulting data (step 3). If the results are correct, planning can continue (step 4). If the results are incorrect (step 5), a *repair phase* can modify the plan (step 6) or re-execute the procedure with different parameters (step 7). The program supervision algorithm is described in table 3.

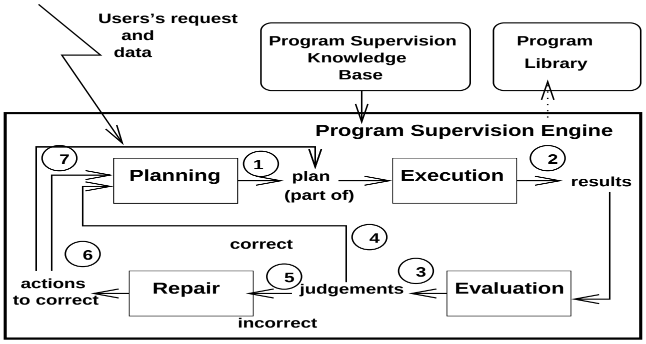

Figure 4. The Program Supervision Component

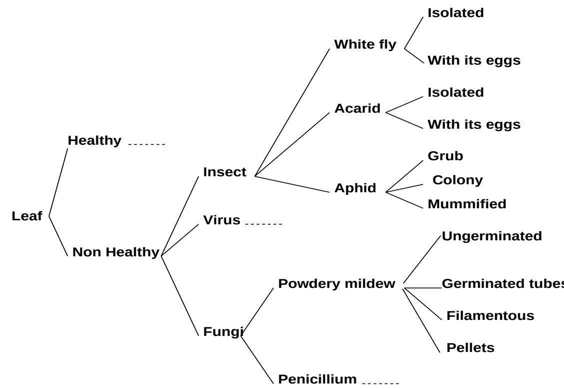

Figure 5. Part of the interpretation knowledge base for the rose disease application

-1- global matching
While not all requests have been processed
-2- selection of the request to be processed
-3- classification of the operators ***(using choice criteria)***
while the request is not satisfied
-4- selection of the best operator
-5- execution of the operator ***(using initialisation or adjustement criteria)***
*-6- assessment of the results (**using evaluation criteria**)*
While assessment is no satisfied
*-7- reparation of the operator (**using adjustement and repair criteria**)*

Table 3. The Program Supervision Engine algorithm

Domain Concept	Mycelium
Part of:	Fungi
composed of :	network of at least 2 ***connected*** hyphae
number of hyphae :	{unknown}
...	

Table 4. The domain concept Mycelium.

Table 4 and table 5 describe two domain concepts :

- *Mycelium* which is a sub-part of the domain concept *Fungi* of rose disease tree.
- *Hyphae* which is a sub-part of *Mycelium*.

6. Application on Rose Disease Recognition

6.1. Domain Knowledge

In our case, the domain ontology is modelled in a knowledge base describing observable signs and symptoms of greenhouse rose diseases. In the task of disease typing, the disease description alone is not interesting. The description of the coupling of organs and symptoms is more interesting. So, the specialization hierarchy of symptoms depends on the organ and on the plant. At present, our knowledge base describes the observable signs and symptoms of greenhouse rose leaves. For knowledge acquisition we work with two experts on plant pathologies by interviews and with two specific acquisition tools for image description called Annotate [1] and Ontovis [11]. The latter is a graphical tool which provides to experts the visual concept ontology previously mentionned. Figure 5 shows a part of the hierarchy of concepts for rose leave symptoms.

[1] http://www-sop.inria.fr/orion/ASTHMA/annotate/annotate.html

6.2. Anchoring Knowledge

The description of the domain concept hyphae involves the visual concept *Line* represented in figure 6. As we can see, some extraction criteria for the building of image processing requests and evaluation criteria for the evaluation of extracted image data are linked to the visual concept. The

Domain Concept	Hyphae
Part of:	Mycelium
has for spatio-temporal concept :	*Line*
	Line.width = {very thin, thin}
	Line.straightness = {almost straight}
Has for color concept :	*Luminosity*
	Luminosity = {bright}

Table 5. High level description of the Domain Concept Hyphae: in italic, visual concepts

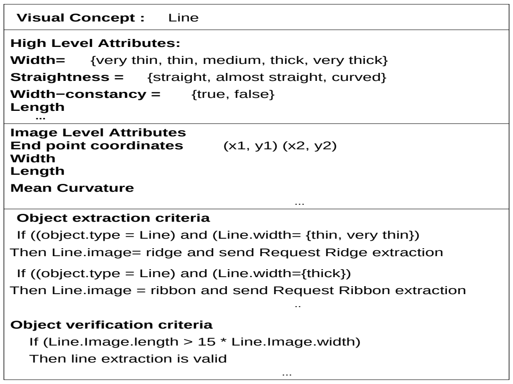

Figure 6. Example of anchoring knowledge

If for two visual objects (O1, O2) spatial description = line and line1.width = line2.width and line1.curvature is continuous to line2.curvature and O1 and O2 are close to each other
Then merge them to one visual object

Table 6. Example of data management criteria

table 6 is an example of data managment criteria useful for the processing of line objects.

6.3. Program Supervision Knowledge

Figure 7 shows an example of program utilization knowledge for the usual ridge extraction task. In the domain of image processing, we call ridge point an image point for which the intensity assumes a local maximum in the main principal curvature. For more details, see [15]. In addition with usual image processing knowledge, the program utilization knowledge base can contain some specific knowledge depending on the application. For instance, in the rose disease application the knowledge about the acquisition device is important and can have an influence on the choice, the initialisation and the evaluation of an operator.

6.4. Utilization Example

Figure 8 shows a session. By exploring the concept tree, the Interpretation KBS makes the mycelium physical object hypothesis. It sends an anchoring request to the Anchoring KBS which corresponds to the description of the physical object in term of visual concepts. In the case of mycelium, the type of the anchoring request is a scene description request (coumpound homogeneous object). It contains an unknown number of identical visual objects and a set of spatial constraints between the objects. In our case, the spatial constraint is that all the visual objects are connected in a network. The Anchoring module builds the ridge extraction request and sends it to the Image Processing module. Then

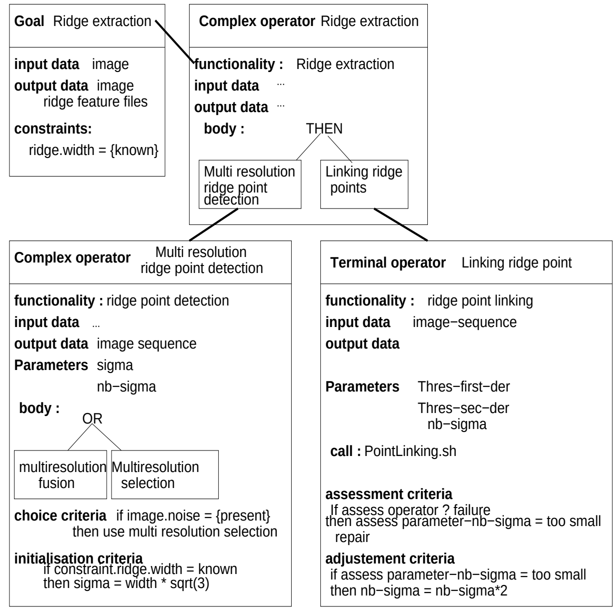

Figure 7. Example of the ridge extraction complex operator

resulting image data are managed (line grouping, matching, network generation) and a visual object instance is built and sent to the Interpretation KBS. This latter classifies the visual instance by matching and refinement (domain concept tree traversal). At the end of the session, the Interpretation KBS makes a diagnosis.

7. Conclusion

We have presented a cognitive vision platform for the automatic recognition of natural complex objects with reusable components. The goal of this paper is to formalize the different types of knowledge involved in the platform and to present the three adapted engine algorithms. We have proposed an original distributed architecture based on three knowledge based systems. This architecture divides the complex problem of image interpretation into tractable sub-problems. Each KBS is highly specialized for the corresponding step of image interpretation : semantic data interpretation, anchoring symbols into image data and image processing. This architecture separates not only the different types of knowledge involved in an image interpretation process but also the three different reasoning strategies. Moreover, our cognitive vision platform is independent of any application domain.

Our current work is the implementation of this cognitive vision platform. We have currently independent pieces of the platform : domain knowledge base, program super-

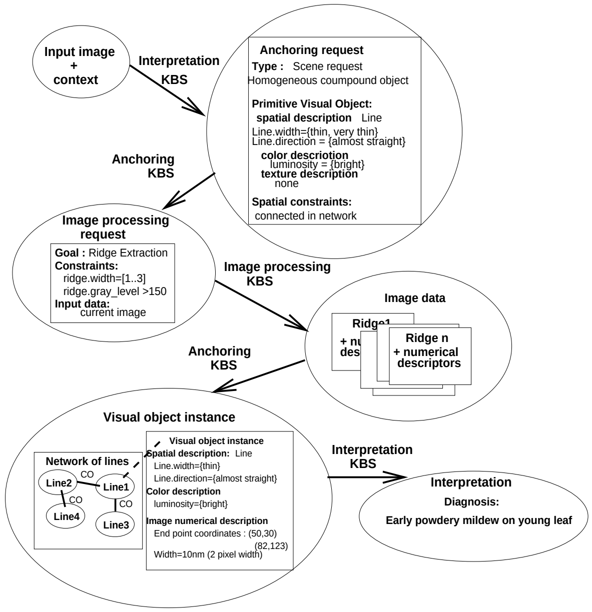

Figure 8. Utilization example of the cognitive vision platform

vision engine, visual ontology, image processing library,... Our future work will be the evaluation of this platform with different applications.

References

[1] G. Camara, M. Egenhofer, F. Fonseca, and A. Montero. What's in an image? In D. Montello, editor, *Spatial information Theory: Fundations of geographic Information Science, International Conference, COSIT*, Lecture Notes in Computer Science, 2205, Morro bay, CA, 2001. Springer-Verlag.

[2] H. S. Cohn A.G. Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, 46(1-2):1–29, 2002.

[3] S. Coradeschi. *Anchoring symbols to sensory data*. Linköping studies in science and technology, Linköping, Sweden, 1999.

[4] S. Coradeschi and A. Saffiotti. An introduction to the anchoring problem. *Preprint submitted to Elsevier Science*, 2003.

[5] R. E. Draper B., Hanson A.R. Knowledge-directed vision: control, learning and integration. *Proceedings of IEEE*, 84(11):1625–1681, 1996.

[6] P. F. E. Clementini and D. Hernandez. Qualitative representation of positional information. *Artifi cial Intelligence*, 85:317–356, 1997.

[7] T. Gruber. Towards principles for the design of ontologies used for knowledge sharing. *Formal Ontology in Conceptual Analysis and Knowledge Representation*, 1993.

[8] D. Koelma and A. Smeulders. A blackboard infrastructure for object-based image interpretation. In E. Backer, editor, *Proceedings Computing Science in the Netherlands*, 1994.

[9] N. A. Le Ber F. The design of an object-based system for representing and classifying spatial structures and relations. *Journal of Universal Computer Science*, 8(8):751–773, 2002.

[10] S. Liu, M. Thonnat, and M. Berthod. Automatic classification of planktonic foraminifera by a knowledge-based system. In *The Tenth Conference on Artifi cial Intelligence for Applications*, pages 358–364, San Antonio, Texas, March 1994. IEEE Computer Society Press.

[11] N. Maillot, M. Thonnat, and A. Boucher. Towards ontology based cognitive vision. In *Third International Conference on Computer Vision Systems*, Graz, Austria, April 2003.

[12] D. Marr. *VISION*. W. H. Freeman and Company, New York, 1982.

[13] S. Moisan and M. Thonnat. Knowledge-based systems for program supervision. In *1st International Workshop on Knowledge Based systems for the (re)Use of Program Libraries*, pages 4–8, Nov 1995.

[14] J. Ossola and M. Thonnat. A distributed architecture for automatic object recognition. In *Advances in Intelligent Data Analysis*, Aug 1995.

[15] C. Steger. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2), 1998.

[16] T. M. Strat. *Natural Object Recognition*. Springer Verlag, 1992.

[17] M. Thonnat. Knowledge-based techniques for image processing and for image understanding. *J. Phys. IV France EDP Science, Les Ulis*, 12:189–236, 2002.

[18] M. Thonnat, S. Moisan, and M. Crubézy. Experience in integrating image processing programs. In Henrik Christensen, editor, *Proceeding of the 1st International Conference on Vision Systems*, Lecture Notes in Computer Science, Las Palmas, Gran Canaria, Spain, Jan. 1998. Springer-Verlag.

[19] J. J. Y. Von-Wun S, Chen-Yu L and C.-C. C. Using sharable ontology to retrieve historical images. In *Proceedings of the second ACM/IEEE-CS Joint Conference on Digital Libraries*, 2002.