Why building knowledge-based image segmentation systems is so difficult ?

Régis Clouard, Christine Porquet, Abderrahim Elmoataz, Marinette Revenu

WHY BUILDING KNOWLEDGE-BASED IMAGE SEGMENTATION SYSTEMS IS SO DIFFICULT

Régis Clouard†, Christine Porquet†, Abderrahim Elmoataz†‡ & Marinette Revenu†

†*Groupe de Recherche en Informatique, Image et Instrumentation de Caen,*
6 Bd Maréchal Juin, F-14050 Caen, France
Phone : 31-45-27-21 Fax : 31-45-26-98
E-mail: Regis.Clouard@greyc.ismra.fr

‡*Laboratoire Universitaire des Sciences Appliquées de Cherbourg,*
Rue Max Pol Fouchet, F-50130 Octeville, France

Abstract

Building general-purpose image segmentation systems is a central concern for many researchers. The challenge is of course to propose more and more flexible and reliable systems, the behaviour of which closely depends on the task to perform and the images to process. In this paper, a knowledge-based system for the dynamic building of image segmentation applications is described. This system is based on the monitoring of a library of image segmentation operators. First, the distinctive features of the segmentation domain are studied, in order to find out which problem solving paradigm is the most relevant in this case - hierarchical planning - and seven behavioral rules for an architecture dedicated to solving segmentation problems are proposed. Our implementation is based on the Blackboard model. Cooperation and knowledge acquisition issues, which are essential in our approach, are also briefly discussed.

Résumé

La construction de systèmes de segmentation d'images non dédiés est au cœur des préoccupations de nombreux travaux de recherche. L'enjeu est évidemment de proposer des systèmes plus flexibles et plus fiable, dont le comportement dépend étroitement de la tâche à réaliser et des images à traiter. Dans ce papier, nous décrivons un système à base de connaissances pour la construction dynamique d'applications en segmentation d'images, basé sur le principe de pilotage d'une bibliothèque d'opérateurs. Pour cela, nous étudions les caractéristiques du domaine de la segmentation d'images dans le but de définir quel modèle de résolution de problèmes semble le plus adapté à la résolution de problèmes de segmentation d'images et quelles doivent être les règles de comportement qu'il doit respecter. Notre implantation du système est basée sur le modèle du Blackboard. Dans la dernière partie, nous discutons brièvement le modèle de coopération et d'acquisition des connaissances que nous avons retenu.

1. Introduction

Building general-purpose image segmentation systems is a central concern for many researchers (Crevier (1993)). The challenge is of course to propose more and more flexible and reliable systems, the behaviour of which closely depends on the task to perform and the images to process. This objective implies special problem solving architectures, enabling to integrate and bring together various pieces of knowledge, either declarative or procedural. Actually, a purely procedural approach cannot be considered because image segmentation is a complex problem, in the sense of systemic. More formally, most image segmentation problems, such as edge detection,

region extraction, shape from texture, shape from shading, surface reconstruction, belong to the mathematical class of inverse ill-posed problems (Poggio et al. (1985)). It means that there exists no unique and stable transformation function (in the sense that it is continuously depending on the observed data), that can build a specific representation of a scene, starting from any kind of observation. A little noise on data can lead to a great variability in the results. Moreover, image segmentation is in no way an optimization problem, because one cannot exhibit a generic and indisputable cost function that could be minimized. To tackle this kind of problem, one has to reduce the number of acceptable solutions, on the one hand by introducing a priori knowledge on the solution space, and on the other hand by considering segmentation processes as being decomposed into a sequence of sub-problems, that are either well-posed problems, or problems for which classical regularization methods exist.

If there are no general enough algorithms to solve any segmentation problem, on the contrary, many dedicated segmentation problems have been solved in various domains. They contributed to bring into light a great number of specific but efficient algorithms. It is thus possible to constitute a large library of operators and then to envision the building of a segmentation application as the generation of a program, through selection, parameter adjustment and linking of such operators.

In this paper, problem solving models that can be used to build knowledge-based systems for the monitoring of a library of segmentation operators are discussed and such a system is described. First, by studying peculiar features and difficulties of the image segmentation domain, we set forth seven behavioral rules that must be followed when considering suitable problem solving paradigms. Then, through the description of the system we have developed, we discuss the choices we made according to these seven behavioral rules.

2. How to monitor a library of segmentation operators

In this section, we are going to examine the peculiar features of the image segmentation domain when its comes to build a system to generate applications by monitoring a library of operators.

2.1 Monitoring a library

Generally speaking, building an image segmentation application consists in generating a program, starting from the specifications of a request, expressed by users in terms of tasks to perform on a class of images. This program is intended to produce, through successive transformations applied to input images, new images and other data that answer this request.

Let us assume that we have at our disposal some library of operators, in which each operator is accessible directly as an executable command. This library can be rather easily made by coding various algorithms that can be found in literature (e.g smoothing filter, edge detection and edge localization, region split and region merge algorithms). We consider it is a large enough to perform various kinds of treatments on various kinds of images. With such a library, a program can thus be generated by assembling basic bricks - the operators of the library - to build a parametrized graph of operators. In what follows, we are questioning ourselves about suitable problem solving paradigms for the automatic construction of such graphs of operators.

2.2 What kind of problem solving paradigm ?

In the domain of image segmentation, there exists a large amount of algorithms that can be coded as operators, together with numerous techniques and strategies that can be represented as pieces of knowledge. Of course, the major challenge is to know when to use them and what their actual effects on images are. The problem of generating segmentation programs by monitoring a library of operators is a complex problem in the sense that "the whole is more than the sum of the parts" (Simon (1969)). In fact, given the properties of operators and the laws of their interaction, it is not a trivial matter to infer the properties of the graph.

Image segmentation possesses some distinctive features related to the images that are processed, to the nature of

operators and to their linking.

First, it cannot be likened to a *combinatorial problem*, because it is difficult to decide when an image can be recognized as a suitable solution. Moreover, one cannot a priori determine parameter values for most operators. In fact, it is well-known that parameter values have a paramount influence on results (e.g. a binarization threshold) and they must be determined with as much precision as possible.

Second, it cannot be likened to a *state space searching problem*. This kind of model would suppose that descriptions of an initial and a final states are available. But, owing to their very nature, images contain an enormous amount of noisy and incomplete data and consequently, they cannot, by themselves, constitute the basis of reasoning. This can explain why reasoning paradigms such as *Means-End Analysis* or *Strip-like Search* cannot be considered in our case, all the more because the knowledge on operators is also imprecise and unreliable (Matsuyama (1989)). It is very difficult to a priori forecast the modifications induced by the execution of an operator, and it is perhaps even more difficult to define how to measure differences between two states.

One can then resort to reasoning in a *plan space* (Currie & Austin (1991)), the only acceptable paradigm when there is an infinite number of potential configurations. It implies that the reasoning is made on actions and their linking and not on input or output data.

2.3 What kind of behavioral rules for solving segmentation problems ?

More formally, in this section we set forth seven behavioral rules for an architecture dedicated to solving segmentation problems.

Rule 1: Choose a problem solving model based on actions.

Contrary to segmentation states, segmentation actions can be described accurately, and they can thus constitute the basis of reasoning. Segmentation actions represent image segmentation decisions that are applied to input images in order to produce output images. A segmentation problem can be decomposed into a sequence of tasks to perform, and segmentation techniques can then be proposed to perform each of these tasks (Clement & Thonnat (1993)).

Rule 2: Use a reasoning model based on hierarchical planning of actions.

Segmenting an image is not a linear process (Gong & Kulikowski (1994)). As a matter of fact, the processing of an image cannot generally be decomposed into a linear sequence of atomic operations, for which it is possible to tell whether they can bring closer to the solution or not. In image segmentation, one often has to switch from one data representation to another in the course of treatments. For instance, an image can be represented as a pixel matrix, a region map, an histogram. Of course, these representations are of different natures and cannot be compared : for instance how can one compare a gradient image and a region image?

Hierarchical planning for image segmentation consists in at least, three level of decisions. First, one has to choose a segmentation strategy adapted to the problem nature. Traditionally, segmentation analysis is roughly composed into three major steps: preprocessing, presegmentation and optimization. The whole process can be carried out in either a top-down, bottom-up or hybrid fashion. Second, for each step of the strategy, segmentation techniques must be determined, according to the peculiarities of the problem, shape and texture features and constraints to satisfy. And finally, effective operators are chosen and parameters are tuned.

Rule 3: Control the execution of operators by dynamic calculation of values of parameters and by focus of attention mechanisms on parts of images.

Because knowledge on operators and segmentation techniques are uncertain and unreliable, one has to execute operators so as to obtain intermediate results that can then be assessed in order to check the relevancy of the current plan. Because it is difficult to a priori forecast the precise effects of an operator on a given image, and in the same way it is difficult to find the optimal values of each parameter of this operator (Matsuyama (1989)), all the more as the quality of results are very sensitive to the values of its parameters. One has to resort to mechanisms enabling

operators themselves to compute their parameter values through optimization or simply satisfaction of some cost function.

Rule 4: Choose a library of operators mainly composed of "atomic" operators.

Atomic operators are operators, the action of which cannot or should not be decomposed further. This rule has two major advantages in the framework of library monitoring (Clouard (1993)). First, it is a means to reduce the number of parameters to be tuned. Secondly, the relatively small-grain knowledge associated to each operator also makes their selection and use easier. Moreover, one needs some special operators for combining results either in a logical, spatial or arithmetical way.

Rule 5: Adopt a hierarchical and delocated evaluation of results.

Controlling the execution of operators is not enough to ensure the relevance of the graph. The optimization of each operator does in no way mean that the whole graph will be optimized. Because evaluating the results of a segmentation process is a problem in itself, and because there are no generic evaluation functions, one has to resort to a hierarchical evaluation at each abstraction level of the planning process. So, each action produced at each abstraction level should be associated with its own evaluation functions. Moreover, the results of operators must be brought up towards upper levels and evaluated so as to take into account more and more general criteria.

Rule 6: Describe a segmentation problem as a set of tasks to perform, together with constraints and a symbolic and numeric description of the application context.

Intentions and context must be taken into account at each reasoning level, to direct choices during the planning process, control the execution of operators and select the relevant evaluation rules. Tasks describe the nature of the problem, whereas constraints define the quality of expected results. The description of the context should include three levels of information (Elmoataz (1990)): physics about image formation (e.g. type of camera, acquisition conditions), perceptual information (e.g. edge type, region type, texture...) and knowledge about the semantics of the scene (e.g. shape and size of objects and relations between objects). It is essential to notice here that, in order to remain as general as possible, the knowledge and vocabulary used should be "weeded out" of all technical references to the domain of application. That is the reason why one should restrict to vocabulary in use in image segmentation (regions, boundaries, lines, pixels...) and to mathematical vocabulary (geometry, algebra...) for describing relations and features of objects. On the contrary, notions related to the domain of application (cells, roads...) should not be mentioned just as they are but described only by means of the authorized vocabulary (shape, texture...). This is a very strong and restrictive constraint but it is the only way to exhibit knowledge that is sufficiently general and reusable. The major interest of using this restricted vocabulary is that it is common to all users. So, it can be the base of the dialogue between them.

Rule 7: Integrate users in the resolution loop to cooperate or collaborate with the system in order to find an acceptable solution.

Despite all the attention that must be paid to the knowledge acquisition process, it is unrealistic to hope that a general-purpose segmentation -segmentation system will ever be totally autonomous when trying to solve new segmentation problems in little-known or unknown domains. The only feasible solution is then to make users enter the resolution loop, by letting them cooperate (or collaborate) with the system during the resolution of a problem. The user can be asked by the system when some information about the goal or the context are missing, or when the system declares being unable to solve the given request. On the other hand, the user can interact opportunistically when he/she is not satisfied by the solution proposed by the system.

3. Conceptual model

We are now going to focus on the choices and principles we have followed to build the conceptual model of our system, in accordance with the seven behavioral rules we have just defined.

3.1 Model of knowledge

We have chosen to study the segmentation domain under five abstraction levels (Table 1) corresponding to the steps generally considered in image segmentation, starting from the problem specification, up to the selection of operators. This five levels correspond to a more detailed view of the three levels of *Rule 2*. The first two levels (Request and Task) specify the problem completely, through the determination of all the tasks to be solved. The Request represent the problem set by the user, whereas a Task defines a part of a some segmentation strategy (e.g Eliminate the background in order to extract regions of interest). The next two levels (Functionality and Procedure) are here to determine the various solutions advocated by segmentation experts, in order to solve each task. Functionalities describe the segmentation method that is selected (e.g a pixel classification method for the Task of elimination of the image background), and Procedures precise the tools that implement these functionalities (e.g a binarization based on the boundary contrast tool for the pixel classification method). The last level (Operator) ensures the instantiation of the plan with operators from the library in use.

Building the plan of actions is achieved by successive refinements of goals at one level into subgoals at the next lower level. Each level corresponds to a more or less coarse version of the solution and is decomposed into an ordered sequence of more technical subgoals. The decomposition of a goal (Fig 1.) can either be a set of subgoals that must be executed sequentially (*THEN* links), or a conjunction of subgoals, in which case the execution order is not specified (*AND* links). Input images to subgoals come either from the decomposed goal or from one of the preceding goals in a sequence. Output images of a goal are those explicitly mentioned in the decomposition of this goal.

More concretely, the Request is split up into primitive Tasks by determining an analysis strategy, reformulating and eliminating ambiguities. Tasks are transformed into Functionalities, which, in turn, are broken down into Procedures. Procedures are implemented by means of Operators, for which the values of each parameter must be calculated. In the case of Operators, for which a priori determination of parameter values is difficult, a set of potential values are calculated and the operators are then executed in a trial-and-error process with the different values so as to optimize some evaluation function, according to *Rule 3*. Images resulting from the execution of operators are brought back upwards within the hierarchy and the output images of the request constitute the final result of the resolution. The user is in charge of the final visual evaluation and can reformulate the request if he/she is not satisfied. This reformulation consists in selecting new tasks and new constraints and must be done jointly by the image segmentation expert and the domain expert, after assessing the first results produced by the system.

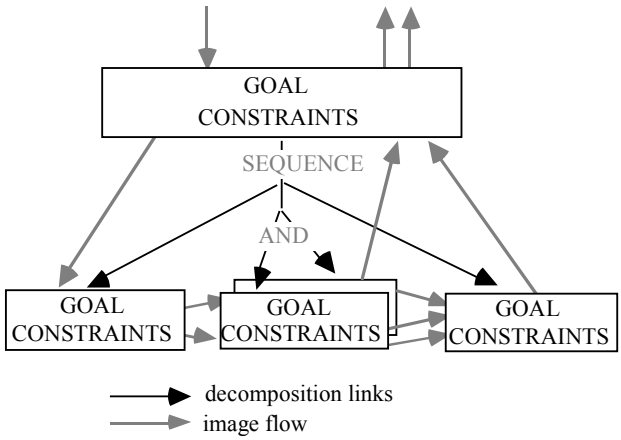

Fig.1: Decomposition of a goal into subgoals

Four types of knowledge are taken into account by the reasoning mechanism:

1. *Knowledge about the domain of image segmentation and its context,* enabling to understand

Request	*Definition of the problem to be solved, given by the user in terms of goals to be reached and constraints on these goals.*
Task	*All the primitive tasks to be solved can be found at this level. They are either deduced from the request or from the chosen analysis strategy.*
Functionality	*Functionalities describe general-purpose segmentation functions that must be implemented in order to solve some task.*
Procedure	*Procedures correspond to classical segmentation operators, but defined independently from any specific implementation.*
Operator	*Our run-time library of operators is our specific implementation of procedures. An operator is characterized by its prototype and the type and domain of each of its parameters.*

Table 1 : The segmentation domain is studied under five levels of abstraction.

the problem data in order to determine a suitable analysis strategy and to raise ambiguities when translating this problem into segmentation tasks. Tasks are our means of setting image segmentation problems.

2. *Knowledge about the expertise in image segmentation*, used for determining which techniques will be considered to perform segmentation tasks, and how results will be evaluated.

3. *Knowledge about the operators of the library*, used for selecting operators and setting the values of their parameters. This knowledge is totally independent from the implementation of the run-time library.

4. *Knowledge about the control of the resolution,* used to direct choices and to solve resolution conflicts. In most systems, this knowledge takes a procedural form. We, on the contrary, intend to express this knowledge declaratively and explicitly.

3.2 Incremental formation of the solution

In our approach, the plan is built step by step, so that it can match the current state of the solution. Each decomposition of a goal is made according to specific constraints and characteristics of the class of images.

The same kind of approach is used for assessing the quality of the plan. The plan evaluation is done in a hierarchical and delocated manner. To each goal are associated rules for assessing its own results. At the highest level, these rules are concerned with semantic features and at the lowest level with syntactic features. This principle enables to organize and distribute rules into the hierarchy.

Moreover, as a consequence of this incremental approach, it is practically impossible to forecast the plan that will come out as final solution.

3.3 Opportunistic problem-solving behavior

There are no predefined algorithms to solve a given segmentation problem, because of the size of data and the large number of alternatives to be taken into account. At each step of the resolution, several actions are feasible, so that the solution can make progress: should we decompose some goal into subgoals, assess the result of the execution of an operator, bring back up results of an evaluation or apply the evaluation rules of some other goal? Selecting the "best" action is very important for it influences directly the quality and rapidity of convergence of the solution. This falls within the competence of the control of resolution mechanism.

Choice is done dynamically, according to the state of progress of the current solution. It hinges upon general heuristics and specific focuses of attention. Our model enables to benefit from several resolution modes, either cooperative, or competitive. Several approaches are available: the goal-directed approach (choosing the actions that create important data), the action-directed approach (choosing the intrinsically important actions), the plan-directed approach (choosing an action in accordance with the current resolution strategy), either in a forward-chaining manner or in a backward-chaining manner.

To that purpose, one has to consider the data of the solution as hypotheses associated with a coefficient of plausibility and a coefficient of importance towards the current state of the solution. These coefficients are used by the control mechanism to calculate priorities in the development of the solution.

A system that combines the above-mentioned features is general, powerful and flexible and can be directly implemented by means of a Blackboard architecture (Nii (1989)).

4. Implementation

In this section, we are now going to describe the distinctive features of our system which is based on the Blackboard architecture: the database, the knowledge sources and the way we solve control problems (Clouard et al. (1993)). The Blackboard model is at once a conceptual, high-level organization of information and knowledge and a general prescription for the dynamic control and use of knowledge for incremental, opportunistic problem solving (Nii (1989)).

4.1 Syntax of segmentation requests

Goal	*a string defining the goal to be reached.*
Constraints	*a list of constraints describing quality requirements on results.*
Input images	*a list of input images.*
Output images	*a list of output images.*
Decomposition	*the set of nodes representing the decomposition of the present node.*
Result	*the path leading to output images through the decompositio of the present node.*
Evaluation rules	*if-then rules to assess results.*
Assessment	*a value telling whether the results are acceptable or not.*
Importance	*a value representing the importance of the present decision towards the current solution.*
Plausibility	*a value giving the confidence degree in the present decision related to the current solution.*

Table 2: The attributes common to all the five levels.

The input of our system is a graphical interface enabling the user to specify segmentation tasks. This interface takes the form of a hypertext through which the user can progressively refine the terms of his/her request by selecting tasks, defining constraints and describing the application context by giving values to attributes at the physical, perceptual and semantic levels.

More formally, the request is defined by the following grammar in BNF form :

request ::= <goal>*<constraint>*<context>
goal ::= (<task><arg>*)
arg ::= <object-type><restriction><value>
constraint ::= ((<restriction><criterium-><quality>)*)
context ::= (<physical><perceptual><semantic>)
physical, perceptual, semantic ::= ((<attribute><value>)*)
restriction ::= <property> <relation> <value>
criterium-to-optimize ::= <property> <relation> <value>
quality-degree ::= <property> <relation> <value>
task ::= enhance | extract | isolate | segment | ...
object-type ::= region | boundary | background | line | ...
property ::= size | form | color | texture | contrast | ...
relation ::= < | > [<= | == | ! | = | ...
attribute ::= scene-type | image-size | image-type | boundary-type | object-distribution | object-aspect | object-size | ...
value ::= <numeric> | <symbolic>

4.2 Database

Goals, input data, results, hypotheses are all stored into a global database, called the blackboard, which is organized vertically following the five levels previously detailed (Fig. 1). A plan is represented as a five-level tree of goals. Each node of the tree is connected to nodes at the next lower level by *sequence (THEN)*, and *conjunction (AND)* links. Links between goals can be seen as channels for the transmission of input and output images from one node to another.

Each level is described by a list of attributes where some are common to all five levels (Table 2). Other specific attributes are defined for each level. For instance, nodes at the operator level have a *parameter* attribute, containing the domain of possible values for each parameter of the operator, as well as a *prototype* attribute giving the usage of the operator.

4.3 Knowledge Sources

The knowledge base is divided into independent and autonomous modules (Knowledge Sources) that totally ignore one another. A knowledge source (KS) contains expertise to solve one part of the global problem in a unique way. So, there are as many KS as way to solve one given problem. A KS constitutes a link between two nodes of the blackboard, one using it as its input and the other as its output (Fig 2). KS are defined in the traditional *condition-action* style. The condition part is responsible for determining when the KS can contribute to the problem resolution. The action part acts on the solution by creating or modifying data. The knowledge base is composed of three kinds of KS :

1. *Decomposition KS* : They contain the knowledge to decompose some goal into subgoals at the next lower level. For that purpose, the description of the context and the constraints associated to the goal are used to build a set of

subgoals and to specify their constraints and the relations existing between subgoals. For each subgoals, a decomposition-KS must define the goal, the constraints, the input image or the path to get them and also the evaluation rules used to check the accuracy of the results with regard to the goals to be reached and the associated constraints. Decomposition KS are given in the following form :

TRIGGER: *∃ a task T / goal(T)=B*
CONDITION: *state(T) ≠ solved*
ACTION:
 if (*∃ an attribute y∈context / val(y)=V)*
 or (*∃ an attribute z∈constraints(T) / z=W)*
 then
  Create *decomposition(T)=(T1,T2,T3)*
 else
  Create *decomposition(T)=(T0', T1',T2',T3'))*
(where T1',T2',T3 are the same tasks as T1,T2,T3
but with different constraints).

2. *Execution KS* : There are five KS of this kind according to five execution modes: *NORMA*L execution, execution of an operator in *OPTIMIZATION* mode with the various combinations of possible values for each parameter, iterative execution (*FOR*), and repetitive execution (*WHILE* or *UNTIL*).

3. *Evaluation KS* : There is one evaluation KS per level, which is triggered on a goal of the tree when the decomposition of this goal is completed. Its action part propagates the results of the decomposition of the goal upwards and applies the associated evaluation rules. The result is judged acceptable when output images are in accordance with expectations.

4.4 Control blackboard and control Knowledge Sources

The task assigned to control is to select the next KS to be executed. The control structure consists of a control blackboard and a set of control knowledge sources. The Blackboard model is strongly inspired by BB1 (Hayes-Roth (1985)), the control blackboard being decomposed into six following levels (Table 3). The first four levels define, at each step of the resolution, the profile of the desired actions, with regard to the current state of the solution, whereas the two last levels define the profile of the actions that are actually feasible at this step of the resolution (the actions that can contribute to the progress of the solution).

The control problem is solved by control KS, following the same principles as those dealing with the domain of application. There are no differences between control-related decisions and domain-related ones. The choice of the next KS to be executed is the result of a compromise between desired KS and executable KS. The profile of the next KS to be executed is determined by general-purpose heuristics and local focuses of attention, resulting from general or specific strategies. Executable KS are put into the agenda. At each cycle, the scheduler has to decide whether to change the current strategy or to continue the development of the current solution on the domain blackboard.

Problem	*definition of the problem to be solved.*
Strategy	*cooperative or competitive strategies used to solve the problem.*
Focus	*the implementation of strategies as current goals.*
Heuristic	*general-purpose heuristics defining the profile of the desired KS.*
Agenda	*a list of KS that are candidates for execution.*
Choice	*the KS selected and executed, together with the events it created.*

Table 3: The six hierarchical levels of the control blackboard.

In the framework of planning image segmentation tasks, the general strategy we use is a breadth-first strategy. It is implemented as a sequence of focuses of attention on successive levels of the database. Specific strategies can also be considered, in order to give first priority to some parts of the graph that are judged important at this step of the resolution (e.g. a temporary depth-first strategy used during the decomposition of procedures into operators). We also can apply general-purpose heuristics to give preference to :

- control actions versus domain actions,
- intrinsically important actions, such as the execution of the evaluation KS,
- actions that work on important data,
- actions that work on data with a high coefficient of plausibility.
- actions that have a low execution cost.

5. How to cooperate

At present, three types of users can cooperate with the system: experts in the domain of origin of images, experts in image segmentation and the designer-programmer of the system itself. Furthermore, two forms of cooperation are available (Clouard et al. (1995)), the first one is essential to the knowledge acquisition process, and the second one to the resolution of applications. Consequently, our system presents two faces, whether the user chooses the "manual"option for knowledge acquisition or the "automatic" one in the case of problem solving. To this purpose, representing an action-based reasoning through a hierarchical plan offers good means of cooperation (Willamowski (1994)) .

5.1 Cooperation for knowledge acquisition

The manual option is simply a plan editor, that enables the image segmentation expert to manually build applications as they are prescribed by the domain expert. The image segmentation expert can thus create his/her own base of plans of actions. Plans are coded in a completely static manner according to the *task/method/tool* paradigm (Revenu et al. (1995)). In our case, tools correspond to operators and methods simply correspond to the links between tasks and tools. During this *procedural* integration process, segmentation techniques are specified as knowledge that is directly operational for the planning module, because the plans created "manually" share the same structure as the plans automatically built by the planning module.

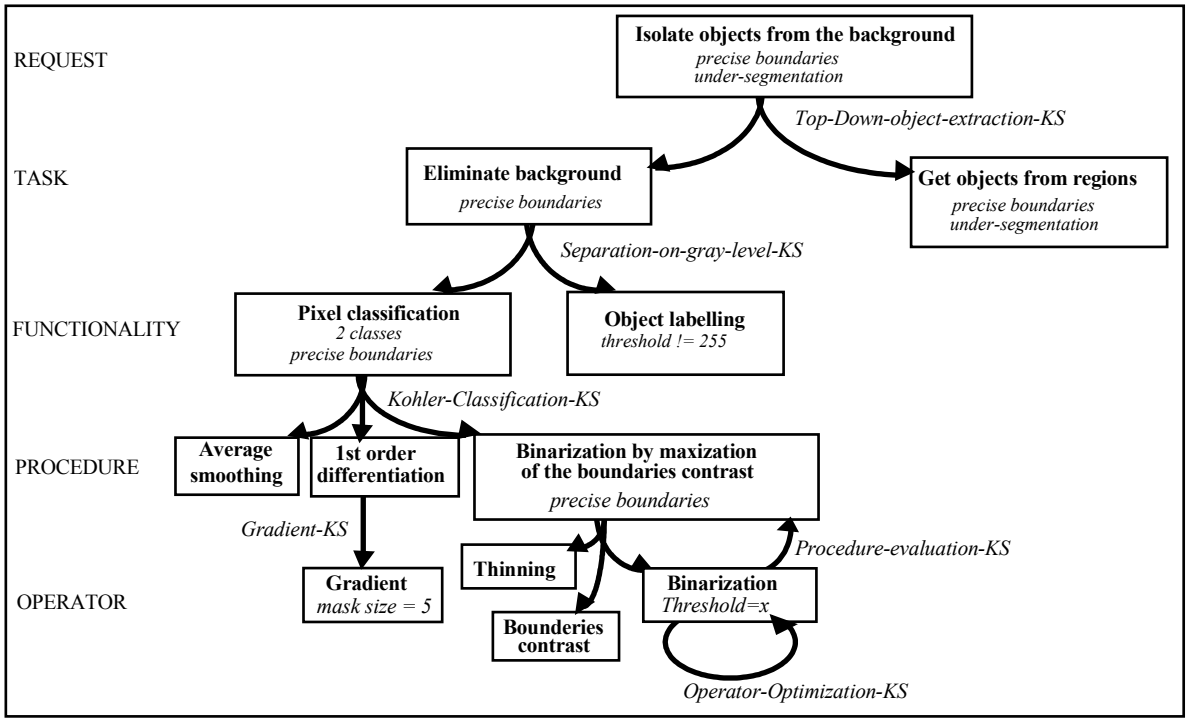

Fig. 2 : An example of a plan on the segmentation blackboard

Knowledge acquisition thus requires an intermediate stage when one has to specify the set of criteria that lead to the decomposition rules of each node of the plan, i.e. type of decomposition (*AND/THEN*) and principally, application and non-application conditions for this decomposition. Then, once a first integration stage is over, the system is used as an experimentation tool when several plans related to the same request can be tried and tested, in order to evaluate their performance, to find reasons for their eventual failure or to propose improvements

5.2 Cooperation for problem-solving

The automatic option corresponds to a planning process involving cooperative work between the user and the reasoning module. In this automatic mode, interventions are on the system's initiative when data are missing (by asking them to the domain expert), and on the user's initiative when he/she wants to get some explanations or to modify previous choices. When an intervention is on the system's initiative, it is the system's opinion that prevails and in all other cases, the user's opinion is prevailing.

The system only appeals to the user in two specific cases: when an input request is considered as incomplete and when visual human assessment is needed.

6. Conclusion

Image segmentation is a typical domain where one has to develop systems establishing a cooperation between several high-level expertises. The resolution of a segmentation problem involves three categories of experts: experts from the domain of application (biologists, geographs), experts in image segmentation and designer-programmers. Domain experts are required to express clearly and exhaustively the purpose of their application, and also to provide several criteria that will be used during the a posteriori evaluation of results. Segmentation experts have to propose some kind of answer in terms of plans of actions adapted to the specific features of the request and the nature of images. Designer-programmers finally have to provide a framework for an efficient implementation.

Rather than aiming at developing a general-purposed system, we are interested in setting up a workbench directed towards the cooperation between various agents involved in the system so as to provide assistance to the working out of some kind of computational theory of segmentation, analogous to Marr's vision theory. This approach is motivated by the development of real-size applications, while being concerned by generic principles. We are addressing issues such as:

- The development of concrete applications with classical methods, in order to precise the nature of real problems and to establish an efficient dialogue between segmentation experts and domain experts, that can be based on concrete results.
- The specification of a software architecture favoring cooperation. The objective is to get a system that can manage knowledge, i.e. to represent it and make it operational. It must be a framework for the development of applications and for the acquisition of know-hows and knowledge on techniques and computer tools. Although there is still a lot of work to do, the blackboard-based system we have described in this paper, starts to bring satisfactory answers to some of these objectives.

Our research takes place within the framework of the "Image Processing and Analysis" Center of Caen. This center covers a wide domain of applications, dealing with biomedical, as well as material or geological images. It constitutes a privileged field of investigation in order to validate and refine the ideas we are advocating and to make various experts cooperate.

References

Clément, V., M. Thonnat (1993). A Knowledge-Based Approach to Integration of Image Processing Procedures, *CVGIP : Image Understanding*, 57, 2 : 164-184.

Crévier, D. (1993). Expert systems as design aids for artificial vision systems : a survey, *In Proc. of the SPIE's international symposium*, Boston, USA, *2055* : 84-96.

Clouard, R., C. Porquet, A. Elmoataz, M. Revenu (1993). Resolution of image processing problems by dynamic planning within the framework of the Blackboard model,

In Proc. of the SPIE's international symposium, Boston, USA, 2056 : 419-429.

Clouard, R., M. Revenu, A. Elmoataz & C. Porquet. (1995). A Software Workshop for Knowledge Acquisition in Image Processing, *In Proc. Intern. Workshop on the Design of Cooperative Systems,* Juan-Les-Pins, France, 298-312.

Currie, K., T. Austin (1991). O-Plan: the open planning architecture, *Artificial Intelligence*, 52:49-86.

Elmoataz, A. (1990). *Mécanismes opératoires d'un segmenteur d'images non dédié: définition d'une base d'opérateurs et implantation*. Thèse de l'Université de Caen.

Gong L., C. A. Kulikowski (1994). VISIPLAN : A Hierarchical Planning Framework for Composing Biomedical Image Analysis Processes, In *Proc. of the International Conference on Vision and Pattern Recognition*, 718-723.

Hayes-Roth, B. A blackboard architecture for control, *Artificial Intelligence*, vol(26), 1985, pp 251-321.

Matsuyama, T. (1989). Expert Systems for Image Processing: Knowledge-based Composition of Image Analysis Processes, *Computer Vision, Graphics and Image Processings,* 48: 22-49.

Nii, H. P. (1989). Introduction, *Blackboard architecture and applications*, Academic Press, pp xix-xxix.

Poggio, C. Koch, and V. Torre, "Computational vision and regularization theory", Nature, Vol. 317, p 314-319, 1985.

Revenu, M., R. Clouard, A. Elmoataz & C. Porquet, (1995). Atelier Logiciel d'acquisition et d'intégration des connaissances en Traitement et Interpretation d'images : Une approche méthodologique. *In Proc. of the 10e journées Acquisition, Validation et Apprentissage des connaissances*, Grenoble, France, Avril, 315-328.

Simon, H. A. (1969). *The sciences of the Artificial*. the M.I.T press, Cambridge, USA.

Willamowski, J. (1994). Modélisation de tâches pour la résolution de problèmes en coopération système-utilisateur. *In Proc. 9 congrès RFIA*, Paris, France, 305-316.