CBR for the management and reuse of image-processing expertise: a conversational system

Valérie Ficet-Cauchard, Christine Porquet, Marinette Revenu

CBR for the management and reuse of Image Processing expertise: a conversational system

Valérie FICET-CAUCHARD, Christine PORQUET & Marinette REVENU

GREYC-ISMRA - 6 Bd du Maréchal Juin - F14050 CAEN cedex FRANCE
tél: +33 (0)2-31-45-27-21 fax: +33 (0)2-31-45-26-98 e-mail: Valerie.Ficet@greyc.ismra.fr

Abstract. The development of an Image Processing (IP) application is a complex activity, which can be greatly alleviated by user-friendly graphical programming environments. Our major objective is to help IP experts reuse parts of their applications. A first work towards knowledge reuse has been to propose a suitable representation of the strategies of IP experts by means of IP plans (trees of tasks, methods and tools). This paper describes the CBR module of our interactive system for the development of IP plans. After a brief presentation of the overall architecture of the system and its other modules, we explain the distinction between an IP case and an IP plan, and give the selection criteria and functions that are used for similarity calculation. The core of the CBR module is a search/adaptation algorithm, whose main steps are detailed: retrieval of suitable cases, recursive adaptation of the selected one and memorization of new cases. The system's implementation is presently completed; its functioning is described in a session showing the kind of assistance provided by the CBR module during the development of a new IP application.

Keywords: Case-Based Reasoning, Image-Processing, application reuse, interactivity, knowledge management, recursive algorithm.

1. Introduction

We are doing research work in the design of an interactive system that can provide assistance during the working out of Image Processing (IP) applications; the system's architecture has been detailed in (Ficet-Cauchard *et al.*,1998). Our system is composed of several modules dealing with the tuning out of IP applications through interactive acquisition and representation of IP knowledge coming from IP experts, the execution of such IP applications and the reuse of applications following a Case-Based Reasoning approach (CBR). This paper is dedicated to a detailed description of the CBR module: in particular, our description of IP cases, similarity calculation between two cases and recursive search/adaptation algorithm are presented and discussed.

In section 2, the framework of our research is briefly presented along two axes: our objectives with regards to IP and our modeling of IP application. Sections 3 and 4 are entirely dealing with the CBR module: first, our definition of an IP case and the functions used for similarity calculation are given (section 3). Then the search/adaptation algorithm is described and explanations are given about the process of case selection, recursive adaptation of the selected solution and memorization of new cases (section 4). Finally, a complete session showing how to use the CBR module for developing an IP application is described in section 5.

2. Research framework: the TMT model

Our primary objective is to represent and structure the knowledge of different IP experts so as to enable knowledge share and reuse. To achieve such a goal, we are advocating for an interactive system enabling knowledge acquisition from IP experts, as it comes to the fore through the development of IP applications. In this section, our approach for the building of applications is presented; we describe our model for the representation of applications and briefly give an idea of the functioning of two essential modules of the system: the interactive creation module and the execution module.

2.1. Representation of applications by hierarchical plans

Our approach to the development of IP applications is based on the smart supervision of libraries of operators. An operator is a program that performs one basic operation on one or several images. It takes as inputs the image(s) to be processed as well as parameters and produces as outputs one or several images as well as numerical and/or symbolical results. With such libraries, the building of an application then "simply" consists in linking operators and tuning their parameters. Users can thus stand back from computer codes and perform programming at the "knowledge" level.

A first category of systems that make the use of such libraries easier are the graphical programming environments. Such systems (IRIS, 1993) (Rasure, Kubica, 1994) help users select and organize operators into sequences, but they do not enable to explain, nor model their reasoning. It is thus difficult to reuse solution elements that were previously built for other applications.

A second category of IP systems that use program libraries are the automatic planners such as OCAPI (Clément, Thonnat, 1993) or BORG (Clouard *et al.*, 1999), the planning system developed within our research team. Such systems are based on an explicit knowledge representation enabling reuse. However the knowledge is rather represented for computational purposes than for the user's understanding, who little intervenes during the search of a solution.

In order to take advantage of the interactive nature of graphical programming environments, our system should enable users to graphically select and link IP operators. But it should also give them a means to represent and make explicit the reasoning that has led them to this series of operators, with an objective to reuse the implemented strategy, in the same way as automatic systems do.

However, a real-size application can lead to sequences of up to tens of operators. In order to represent the reasoning associated to such sequences, we suggest to use a representation based on trees of tasks, that we call "IP plans". Such trees correspond to hierarchical decompositions of problems into sub-problems, each problem or sub-problem being related to an IP task. As is shown in figure 1, a plan not only represents the linking of IP operators corresponding to the leaves of the tree, but also all the reasoning necessary for the creation of such a linking, which is represented by IP tasks schematized as gray boxes.

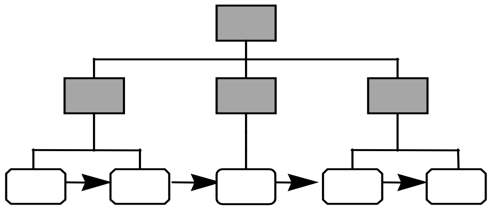


Fig. 1: representation of an IP plan

2.2. The TMT model

In our system, IP plans as well as control tasks (dealing with plan management and system control) are uniformly represented within the “task – method – tool” model. In this model, a task represents a goal or sub-goal; a method describes a know-how, it specifies how a task can be performed; a tool reifies a computer code (IP operator, Lisp or C function) in conceptual terms with a link to the code enabling to run it. There exist two types of methods: “terminal” methods (fig. 2a) that achieve a task by calling to a computer code through the medium of a tool and “complex” methods (fig. 2b) that decompose a task into sub-tasks by means of a “THEN” tree. Finally, as there may exist several strategies to solve an IP problem, a task can be associated to several methods (fig. 2c) by means of an “OR” tree, the choice of the method to be applied being made at the time of execution.

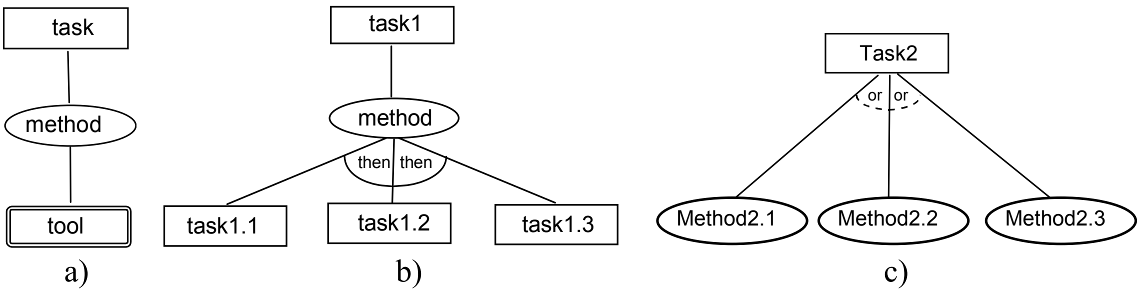


Fig. 2: various possible links between tasks, methods and tools

2.3. System’s functionalities

Our system is provided with a graphical interface, in which several functionalities have been defined for the interactive construction and the interactive execution of IP applications. In particular, they include the visualization of applications as trees of tasks, so that users can study the reasoning associated to any given IP plan.

In order to create a new IP plan, the user has to define his/her tasks and tools by filling in fields in appropriate windows; he/she can link then by defining methods and data flows between tasks and sub-tasks or tasks and tools. There are three ways for specifying the way to get the values of parameters for tasks and tools: computed from another task or tool, fixed once and for all, or to be required from user.

When they want an application to be executed, users simply have to select the root task of the corresponding plan in a menu and the plan is immediately visualized on screen as a schematic tree of tasks. The plan can then interactively be executed: users are required to choose between methods when several methods exist to perform some given task, and also to provide values for "user" parameters. Once the execution is completed, they can have access to any information about tasks and tools that have actually been executed and moreover, visualize any intermediate image in order to assess critical points.

In addition to the creation and execution functionalities that have just been described, the third and most original functionality integrated into the system consists in a second mode for creating applications through CBR. The CBR part is here to help users reuse knowledge already stored into the system, by providing means for memorizing all interesting cases and retrieving the best one. The advantages of CBR in domains characterised by a weak or ill-structured theory, such as the IP domain, are manifold:

- representing exceptions,
- allowing the use of missing or noisy data,
- solving a complex problem, through interactions between solutions of more simple problems,
- dynamic learning.

The corresponding CBR module is detailed in the next two sections.

3. Case representation and similarity

A case is broadly composed of two parts: description of the solution and description of the problem. In our system, a solution is represented as a TMT tree, which can be accessed through its root task. In Case-Based Planning (Prasad, 1995) (Veloso *et al.*, 1996) or Case-Based Design (Smyth, 1996), a solution is generally built by combining parts of several plans coming from several cases. In order to make this kind of design possible, we have decided to associate several cases to one single plan: the first case is associated to the root task of the plan, the others to some sub-tasks of the same plan, that are considered as representative of specific IP techniques. In the example of figure 3, cases are associated to tasks Ta1, Ta2 and Ta4 that correspond to some specific strategy in IP; by contrast, no case is associated to tasks Ta3, Ta5, Ta6, Ta7 and Ta8.

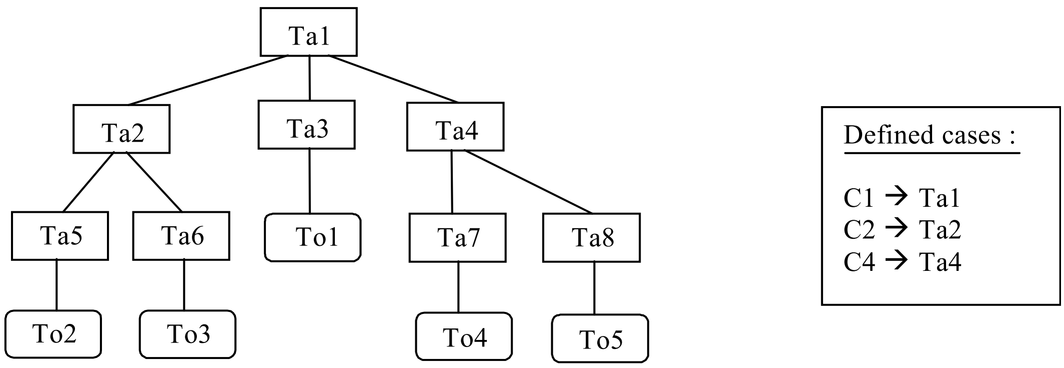


Fig.3: association of a set of cases with a TMT plan

The problem's description is made thanks to a set of discriminative criteria, which have been found out from a thorough study of the IP domain. Results of this study are presented in section 3.1; the similarity functions for comparing cases are described in section 3.2.

3.1. Criteria for case selection

The finding out of a relevant set of similarity criteria enabling to characterize an IP problem is based, on the one hand, on a study of IP systems detailed in (Ficet-Cauchard, 1999) and, on the other hand on the study of books and Ph.D. dissertations dedicated to IP techniques (Elmoataz, 1990) (Russ, 1995).

The major issue is here to choose an indexing vocabulary that can be shared and accepted by any IP programmer. Except for low-level actions (corresponding to operators from an IP library), there really exists no consensus on IP terms. In particular, this can be explained by difficulties to cut oneself off from the domain of application (most IP programmers work on one type of application at a time and thus only use terms from their current domain of application).

The criteria we put forward come from a classification of the most often encountered terms used to describe IP actions and data. We have made a distinction between two broad categories of criteria: criteria related to the task definition and criteria related to the image description.

Criteria related to the task definition

This first category includes data related to the operation performed by a task and to its position in the plan in relation to other tasks. Such criteria include IP type or phase, problem definition and abstraction level.

IP type or **phase** broadly corresponds to the type of problem that is solved by a task. According to the task's abstraction level, one can take into account:

- either the IP type: the root task of a complete plan defines a high-level processing, which belongs to an IP type (*detection*, *segmentation, classification,…*),
- or the IP phase: each sub-task of a plan defines one part of the complete processing, which corresponds to one specific step (*pre-processing*, *seed determination*, *region determination,* …).

The various IP phases correspond to a vertical division of the plan (fig. 4); for some types of problems, some phases may be optional.

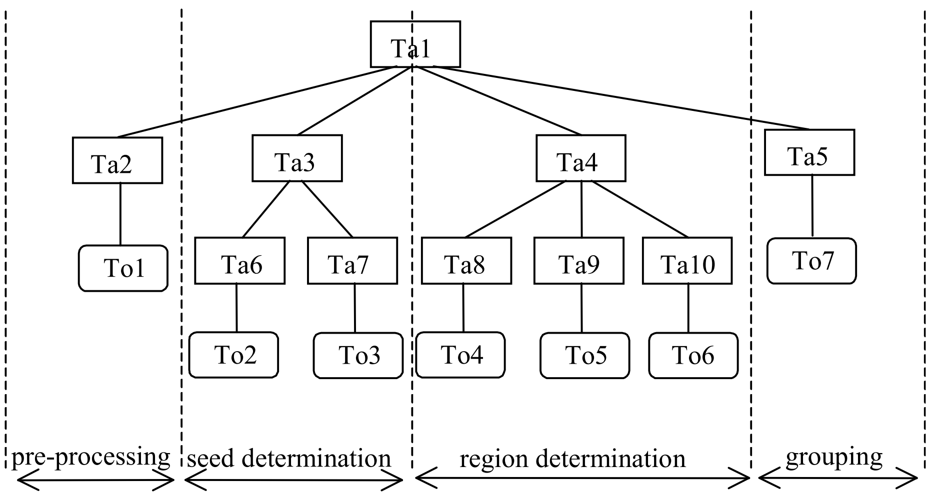


Fig.4: vertical division of a plan solving a segmentation problem

The **definition of the problem** is composed of a set of keywords selected among three pre-defined lists: 1. a list of **verbs** describing the operations performed by the task (*detect*, *classify*, *binarize, smooth, …*), 2. a list of **nouns** corresponding, either to objects on which the action is performed (*contours*, *regions, image background, …*), or to IP techniques (*region growing, region division, …*) and 3. a list of **adjectives** qualifying, either the objects on which the action is performed *(small, local*, …), or the action itself (*partial*, *strong, …*).

As can be noticed in previous examples, the vocabulary from these three lists of keywords is completely independent from the domain of application.

Finally the **abstraction levels** that correspond to a horizontal division of the plan (fig. 5) are based on the abstraction levels of the automatic planner BORG (Clouard *et al*., 1999).

- Tasks belonging to the *intentional level* answer question such as "what to do ?" and deal with IP objectives.
- Tasks belonging to the *functional level* answer questions such as "how to do ?" and refer to some IP technique, leaving aside technical constraints related to their implementation.
- Tasks belonging to the *operational level* answer questions such as "by means of what ?" and represent IP technical know-how that can be implemented as algorithms.

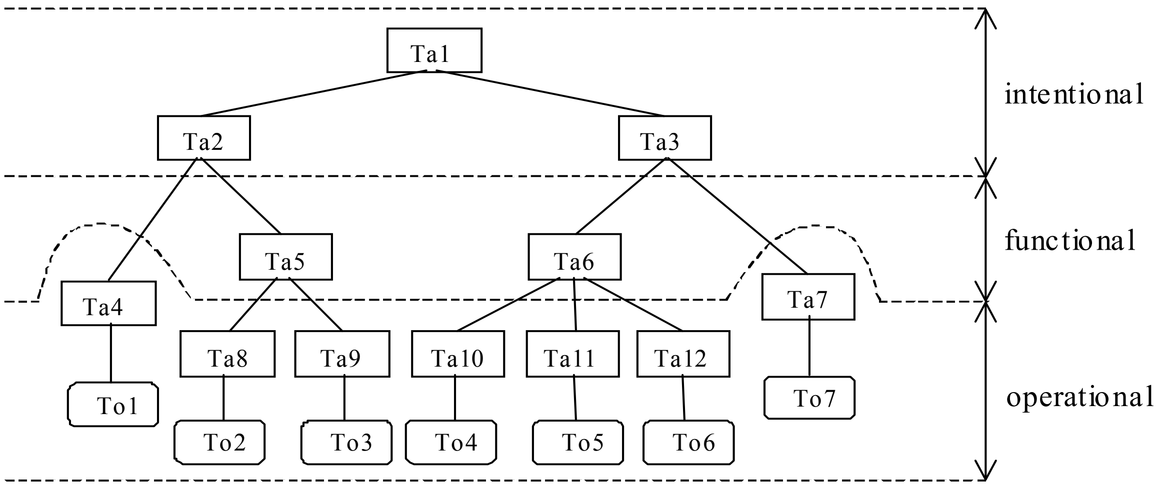


Fig.5: horizontal division of a plan

Criteria related to the image description

Among the criteria related to the context of images, some correspond to physical knowledge (related to image formation) and describe image quality (e.g. **type of noise**, **amount of noise** and **quality of contrast**). These criteria are of paramount importance for the choice of the pre-processing steps.

Other criteria rather correspond to perceptual knowledge (symbolic description in terms of visual primitives). They include the **presence** or **absence** of an image **background** and the **aspect of objects** (*homogeneous gray level, light color, texture, thick boundaries, …*).

The third group of criteria corresponds to semantic knowledge (scene analysis and components of the scene) and describes the appearance of what is to be detected, but in abstract terms, independent from the domain of application. These latter criteria include the **form** of objects (*convex*, *concave*, *elongated*, *compact*, *square, round*, …), the **relative size of objects**, their **position** (*left, middle, right, top, bottom, center)* and **inter-object relations** (*proximity, connectivity, inclusion, …*).

3.2. Similarity calculation between two cases

One can consider two principles for the determination of similar cases, either maximize similarity (Caulier, Houriez, 1995) or minimize adaptation effort (Smyth, 1996). Owing to the absence of any automatic method for evaluating IP results, we have chosen the former. First the functions used for similarity calculation between a source case and a target case are described. Then comparison modes for each type of criterion are detailed. Finally, the management of missing values for a criterion is explained; in fact, as it is the case in ISAC (Bonzano *et al.*, 1997), all previously enumerated criteria need not be taken into account in any application.

Similarity functions

Our first group of criteria (i.e. criteria related to the task definition) is here to characterize the action performed by a task, and is thus closely dependent on the TMT model. Such criteria define a set of tasks that can solve one "type of problem". They are "compulsory" (each criterion of the target case must have a value) and are used to reduce the search space. A first similarity function Φ_t using the criteria related to the task definition will thus be applied to reduce the set of candidate target cases. This function is defined by formula (1) as the weighted average of the similarity results for each criterion: S is the source case, T is the target case , α_{Cr} is the importance coefficient associated to criterion Cr and $\varphi_{Cr}(S,T)$ is the similarity between S and T related to criterion Cr. The result value of any φ_{Cr} function is between 0 (if values of Cr between both cases are very different from each other) and 1 (when they are deemed identical). All α_{Cr} coefficients are also comprised between 0 and 1, in order to normalize the Φ_t function (return values between 0 and 1).

$$\Phi_t(S,T) = \frac{\left(\sum \alpha_{Cr} \times \varphi_{Cr}(S,T)\right)}{\sum \alpha_{Cr}} \qquad \forall Cr \in \{\text{criteria related to the task definition}\} \qquad (1)$$

The second group of criteria (i.e. criteria related to the context of images) characterizes the objects to be detected and depends on the current image. Such criteria are not meaningful for any application: for instance, contrast quality has no sense when processing a region map. This second group of criteria are "optional" ones (all criteria of the target case need not be filled in); they enable to select the nearest cases among the candidates obtained after applying function Φ_t. The second similarity function Φ_i is thus used to reduce the set of selected cases, in order to get a list of reasonable size. This function is defined by formula (2) as the weighted average of similarity results on each criterion; notations and properties are the same as in formula (1).

$$\Phi_i(S,T) = \frac{\left(\sum \alpha_{Cr} \times \varphi_{Cr}(S,T)\right)}{\sum \alpha_{Cr}} \qquad \forall Cr \in \{\text{criteria related to the image description}\} \qquad (2)$$

The definitions of functions φ_{Cr} that are in charge of similarity calculation for each category of criterion are given in the next paragraph. The use of similarity functions Φ_t and Φ_i in the selection/adaptation algorithm, as well as the adjustment of importance coefficients are explained in section 4.

Criterion comparison modes

It is clear that the list of criteria related to the context of images cannot be exhaustive: the criteria we put forward are coming from our study on IP literature and the development of our own applications. It should be completed in the course of further applications. Each criterion type is associated to a generic similarity function, in order to easily integrate new criteria. Here are the types of criteria that are presently available:

- strict numerical criterion: the value must be of integer or real type and comparison between two values returns 1 when values are strictly equal and 0 otherwise,
- strict symbolical criterion: the value is a symbol and comparison between two values returns 1 when values are strictly equal and 0 otherwise (e.g. presence of an image background),
- gradual numerical criterion: the value belongs to integer or real intervals and comparison between two values returns the difference between the two values divided by the interval length (e.g. relative size of objects),
- gradual symbolical criterion: the value belongs to an ordered set of symbols and comparison between two values returns the difference between the two values according to their order in the set, divided by the interval length (e.g. noise amount),
- multi-valued criteria: the value is defined as a non-ordered list of symbols and/or numbers and comparison between two values returns the ratio of the number of common elements in both lists to the length of the target case list (e.g. verbs used in the problem's definition).

A missing criterion value for a given case can be due to several causes (no meaning, usefulness, …) and can be taken into account in several ways (do not take into account, consider as a specific value, …). Our point of view on that issue differs whether one considers the source case or the target one:

- the absence of a value in a target case means that the value is considered as irrelevant for this case (either it is meaningless, or it has been judged as useless by user), that absence will have no consequence on similarity calculation ($\varphi_{Cr}(S,T)=0$ and $\alpha_{Cr}=0$),
- the absence of a value in a source case (while this value is present in the target one) means that one similarity condition is not respected; that absence should lower the result of similarity calculation ($\varphi_{Cr}(S,T)=0$ and $\alpha_{Cr}\neq 0$).

Both conditions are respected by the set of generic functions that compute similarity for each criterion type.

4. Recursive selection/adaptation algorithm

In the selection/adaptation process of most CBR systems, one can notice, on the one hand, the existence of a preliminary step in the selection process, aiming at reducing the case base (Bonzano *et al.*, 1997) (Netten, Vingerhoeds, 1996), and on the other hand, the fact that the selection/adaptation cycle must be applied iteratively, in particular in CBR planning (Prasad, 1995) (Smyth, 1996). Our approach (fig. 6) is also based on a selection/adaptation cycle, iteratively applied at various levels of the plan, but in addition, at each cycle loop, a reduction step of the case base has been included.

We consider several levels of abstraction for cases, but the notion of "level of abstraction" is different from the one presented in (Bergmann 96): our system does not reason with concrete cases and abstract ones (the latter modeling the world in a less detailed way) but only with concrete cases that can be related, either to complete IP plans (tasks at a high level abstraction) or to parts of them (tasks at a lower level of abstraction).

Contrary to most CBR planning systems (Prasad 95) (Smyth 96) (Veloso 96), which proceed by progressive refinement of an abstract plan, in our system, each tuning step produces a complete plan that can be tested and assessed. This choice of ours is due, on the one hand to our will to develop a conversational system and not a completely automatic problem-solver, and on the other hand, to issues raised by result assessment of IP applications (there exist no general function for comparing produced results with desired ones). One can thus very rapidly obtain a solution that will serve as a starting point for the IP expert and the only evaluation method that can generally be applied to any IP application is used: visual evaluation of output images by the expert.

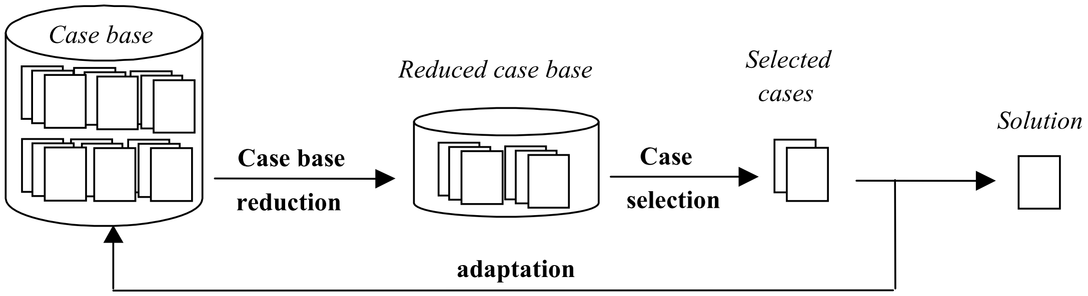


Fig. 6 : schema of our selection/adaptation process

The reduction of the case base can be achieved, either by using criteria corresponding to strict constraints, or by considering that two cases can only be compared when defined by the same set of criteria. The latter technique is not adapted to our domain. As a matter of fact, among the criteria related to the context of images, some of them bring nothing new about the target case, without disqualifying the source case. The reduction step can thus be achieved by means of function Φ_t using the "compulsory" criteria related to the task definition, while the selection step makes use of function Φ_i with the "optional" criteria related to the image description.

The objective of our CBR module is to provide some assistance to IP programmers when they are building applications, by helping them reuse solutions of previously-solved problems that are somewhat analogous to their current problem. The selection/adaptation process must thus take place in co-operation with the user, according to the following algorithm:

```
1. Ask user for values of criteria related to the task definition
2. Determine the set Σ of cases matching the desired criteria by
   means of Φt
3. Ask user for values of criteria related to the image description
4. While Σ is not of reasonable size do
          Modify the weight of criteria
          Reduce the set Σ by mean of Φi
5. Ask user to choose a case among the set Σ
6. Present the plan associated to the chosen case to user and
   propose him/her to modify the unsuitable sub-tasks, either by re-
   running the algorithm, or by building it from scratch, via the
   interactive creation module
```

Steps 1 and 3 correspond to the input of the description of the target case. Step 2 is the reduction step of the case base. The selection of candidate source cases is done in step 4; step 5 corresponds to the user's final choice. At last, step 6 consists in adapting the plan associated to the selected source case to the current problem.

Principles for selection and adaptation of cases used in our algorithm are detailed in the next two sections.

4.1. Selection of a source case

In the course of step 2 of the algorithm, the reduction of the search space consists in selecting source cases that solve the same type of problem as target case T. It corresponds to a selection of cases S such that $\Phi_t\ (S,T) > \alpha_t$ where α_t is a threshold fixed beforehand (as function Φ_t returns a value between 0 and 1, α_t is fixed to a default-value of 0.5). The weights of each criterion in function Φ_t are also fixed: the same importance is granted to all criteria. This step provides a first set of cases Σ.

So that the user can choose a case at step 5, the set of cases resulting from step 4 must be of reasonable size. If the set is too small, the user's choice will loose importance, and if it is too large, the user's choice will be difficult. The iterative nature of step 4 enables to get a set whose size can be shown to the user as a list: he/she can then examine each case in detail, before making the final choice, which well accounts for the intuitive aspect that characterizes the way IP experts work. The modification of set Σ at each iteration is done by means of a relaxation process, by modifying the weights of criteria and/or the selection threshold. To implement this kind of relaxation, when the user enters the values of criteria for the target case, he/she must indicate whether the criterion is considered as important or not. All importance criteria are initialized with 0.5. At each iteration, the system keeps the cases S from set Σ such that $\Phi_i\ (S,T) > \alpha_i$ where α_i is the selection threshold. If the size of the resulting set is too small or too large (by default between 2 and 5 cases), the coefficients of the most important criteria are raised by 0.1, whereas those of the least important ones are lowered by 0.1 for the next iteration. When it is no longer possible to modify coefficients (coefficients of the least important criteria have reached 0), if the set of source cases is still too small or too large, a second relaxation mode consisting in lowering threshold α_i is applied.

4.2. Interactive plan adaptation

Case adaptation by means of parts of other cases is particularly worthwhile in the domain of CBR planning. In our system, a case can be adapted at several levels and in several ways: locally or globally, either by means of the CBR module, or by means of the interactive creation module.

The plan solution to a case may only require minor local modifications. For instance, the parameters of an operator must be tuned, or an operator should be replaced by another one that better matches the current problem. This first type of modification can be taken into account by using the modification menu of the interactive creation module.

But a plan may also require broader modifications, i.e. necessitate the replacement of a whole sub-plan by another one. To achieve such modifications, step 5 of the selection/adaptation algorithm offers a means to adapt the solution of the current case by replacing the root task of any sub-plan of the current plan by another task. The substitution task can be obtained, either by re-running the algorithm in order to retrieve a similar case, or by building it from scratch, via the interactive creation module. In the example of figure 7, a plan is adapted along three successive steps:

- replacement of sub-plan A by sub-plan A’, which is obtained by re-running the selection algorithm,
- replacement of sub-plan B by sub-plan B’, which is built via the interactive creation module.
- transformation of tool C into tool C’, simply by changing the operator linked to tool C.

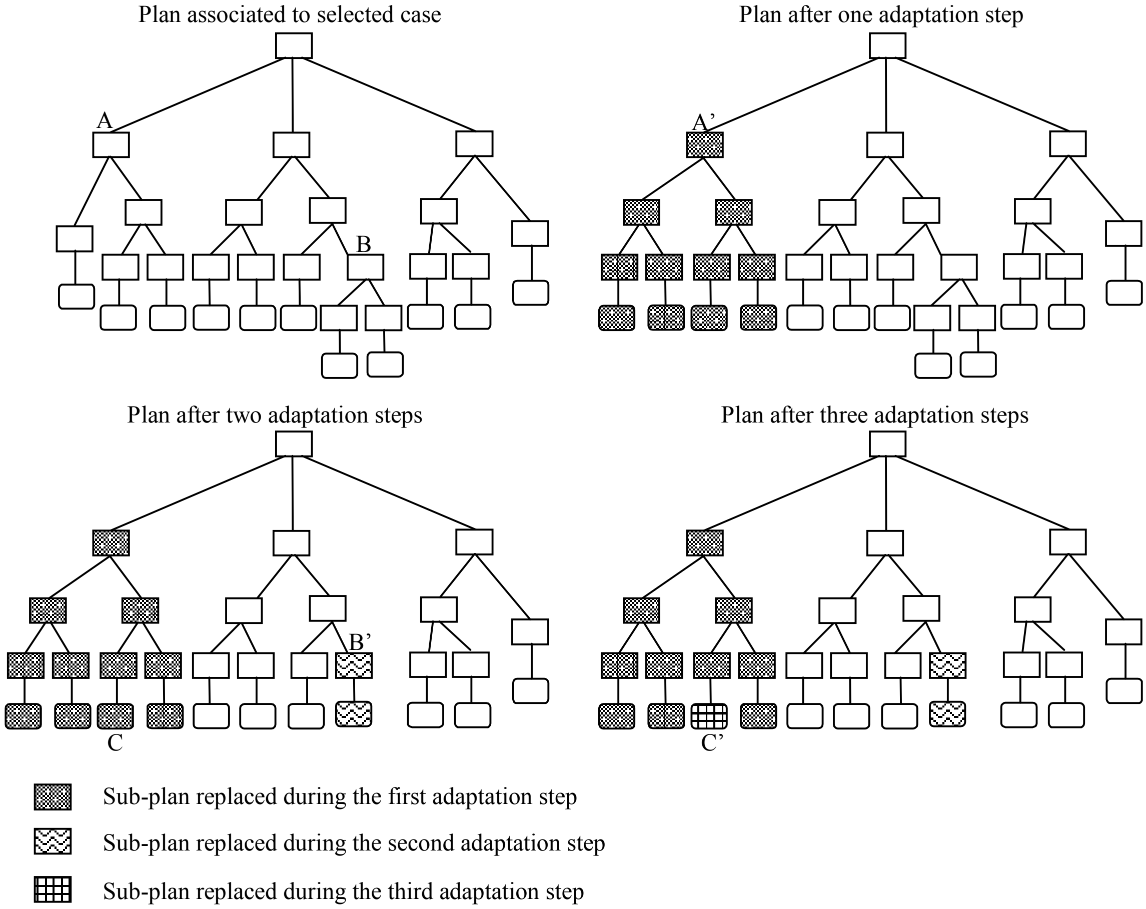


Fig. 7: adaptation of a solution plan along three steps

This example shows the interest in having a recursive algorithm: a plan can be adapted, whatever its level within the tree of tasks (A is a high-level task, B a low-level task, C an operator) and as long as necessary (A is replaced by A’, then A’ is adapted by replacing C by C’). Once a new plan is completed, one has to decide whether new cases associated to this plan should be added to the case library. This issue is discussed in the next section.

4.3. The memorization step

Memorizing a new case should only be considered if it brings new knowledge to the base. It implies that a case must respect two conditions in order to be integrated: the corresponding knowledge must be correct and it must be different enough from the knowledge of the cases that are already in the base.

Checking the first condition consists in verifying the consistency and efficiency of the produced plan. A plan is consistent when its execution is normal and it is efficient if it produces satisfactory results. Consistency can be checked by the correct progress of the plan execution, while its efficiency must be assessed by the user, who is the only judge of its relevancy. The integration of new cases will thus be achieved, on user's requirement, once the solution has been validated through a set of tests.

Several cases associated to one complete plan can be integrated into the base: in fact, if the complete plan represents the solution of a high-level problem, its various sub-plans represent solutions of problems at lower levels. When the integration of a case is required, a first step consists in determining the list of plans and sub-plans that are candidates to integration. This list corresponds to the plans that have been adapted, i.e. the ancestors of replaced sub-plans that are large enough (at least three levels of tasks). If the substitution plan has been built via the interactive module, it will also be inserted into the list. Figure 8 takes up again the plan adapted in figure 7; the determination of the candidates to integration is achieved by examining the three replaced sub-plans:

- sub-plan of root A': D is inserted into the list; A' is not inserted because it stems from a case of the base,
- sub-plan of root B': plans of roots E and F are inserted into the list; B' has been manually built but it is not inserted because it has only two levels.
- sub-plan of root C': plans of roots A' and G are inserted into the list, whereas H and C' are not because they have less than three levels.

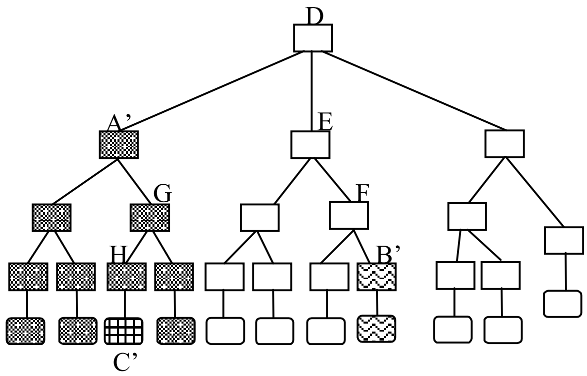


Fig. 8: determination of candidate cases to memorization

Then, for each plan in the list, the user has to provide values for the criteria of the corresponding case that have been modified. The system searches the case base for the most similar case to the new case and integrates the latter if similarity is lower than a given threshold (memorization threshold), i.e. the new case is different enough from all base cases, either according to the task criteria, or to the image criteria. The similarity here considered corresponds to the minimum between similarity on task criteria and similarity on image criteria : if that minimum is lower than the memorization threshold, it means that the case is considered as different enough from all base cases, according to *at least one* of the two types of criteria.

5. The CBR module at work: an example

In this section, a session showing how the CBR module can be used during the creation of a new application is described. The new problem consists here in extracting objects in an image from industrial origin (image (2), fig. 9). The user begins by defining his/her target case through an input window: **IP type** is *segmentation*, **problem** is defined as *extract* and *object*, task's **level** is *intentional*, **amount of noise** is *low,* **quality of contrast** is *medium*, there is an image **background**, objects are characterized by their *light gray level* **aspect**, *convex* **form**, **size relatively** *large* and *connectivity* **relation**. Background, aspect, form and relation are considered as important by the user.

The selection algorithm is then run in two steps. The first selection step only keeps cases whose similarity with regards to the target case according to the task criteria is higher than 0.5 (i.e. cases related to a type of problem similar to the type of problem solved by the target case). Then, during the second selection step, the weights of criteria and the selection threshold are tuned until the selection of cases according to the image criteria returns a set containing between two and five cases.

In our example, a list of four cases is returned, among which the user chooses the case that seems to be the best match for his/her problem. The plan solution to the selected case can be visualized, so as to study its strategy and it can also be executed.

The root task of the selected plan (fig. 9) is “isolate objects from background”; this plan has been built for a cytology application (images (1) and (3)), for the extraction of some categories of cells.

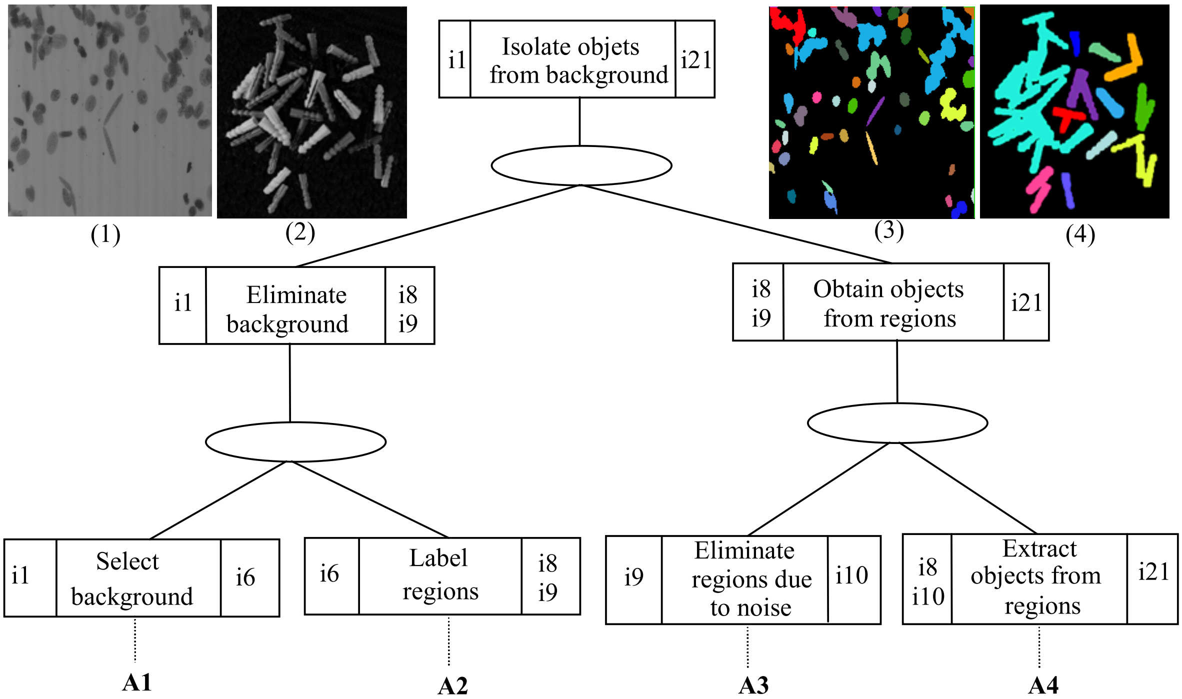


Fig. 9 : plan associated to the selected case with input and output images

The user can then start adapting the proposed plan to his/her new problem. The first modification deals with the “select background” task: in the initial plan, the problem was to isolate dark objects on a light background, whereas here, objects are light and background is dark. The first adaptation step simply consists in inverting the selection of objects (sub-plan F1, fig. 10) and is thus achieved via the interactive module. As results after execution are still unsatisfactory (imprecise localization of contours, objects not properly separated, image (4)), the user considers a second adaptation step by re-running the selection algorithm in order to find another sub-plan for the task “obtain objects from regions”. A new target case corresponding to this sub-problem is thus defined, the algorithm is re-run and the user finally chooses substitution sub-plan F2 (fig. 10). After replacement, the resulting plan (fig. 10) may further be improved by local modifications (e.g. replacement of an operator by another one).

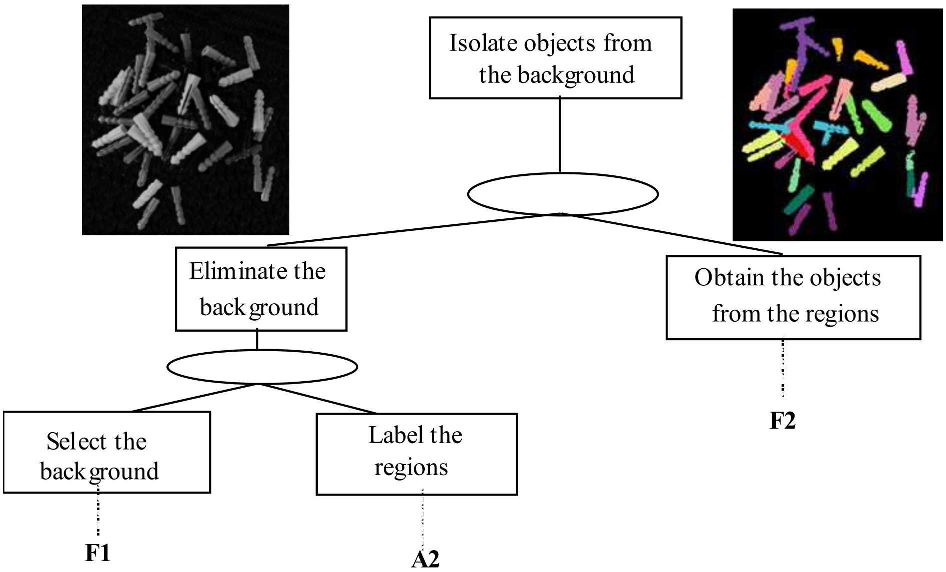


Fig. 10 : partial representation of plan after adaptation

Once all adaptations are completed, one has to define the new cases to be integrated into the base. The system produces the candidates to integration: they are the plans of roots “select background”, ”obtain objects from regions”, “isolate objects from background” and ”eliminate background”. For these four tasks, the user is required to define the corresponding cases: two of these four cases are integrated into the base.

The assistance provided by the CBR module for the tuning of this plan shows the aptness of our selection criteria and the efficiency of the selection/adaptation algorithm: the interactive and recursive nature of this algorithm enables to rapidly get a satisfactory solution. However, the number of further local adaptations that must be made reveals the scarcity of our present case base, which must now be enlarged by systematically integrating all plans and cases corresponding to the applications developed within our research team.

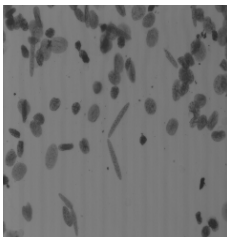
6. System validation

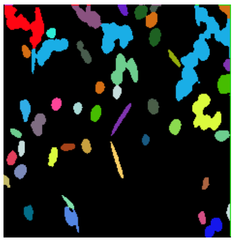
In this section, a few examples of applications developed thanks to the TMT system are presented. The corresponding IP plans served to test our system along three main axes: validation of the model and architecture, experimentation of the interface by an inexperienced user and search for similarities between applications from different fields.

6.1. Validation of model and architecture

Our first two IP plans have enabled to validate the TMT model and the system’s architecture by taking into account actual applications. They were developed in our research team by A. Elmoataz (Elmoataz *et al.*, 1996) and F. Angot (Angot, 1999) to process biomedical images of cytology and histology provided by the cancer-research centre F. Baclesse of Caen.

The former (plan A) works on cytological images. The goal is to detect epithelial cells. A precise description of the “objects” that can be found in such images was done in collaboration with a domain expert (e.g. image background is homogeneous and its grey level is lighter than the rest of the image). Plan A returns a region map where the various objects are labelled with different colours. From this region map, it is then possible to do further processing, such as eliminating objects that do no correspond to epithelial cells, according to size, shape or grey level criteria.

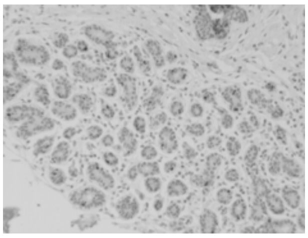


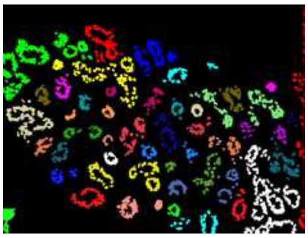


Plan A was the first plan that was integrated into our system. It has enabled to validate the TMT architecture by showing, on the one hand, that the decomposition of tasks into several abstraction levels gave a good representation of strategies and a good medium for dialogue between experts, and on the other hand, that the resulting plan was directly computational (i.e. could be immediately executed). It has also allowed to check the good functioning of the various execution modes of tools (simple execution, multiple execution, execution until a constraint is satisfied).

Besides, it has enabled to define more precisely some functionalities of the graphical interface concerning plan creation and execution. Its integration has also brought to the fore the need for syntactic verifications in the course of the modelling: as a matter of fact, problems due to the absence of syntactic consistency checking, only occur at the time of execution, and it is then difficult to determine what causes them.

The second plan (plan B) was created for an histological application. The goal was to detect significant groups of cells, such groups suggesting the presence of tumoural lobules. The plan returns a region map where each group of cells is labelled with a different colour.





As this second plan was relatively complex, it has enabled to complement and validate our functions for checking syntactic consistency of plans.

The input image of plan B is of the same type as the input image of plan A (same domain: biology, same acquisition device: microscope). The representation of both applications as TMT plans revealed the use of different IP techniques during the first step with corresponds to background "elimination" (use of contrast on boundaries in plan A, and use of inter-class variance in plan B). As a matter of fact, as both techniques can be applied to plan A, we have added a second method to the task "select background" of plan A (fig. 11), this additional method being a technique used in plan B. Such an operation shows, on the one hand, that it can be worthwhile to choose between several methods, in order to try various techniques for the same application, and on the other hand, that the representation of applications as TMT trees can be a medium for knowledge share and discussion between experts.

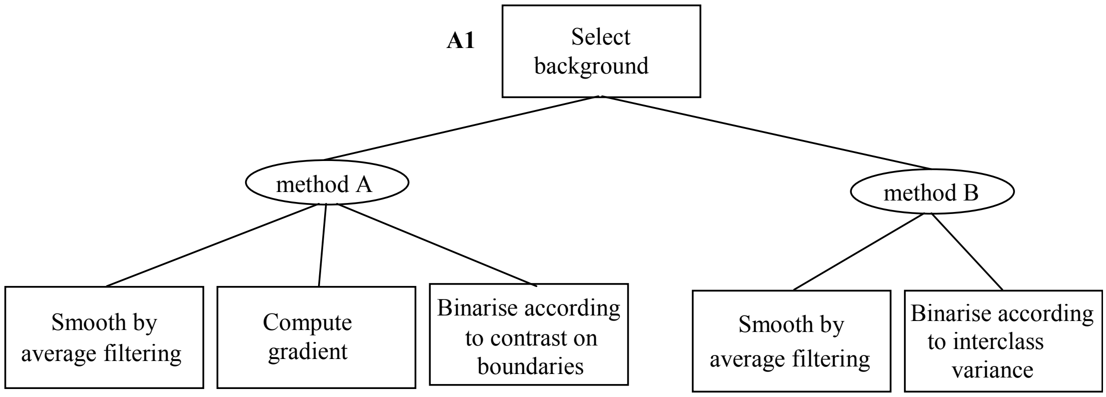


Fig.11 : addition of a second method to a sub-plan of plan A

6.2. Experimentation by a novice

A third plan (plan C) was integrated into the system by a novice who was both inexperienced in the system and in IP. The goal of this experimentation was to test if a new user could rapidly take the system in hand and also to validate the functionalities of the graphical interface.

Plan C was created to deal with images of human faces. The problem was set within the framework of "GDR-PRC ISIS", which is a French research group in Signal and Image Processing. It consists in localising the inner mouth corners. The novice developed three different versions (C1, C2, C3) to achieve this goal. The results presented in this paper correspond to plan C1, which performs the first step of the whole processing, i.e. extraction of the region corresponding to the teeth, and only works for open mouths. The area to be extracted is defined as a light area situated in the centre of the image.

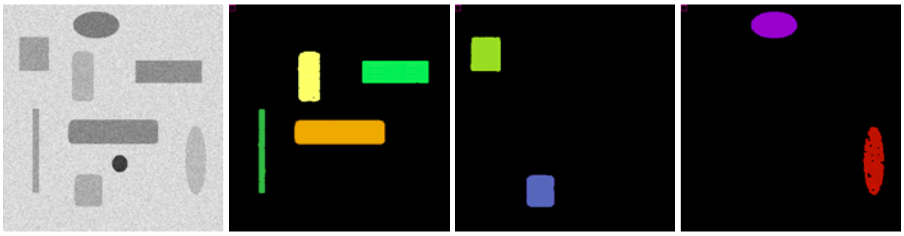
The integration of this plan by a novice has enabled to notice that system TMT was actually easy to take in hand: even if it is not always easy to give relevant names to high-level tasks, the modelling principles appeared clear and handy. This work has also raised new issues about the validation of the integrated knowledge (need to check keyboard errors as much as possible) and about man/machine communication (inadequate vocabulary or erroneous order of operations). This plan has also shown that the TMT system was not limited to operators of the library (although our library is rather exhaustive), as it includes a tool implemented as a C function and written on this occasion.

To become acquainted with our Pandore library of operators (Clouard *et al.*, 1997), the novice first implemented its application as a Shell script. Then it was modelled as a TMT plan so as bring to the fore the underlying strategy, which demonstrated the educational aspect of our approach. As a matter of fact, if the first plan (C1) was built in a bottom-up manner, the two others (C2 and C3) were developed in a top-down manner, by relying on the strategy discovered in the former.

6.3. Search for similarities between applications from different fields

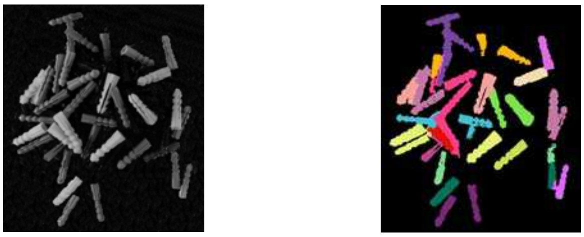
The two last plans we are now going to describe are working on images from two different origins (synthesis image and industrial image). Their integration into the TMT system have enabled to determine descriptive criteria that should be common to applications from different fields, and also to test the CBR module extensively.

The fourth plan (plan D) has been developed for testing the TMT system and, more precisely, some operators for selecting objects along shape criteria. It works on synthesis images, with the objective to sort objects according to their geometrical shape. The plan results are three distinct images, containing respectively rectangles, squares and ellipses.



This plan has enabled to test the interweaving of tools of various types: tools calling to Pandore operators and tools calling to Lisp functions. It shows that programming at the knowledge level facilitates the reuse of programming blocks written in different languages. Besides, this plan performs a new type of processing (detection of a specific shape) and works on a new image type. It has thus enabled to enrich the case base with cases related to pattern recognition tasks and to increase the set of indexing terms.

The fifth plan (plan E) was entirely built thanks to the CBR module. The goal is to isolate and separate the objects of an industrial image. The result is a region map, where each object is labelled with a different colour.



The first case selected by the CBR module was the case associated to plan A; it was then adapted by using parts of plan B. We could thus demonstrate the relevance of selection criteria and the efficiency of the selection/adaptation algorithm: thanks to its interactive and recursive nature, this algorithm enables to rapidly get a first solution. However, the number of further local adaptations that we had to do, revealed the scarcity of our present case base, which must be enriched with plans performing more varied treatments.

6.4. Experimentation assessment

The experiments we have just briefly described have been conducted in the very course of the system's design, in order to detect weaknesses as soon as possible, to determine their cause and correct them.

Future experiments must include the integration of a wide variety of applications in different domains, with a view to enrich the vocabulary used for case description and increase the CBR module role. In particular, we are presently complementing the Pandore library with image interpretation operators, which should enlarge our field of investigation. The system is currently in practical use but only its designers are playing the part of the IP experts. We must now consider testing the system in “real conditions”, i.e. have it validate by actual IP experts and not only by our team mates.

7. Conclusions

In this paper, a CBR module providing assistance to knowledge reuse has been described. It enables an IP expert to retrieve an existing plan that solves a problem similar to his/her current problem and adapt it to the new situation. He/she can thus reuse his/her own knowledge or knowledge previously modeled by other IP experts. Our recursive selection/adaptation algorithm alternates retrieval and adaptation steps, thus enabling to build a plan by combining parts of other plans. Criteria for selecting cases are based on a definition of IP tasks and a description of images.

Similar ideas can be found in HICAP (Munoz-Avila *et al.*, 1999), a general-purpose planning architecture that is applied to the planning of military evacuation operations. It is also a CBR system that can assist users during the construction of hierarchical plans of tasks. The system integrates a user-friendly task editor conducting an interactive conversation with the user. For tasks that can be decomposed in multiple ways (i.e. problem-specific tasks), a case is associated to each available decomposition method (whereas in our system, cases are associated to tasks and not to methods). So in HICAP, the user has to define a case in order to select each method used in the plan, which seems to be more constraining and time-consuming for the user.

In order to restrain the scope of the problem, tests have presently been limited to segmentation applications. Further work will consist in diversifying the content of our libraries (plans and cases) by integrating applications dealing with more varied treatments (from image restoration to image interpretation) and applied to images from various domains. This should also enable to enrich the vocabulary used for the description of cases, and thus complete our set of criteria, so as to get a more exhaustive lists of terms. We could accordingly build a more exhaustive IP ontology including a classification of terms and the definition of relations between them: such an ontology would support the dynamic indexing of the case base. Guidelines for achieving such an objective can be found in the works of Fensel (1998), who states that ontological engineering is one of the key issues for problem-solving methods reuse and proposes an “ontologist” module that acts as a kind of negotiator between user and system during method selection.

In addition, one should consider means to alleviate the user’s task in the course of the adaptation step. By using “simple” rules based on the comparison of some criterion values, the system could provide more assistance to user by indicating which parts of the plan need an adaptation. As the lack of theory in image processing does not allow us to establish such rules in advance, this objective could be achieved by performing an automatic search of adaptation rules. By keeping an history of all performed adaptations, one could automatically extract “simple” adaptation rules that are representative of our experts' work habits. The rule premises would correspond to comparisons between values of some criteria, and their conclusions would propose some modifications of the solution plan.

Acknowledgment

Our research on IP is carried out within the frame of the "Pôle Traitement et Analyse d'Image, TAI, de Basse-Normandie". This work is partially supported by a grant from the ministry of University Education and Research. Special thanks to our IP experts F. Angot and A. Elmoataz for their co-operation.

References

Angot F., 1999. Segmentation d'images 3D; application à la quantification d'images de tissus biologiques obtenues par microscopie confocale, PhD Thesis, Caen.

Bergmann R., Wilke W., 1996. On the Role of Abstraction in Case-Based Reasoning, *EWCBR'96*, Springer.

Bonzano A., Cunningham P., Smyth B., 1997. Using introspective learning to improve retrieval in CBR: A case study in air traffic control, *ICCBR'97*, Rhode Island, USA.

Caulier P., Houriez B., 1995. A Case-Based Reasoning Assistance System in Telecommunications Networks Management, *XPS'95*, Kaiserslautern, Germany.

Clément V., Thonnat M., 1993. A Knowledge-Based Approach to Integration of Image, Processing Procedures, *CVGIP: Image Understanding*, vol. 57, n° 2, Academic Press, pp. 164-184.

Clouard R., Elmoataz A., Angot F., 1997. PANDORE : une bibliothèque et un environnement de programmation d'opérateurs de traitement d'images, Rapport interne du GREYC, Caen, France. http://www.greyc.ismra.fr/~regis/Pandora/

Clouard R., Elmoataz A., Porquet C., Revenu M., 1999. Borg : A knowledge-based system for automatic generation of image processing programs, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, n° 2, pp. 128-144.

Elmoataz A., 1990. Mécanismes opératoires d'un segmenteur d'images non dédié: définition d'une base d'opérateurs et implémentation, Thèse de Doctorat, Caen.

Elmoataz A., Belhomme P., Herlin P., Schupp S., Revenu M., Bloyet D., 1996. Automated segmentation of cytological an histological images for the nuclear quantification: an adaptive approach based on mathematical morphology, Microscopy, Microanalysis, Microstructure, Vol. 7, pp. 331-337.

Fensel D., Benjamins R., 1998. Key Issues for Automated Problem-Solving Methods Reuse, *ECAI'98*, Brighton, UK, pp.63-67.

Ficet-Cauchard V., Porquet C., Revenu M., 1998. An Interactive Case-Based Reasoning System for the Development of Image Processing Applications, *EWCBR'98*, Dublin, Ireland, pp. 437-447.

Ficet-Cauchard V., 1999. *Réalisation d'un système d'aide à la conception d'applications de Traitement d'Images: une approche basée sur le Raisonnement à Partir de Cas*, Thèse de Doctorat, Caen.

IRIS Explorer User's Guide, 1993. Silicon Graphics, Inc., Mountain View, California, n°007-1371-020.

Munoz-Avila H., Aha D., Breslow L., Nau D., 1999. HICAP: An Interactive Case-Based Planning Architecture and its Application to Noncombatant Evacuation Operations. IAAI-99.

Netten B.D., VingerhoedsR.A., 1997. Structural Adaptation by Case Combination in EADOCS, G*WCBR'96*, Bad Honnef, Germany.

Prasad B., 1995. Planning With Case-Based Structures, *AAAI Fall Symposium*, MIT Campus, Cambridge, Massachusetts.

Rasure J. & Kubica S., 1994. The Khoros Application Development Environment, Experimental Environments for Computer Vision and Image Processing, editor H.I. Christensen and J.L. Crowley, *Word Scientific*, Singapore, n° 1-32.

Russ, J.C., 1995. *The Image Processing Handbook*, second edition, CRC Press.

Smyth B., 1996. *Case-Based Design*, Doctoral Thesis of the Trinity College, Dublin, Ireland.

Veloso M. ,Munoz-Avila H., Bergmann R., 1996. Cased-based planning: selected methods and systems, *AI Communications*, vol. 9, n. 3.