IMAGE ANALYSIS OPERATORS AS CONCEPT CONSTRUCTORS

Philippe DEJEAN , Patrice DALLE

Institut de Recherche en Informatique de TOULOUSE
Pôle 2IL/TCI
Université Paul SABATIER
118, route de Narbonne
31062 TOULOUSE Cedex
-FRANCE-
dejean@irit.irit.fr
dalle@irit.irit.fr

ABSTRACT

In this paper we present a new approach for the modeling of image processing operators. This model accounts for the information transformations made by the operators, and describes precisely the produced entities intervening as visual cues. Therefore, an operator sequence is considered as a concept constructor supplying a description of the concepts of specifics fields.
First, we describe the elementary components of this model and give an example of concept construction. Then, we show how this model is used to solve objectives of application areas.

I. INTRODUCTION

The automation or the help in configuration of image processing operator sequences assumes that one is able to determinate, at each step of the process, the operation to perform *(data selection, choice of the operators, determination of the parameters)*. We propose a new approach for the modeling of image processing operators. This model accounts for the transformation of informations made by the operator, and describes precisely the produced entities intervening as visual cues *(patterns, spatial relations, etc...)* during the interpretation process. These visual cues will become concepts *(roads, rivers, fields, or cells, cytoplasm, nucleus, etc...)* according to the application area. Therefore, an operator sequence is considered as a concept constructor supplying a description of these concepts. This approach differs from systems based on scripts [8], on functional decomposition of operators [2] or on transfer graph of input-output image processing entities [9]. In this way, our model can be used by a system providing help in configuration to describe image processing objectives and to plan or to rectify corresponding sequences of operators.

II. MODELING OF IMAGE ANALYSIS OPERATORS

An image analysis operator extracts a particular information from an image. The description has to explicit the detection and representation steps which make it possible to distinguish between the information to reach and the rest [4]. Therefore an operator can be considered from two points of view :

On the one hand, it achieves splitting and merging operations which specialize or generalize corresponding concepts, and representation operations which correspond to the nomination of these concepts.

On the other hand, it builds or manipulates entities *(pixels, connected components, ...)* applying mechanisms of structuration and distructuration.

- *Specialization* increases the information associated with a concept specifying a new predicate *(characteristic, relation, domain)* verified by the instances of this concept.
- *Generalization* consists in disjunctively combining concepts.
- *Nomination* of concepts is materialized by the representation of produced data. In image processing area, the space of representation is often insufficient *(binary image, byte-coded image, ...)*. Consequently, concepts, considered as secondary ones for the realized operation, are mixed in the same representation. For example, in binary images, the "zero values" often contains everything that has been filtered during the previous steps of the process. The model we propose accounts for this reality in image processing.
- *Structuration* consists in merging, in a higher level entity, the ones having some common characteristics and respecting a given spatial relation *(for example : merging the connected pix-*

els whose intensity value is greater than a given threshold).

- *Distructuration* allows to consider, in order to treat it, information relative to the lower level elements of the manipulated entity *(for example : the value "distance to the border of the form" of the connected component pixels).*

III. DESCRIPTION OF OPERATORS

Our model does not aim to describe image processing operators functionalities, but rather the transformations of information the operator induces on input data. It integrates five elementary operations which occur during a transformation :

- *Indexation* determines the substructures of the initial data. These data elements corresponding to the same index will be subjected to the same treatment.
- *Measurement* produces an output value directly computed from the input data. This evaluation is made by using a given characteristic, linked to the operator.
- *Partition* consists in separating entities of the same class -index- according to a given predicate [12] linked to the operator. This constitutes the operation of specialization of information attached to the index by the operator predicate.
- *Coding* consists in attributing an output value to the different entities, coming from the partition.
- *Merging* brings together, in a same representation field of the produced data, the elements issued from previous operations and which have received the same coding value.

The figures below summarize such a process of transformation of information (fig.1) and show the decomposition of the operator performing the dilation

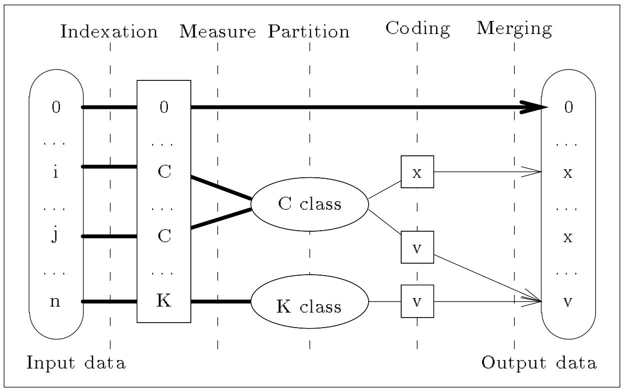
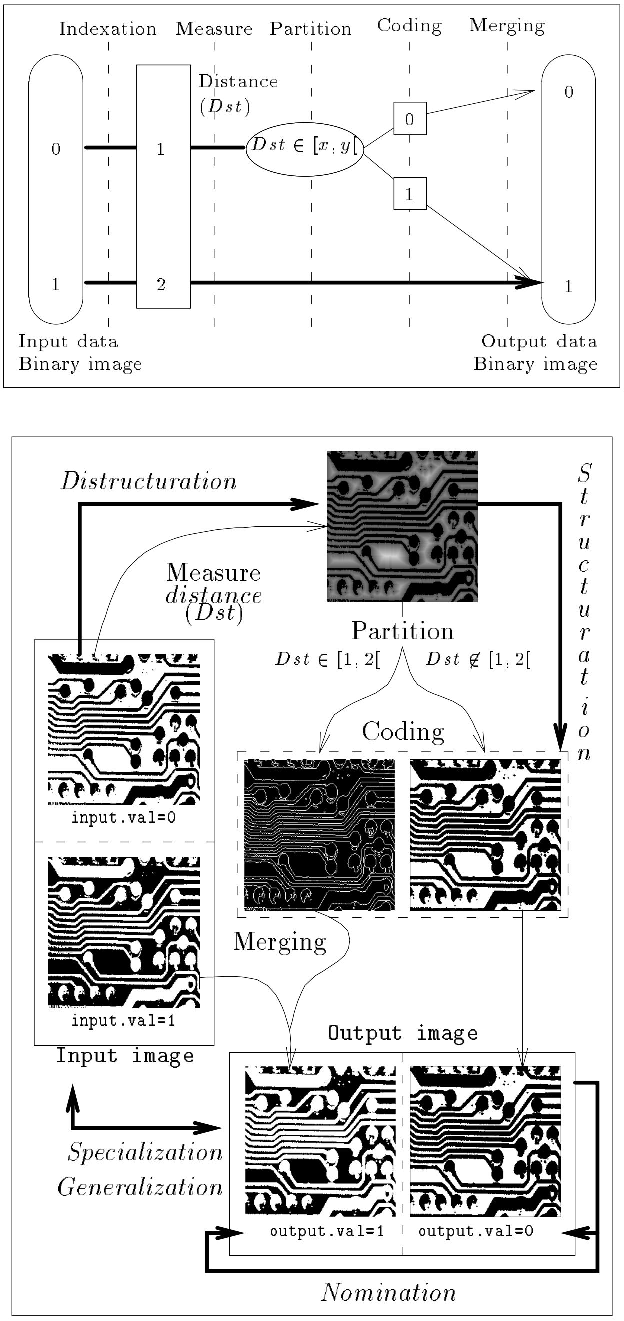


Figure 1: Transformations of information

of connected components by a structuring element -*rad.=1*- (KHOROS [7] operator fig.2).

Dilatation operator	`vmdilate`
Main transformation :	Partition / Merging
Partitioned substructure :	"0"
Partition characteristic :	Distance to the connected component edge
Separating values :	Parameter
Transmitted substructure :	"1"
Produced substructure :	"0" and "1"
Merging in substructure :	"1"

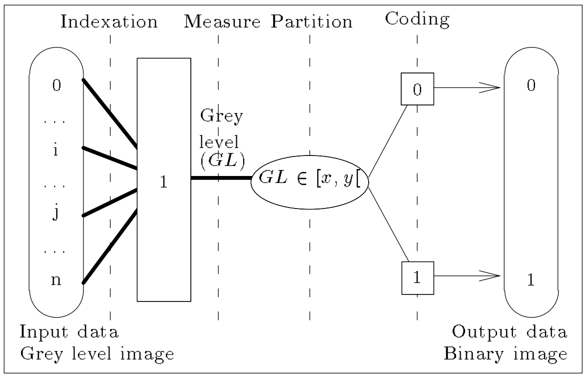


Textual description (Michalski's notation [11]) `inval` : input.val — `outval` : output.val
`outval=1` : `inval=1` $\vee$ (`inval=0` & $[Dst = 1]$)
`outval=0` : `inval=0` & $[Dst \neq 1]$

Figure 2: Simple example : the `vmdilate` operator

In figure 3 we show the decomposition of the threshold operator applied on a grey-level image :

Threshold operator :	`vthresh`
Main transformation :	Partition
Partitioned substructure :	Full image
Partition characteristic :	Pixels representation value
Separating values :	Operator parameters
Produced substructure :	"0" and "1"



Textual description (Michalski's notation [11]) `inrep` : pixels representation value (GL, ...)
`outval=1` : $[\texttt{inrep} = X..(Y-1)]$
`outval=0` : $[\texttt{inrep} \neq X..(Y-1)]$

Figure 3: `vthresh` KHOROS operator

IV. CONCEPTS CONSTRUCTION

The textual descriptions of the informations transformations performed by the image processing operators is used to develop a low-level concept descriptive language using an annotated predicate calculus [11] extension [6] and also all its generalization, specialization and simplification rules.

Therefore, we can associate a textual description to an image processing operator job and then to an image processing operators sequence (fig.4).

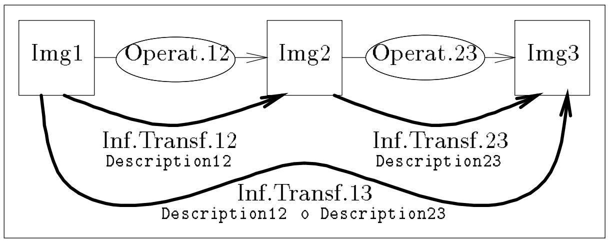


Figure 4: Propagation

Thus, we can express and handle some generic concepts as :

`filled objects` :
$\quad$ $\texttt{objects} \vee In(\sim \texttt{objects}, \texttt{objects})$
`objects thickness` :
$\quad$ $Max(Dst(\texttt{objects}))$
or more complex `thicker objects` :
$\quad$ $Dst(\texttt{objects})\&[MaxDst = Max(Dst(\texttt{objects}))]$

These descriptions are used to express more specific concepts and to develop a specific language for an application area. We obtain an "executable vocabulary" ; each concept is associated to an image operator sequence.

In the example below we propose some simple descriptions of printed circuits concepts (fig.5) :

`Objects` :
$\quad$ $[GL = 78..255]$
`SlotsWithStrips` :
$\quad$ $\texttt{FilledObjects}\ \&\ [MaxDst = 5..127]$
`Unconnected slots` :
$\quad$ $NoInter(\texttt{SlotsWithStrips}, \texttt{ImageBorders})$
`StripsUnconnectedWithSlots` :
$\quad$ $\texttt{FilledObjects}\ \&\ [MaxDst = 2..4]$
`CrossingStrips` :
$\quad$ $Inter(Inter(\texttt{StripsUnconnectedWithSlots}, \texttt{LeftBorder}), \texttt{RightBorder})$
or completely developed :

$$Inter(\ Inter(\ ([GL = 78..255] \vee In(\ [GL = 0..77], [GL = 78..255])\ )\ \&\ [MaxDst = 2..4], (fenetre\ 0\ 0\ 0\ 255), (fenetre\ 255\ 0\ 255\ 255))$$

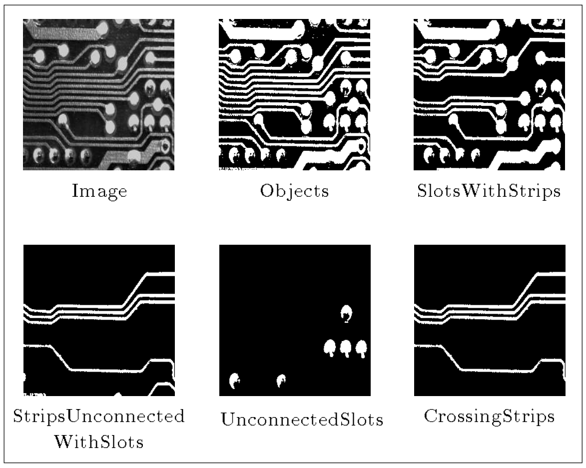


Figure 5: Concepts Descriptions

V. EXPLOITATION

Applied to a sequence of image processing operators, this model allows us to obtain, at each step of

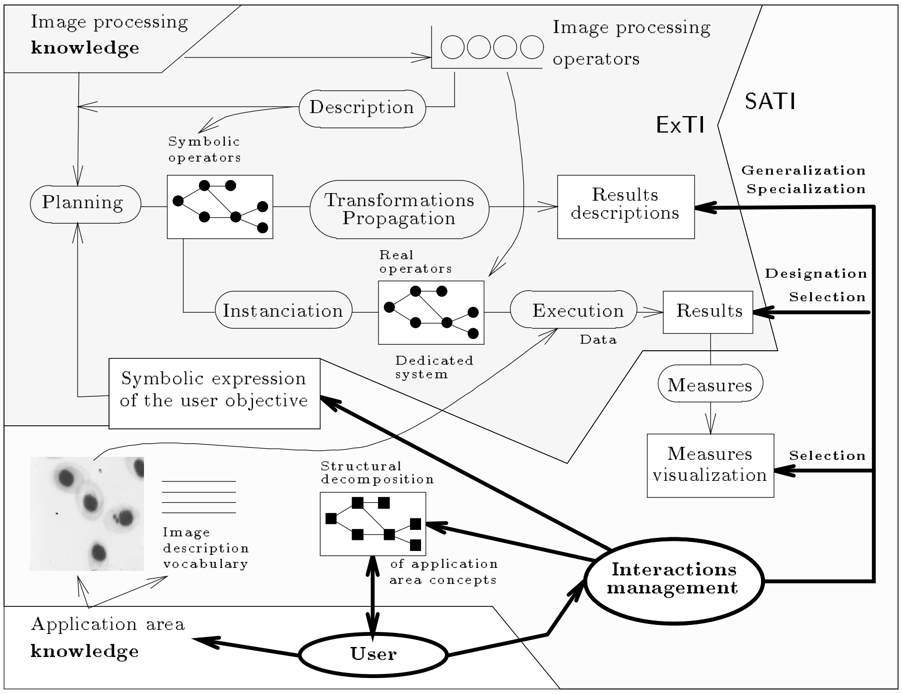


Figure 6: Cooperative process for program conception

this sequence, the description of the entities contained in the data. It correctly accounts for the transformations performed by image processing.
Inversely, if we describe an image processing concept with this formalism, the symbolic expression can be considered as an objective to reach. The model of operators, because of its possibility to propagate transformations, allows the system to plan the sequence of operators producing the described data.

The system called ExTI [6] uses such a model. It is built on a library of real operators, a base of symbolic operators containing their description, a descriptive image data model [5] and a process which can plan and combine expressions describing the entities. These expressions can be built interactively by using an human-machine interface called SATI [1] which implements the PAC hierarchy model [3]. In this double system ExTI-SATI (fig.6) we put together an "end-user module" which holds application area knowledge, and a "planning module" which holds image processing knowledge. The interface between these modules is implemented according to a symbolic expression language of the concepts.

On the one hand, the operators handled by the planner are thought as visual cues constructors. Our symbolic operator model allows to represent and to manage output types or data, and to describe information transformations the operators do.

On the other hand, a user describes, validates and combines these concepts by interactions (generalization, specialization) on their textual description or (providing examples or counterexamples on processed images) on the instances of the concepts produced by plan execution. In this way he can specify concepts characteristics, in several spaces of the visual representation. The dialog manager translates each user interaction into a symbolic description of connected concepts. So the user task is to define a good formulation of his goal, not to solve it. Conversion from mental image of the problem to solve to its expression in visual cues terms is made easier by using a visual language.

VI. CONCLUSION

The model of operators proposed allows us to develop an environment helping the configuration of interpretation systems. Actually we can build, above this internal formalism, an interactive module providing an objective description language and rectification tools. Completed by a planning module as the ExTI one, it can constitute the image processing expert package of an interpretation system, such as LLVE in SIGMA [10] but with focusing on information instead of type entities. Finally, this formalism can be used to build object models from examples. From a description of the processing of a specific scene image, it allows to interactively generalize the description linked to the objects contained in such a scene. In conclusion, this model has two interests : descriptive and constructive. It precisely accounts for the effect of a processing sequence and it makes it possible to build sequences producing a given objective.

VII. REFERENCES

[1] O. Capdevielle and P. Dalle. Image Processing chain construction by interactive goal specification. In *ICIP, Austin Texas (U.S.A.)*,1994.

[2] V. Clement and M. Thonnat. A knowledge-based approach to integration of image processsing procedures. *CVGIP: Image Understanding*, 57(2):166–183, march 1993.

[3] J. Coutaz. PAC : an implementation model for dialog design", *Interact'87*, pages 431–436. H.J. Bullinger, B.Shackel ed., september 1987.

[4] P. Dalle and J.-M. Inglebert. Modèle de description d'opérateurs de traitement d'images. In *8th RFIA, Lyon-Villeurbanne (France)*, pages 257–266, 1991. (in french).

[5] Ph. Dejean and P. Dalle. Modèle symbolique de la donnée de traitement d'images. In *10th RFIA, Rennes (France)*, pages 746–755, 1996. (in french).

[6] J.-M. Inglebert. Planification automatique de chaînes de traitement d'images. PhD thesis, Université P. Sabatier de Toulouse (France), 1992.

[7] Khoros Reference Manual, University of New Mexico, 1992.

[8] C.-E. Liedtke and A. Blömer. Architecture of the knowledge based configuration system for image analysis CONNY. In *11th ICPR, The Hague (The Netherlands)*, (1):375–378, 1992.

[9] T. Matsuyama. Expert systems for image processing : knowledge-based composition of image analysis processes", In *CVGIP*, 48(1):22–49, october 1989.

[10] T. Matsuyama and V. Hwang. SIGMA: a knowledge-based aerial image understanding system. Plenum Press, 1990.

[11] R.S. Michalski. A theory and methodology of inductive learning. In *Artificial Intelligence*,20(2):111–161, 1983.

[12] R.S. Michalski, J.G. Carbonell and T.M. Mitchell. A theory and methodology of inductive learning. In *Machine learning : an artificial intelligent approach*, (1):83–134, Berlin : Springer-Verlag", 1984.