Timed Automata Model to Improve the Classification of a Sequence of Images

Christine Largouët[1] and **Marie-Odile Cordier**[2]

Abstract. The aim of this paper is to propose the use of a dynamic plot model to improve landcover classification on a sequence of images. This new approach consists in representing the plot as a dynamic system and in modeling its evolution (knowledge about crop cycles) using the timed automata formalism. In order to refine results obtained by a traditional classifier, observations given by a preliminary classification of images are matched with expected states provided by an automaton simulation. The paper presents the modeling captured by the timed automata formalism and the general method, which is based on prediction and postdiction mechanisms, that have been adopted to improve the classification of a sequence of images. Finally, the interest of the method is demonstrated through experimental results.

1 Introduction

The classification and recognition problem has been intensively researched in the recent past, incorporating new achievements of statistical, neural networks and Artificial Intelligence techniques. The reason for classifying remotely-sensed or aerial images is to assign to each object a label. In this study, images are related to an agricultural site and describe agronomic plots we want to label with landcover type (corn, barley, forest, water, etc.). The result is called a thematic map. The traditional statistical classification tools, that we have experimented, provided unsatisfactory results. Inconsistencies are often due to the same causes: the confusion between landcover types of similar spectral signatures and the existence of isolated pixels of one class in an object identified as belonging to another class. Thus, results provided by these tools usually need to be completed with manual processing so as to have as good results as those obtained by human photo-interpreters. With regard to these problems, two kinds of approaches are possible: to develop more sophisticated statistical methods in one hand or to develop systems using domain knowledge in the other hand. This paper focuses on the second approach. Matsuyama [11] developed an expert system for the classification of images where image processing techniques used to extract scene information are distinguished from knowledge representation used to identify objects. In scene interpretation, paradigms such as frames, semantic nets and production systems [14, 12] are commonly used to describe structural knowledge such as spatial-relations properties of objects. These systems have been tested on specific domains including the recognition of airports [12], roads and buildings. In the domain of landcover classification, knowledge-based image analysis is also applied. These systems incorporate image context and spectral characteristics in the ruled-based classification. Even if these methods provide better results than those obtained using traditional classifiers, a significant part of the image still contains confusions between classes. Some authors [13] suggest that the conjunction of multi-seasonal remotely-sensed images is useful to discriminate between different categories of vegetation.

The aim of this paper is to present a new approach to improve landcover classification. We suggest working on a sequence of images and exploiting the dynamics of the entity we want to classify, i.e the plot of land. The idea consists in modeling the evolution of the plot and comparing the expected state of the plot obtained by a model simulation with the observations extracted from the images, in order to improve the classification of images. This approach, which involves revising the observations by taking into account the expected state of the system, has some similarities with approaches proposed for monitoring dynamic systems, such as Kalman filtering [5], and with the belief change/update domain [3]. We use a preliminary classification to define all plausible classes for each plot [3]. The knowledge about crop cycles, evolution of main crop states and rotation practices are described in the timed automata formalism [2]. The refinement of the classification is based on prediction and postdiction mechanisms where the model is used to give the expected state of the plot at each date referred to here as time.

The paper is organized as follows. Section 2 presents the problem and the general mechanism adopted to improve the classification of a sequence of images. Section 3 defines the modeling formalism chosen to represent the evolution of the system and which is based on timed automata. Section 4 explains how this model is exploited on a sequence of images by prediction and postdiction steps. Experimental validation and results are presented in section 5, followed by a conclusion and further lines of research in section 6.

2 Refinement problem

In this paper we consider the landcover classification of a sequence of images $I_1, \cdots, I_n$ taken at time $t_1, \cdots, t_n$. Images may have been acquired by different sensors (SPOT, LANDSAT, aerial, etc.) as they represent the same landscape area. A preliminary per-plot classification is performed on each image and associates to each agricultural plot of the area an observation, i.e the set of all plausible classes. Thus since we have a sequence of images, we get, for each plot, a sequence of observations coming from the preliminary classification. The problem is to improve such a classification using knowledge describing the possible evolution of the plots, and this will be referred to as refinement of the classification. The general algorithm is:

[1] ENSAR, 65 rue St Brieuc 35042 Rennes cedex & IRISA, campus de Beaulieu 35042 Rennes cedex, France, email: clargoue@irisa.fr
[2] INRA, 65 rue St Brieuc 35042 Rennes cedex & IRISA, campus de Beaulieu 35042 Rennes cedex, France, email: cordier@irisa.fr

[3] The classification step itself will not be discussed in this paper.

for each image I_i
  for each plot P
    preliminary classification $\longrightarrow O_i$
for each plot P
  refinement of the classification
    $(\mathcal{A}, [O_1, \cdots, O_n]) \longrightarrow [K_1, \cdots, K_n]$

Let $\mathcal{C}$ be the set of classes. We denote O_i the observation about a plot P at time t_i. $O_i \subseteq \mathcal{C}$ is the set of all plausible classes describing the plot and resulting from the classification of the image I_i. The refinement problem takes as input the pair $(\mathcal{A}, [O_1, \cdots, O_n])$ where $\mathcal{A}$ is the plot evolution model. The objective of the refinement is to provide, for each plot, a sequence $[K_1, \cdots, K_n]$ where $K_1 \subseteq \mathcal{C}, \cdots, K_n \subseteq \mathcal{C}$ and where the "quality" of $[K_1, \cdots, K_n]$ is better than $[O_1, \cdots, O_n]$. Two criteria enable us to judge the quality of K_i:

- the cardinal of K_i: the fewer plausible classes in K_i, the better it is. The best case is when K_i is restricted to one class.
- "validity" correctness: the real landcover type should belong to K_i. If a ground truth is available for this plot, K_i and the class given by the ground truth are compared in order to assess the accuracy of the classification.

The refinement is realized in two steps. The evolution model of the plot $\mathcal{A}$ is used in simulation to determine the set of expected data at time t_i, denoted E_i with $E_i \subseteq \mathcal{C}$. A matching is applied between the observation and the expected data. The matching process is illustrated in Figure 1 and is defined as follows. Let O_i be the observation at t_i. Let $\otimes$ be the matching operator such that $K_i = O_i \otimes E_i$.

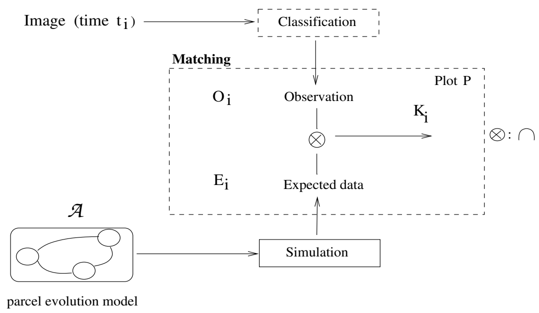

Figure 1. Refinement of the classification: simulation and matching

In the following, since sets of classes are considered, the matching operator is the intersection. The matching may be extended to probabilistic or ranking data where the operator represents any fusion or combination rule.

Expected data are provided by simulation according to a prediction or a postdiction mechanism.

Prediction. We define the prediction mechanism as $< \mathcal{A}, K_{i-1}, t_i >$ where $\mathcal{A}$ is the model of the system, K_{i-1} is a state of the system at time t_{i-1} with $t_{i-1} < t_i$. Given the model and a state of the system at time t_{i-1}, the prediction mechanism consists in determining the state of the system for the current time t_i.

Postdiction. We define the postdiction mechanism as $< \mathcal{A}, K_{i+1}, t_i >$ where $\mathcal{A}$ is the model of the system, K_{i+1} is a state of the system at time t_{i+1} with $t_i < t_{i+1}$. Given the model and a state at time t_{i+1}, the postdiction mechanism consists in determining the state of the system for the current time t_i.

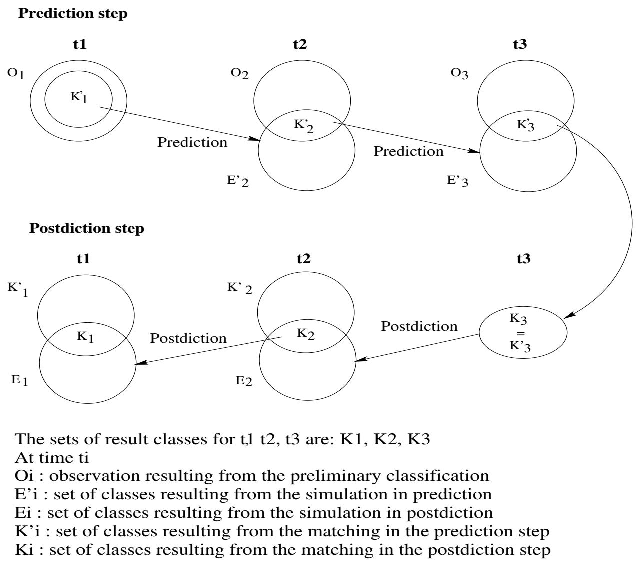

Figure 2. Prediction and postdiction on 3 dates

In order to refine the sequence of images, the method relies on both prediction and postdiction steps. We denote E'_i (resp. E_i) the set resulting from the prevision step (resp. postdiction step) and K'_i (resp. K_i) the set resulting from the matching in the prediction step (resp. postdiction step). K'_1 is the set of classes resulting from O_1, from which invalid classes at t_1 have been discarded, and is used to initialize the plot evolution model. The refinement algorithm, based on prediction and postdiction steps, is as follows.

Refinement of a plot:
1- Prediction
Initialization: $K'_1 \subseteq O_1$
For all t_i, with $2 \leq i \leq n$
  Prediction: E'_i = prediction $< \mathcal{A}, K'_{i-1}, t_i >$
  Matching: $K'_i = O_i \otimes E'_i$
2- Postdiction
Initialization: $K_n = K'_n$
For all t_i, with $n - 1 \geq i \geq 1$
  Postdiction: E_i = postdiction $< \mathcal{A}, K_{i+1}, t_i >$
  Matching: $K_i = K'_i \otimes E_i$

In an ideal case, there is no more confusion between classes and only one good landcover type is proposed in each K_i. Figure 2 shows the refinement of a classification on a sequence of 3 images.

3 Timed automata modeling

Let us come now to the model and to the formalism we have chosen to encode it. Due to meteorological factors and to availability of labor and machinery, the main feature of the system is its non-determinism, which means that the system could be in multiple states at the same time. Consequently, the transition-based model needs to deal with uncertainty on temporal constraints over transitions. Note also that the formalism adopted should be able to model crop cycles, thus permitting us to define one generic model for several years of study. Timed automata [2] bring together all the properties required to model the evolution of the plot.

3.1 Formalism

Timed automata extend the automata formalism by adding clocks. The vertices of the graph are called *locations*. Clocks are real-valued variables increasing uniformly with time. Several independent clocks may be defined for the same timed automaton. They define timing constraints associated to locations or transitions. In a timed automaton, transitions are instantaneous and allow the resetting of clocks. A timing constraint related to a location is called its *invariant*. It is possible to stay in a location as long as its invariant is true. A timing constraint related to a transition means that the transition is enabled only when the value of the clocks satisfies the constraint.

Clock constraints. A clock constraint is the conjunction of atomic constraints which compare the clock value with a non-negative rational. We consider $\mathcal{X}$ the set of clock variables and $\Phi(\mathcal{X})$ the set of clock constraints φ defined by the following grammar:

$$\varphi ::= x \leq c \mid c \leq x \mid x < c \mid c < x \mid \varphi \wedge \varphi$$

where $x \in \mathcal{X}$ and c is a constant in $\mathbb{Q}$. A clock valuation v for a set $\mathcal{X}$ assigns a real value to each clock such that for all $x \in \mathcal{X}$, $v(x) \in \mathbb{R}$.

Definition. A timed automaton $\mathcal{A}$ is a tuple $< \mathcal{S}, \mathcal{X}, \mathcal{L}, \mathcal{E}, \mathcal{I} >$ where:

- $\mathcal{S}$ is a finite set of locations and $s_o \in S$ is the initial location.
- $\mathcal{X}$ is a finite set of clocks.
- $\mathcal{L}$ is a finite set of labels.
- $\mathcal{E}$ is a finite set of edges, each edge e is a tuple $(s, l, \varphi, \delta, s')$ such that e connects the location $s \in \mathcal{S}$ to the location $s' \in \mathcal{S}$ on symbol l. The enabling condition required for all clocks is captured in φ and $\delta \subseteq \mathcal{X}$ gives the set of clocks to be reset when the edge is triggered.
- $\mathcal{I} : S \rightarrow \Phi(\mathcal{X})$ maps each location s with a clock constraint called an invariant.

Let us consider the preliminary example shown in Figure 3. This

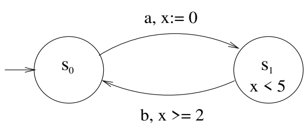

Figure 3. Preliminary example of a timed automaton

timed automaton is described by two locations s_0 and s_1 and a single clock x. The system is in the initial location s_0 where it can stay during an arbitrary time since no invariant is associated with it. The system moves from the location s_0 to the location s_1 on the transition labeled by *a* and the clock x is set to 0. The location s_1 is possible as long as x is less than 5. The *b*-labeled transition can then be triggered during the uncertain interval [2,5[.

Semantics. The semantics of a timed automaton $\mathcal{A}$ is defined by a transition system $< \mathcal{Q}, \rightarrow, (s_o, v_o) >$ where $\mathcal{Q}$ is the set of states, $\rightarrow$ the set of transitions and (s_o, v_o) the initial state. A state of $\mathcal{A}$, denoted (s, v), is given by a location s and a valuation v such that v satisfies the invariant $\mathcal{I}(s)$. For all clocks $x \in \mathcal{X}$, we have $v_o(x) = 0$. At any state, $\mathcal{A}$ can evolve through one of the outgoing edges or remain in the location while time passes. Consequently, two kinds of transitions are distinguished.
Discrete transition. Let $e = (s, l, \varphi, \delta, s') \in \mathcal{E}$. The state (s, v) has a discrete transition to the state (s', v') if v satisfies φ. This discrete transition is denoted $(s, v) \xrightarrow{l} (s', v')$ where $v' = v[\delta := 0]$. $v[\delta := 0]$ gives the new clock valuation v' such that all clocks $x \in \delta$ are set to 0, the others keeping the value they had in the location v. For instance, in the preliminary automaton a discrete transition would be $(s_0, x = 1) \xrightarrow{a} (s_1, x = 0)$.
Timed transition. Let $t \in \mathbb{R}$. The state (s, v) has a timed transition to the state (s, v_t) denoted $(s, v) \xrightarrow{t} (s, v_t)$ if, for all $t' \leq t$, $v_{t'}$ satisfies the invariant $\mathcal{I}(s)$ where v_t (resp. $v_{t'}$) is the valuation obtained by adding all the clocks with t (resp. t'). A timed transition in the example is $(s_1, x = 1) \xrightarrow{1} (s_1, x = 2)$.

3.2 Plot evolution model

We consider landcover types as the locations of a timed automaton. Clock constraints are derived from crop calendars, generally expressed using dates. As the formalism requires rational values for clock constraints, dates are transformed into numerical values in $[0, 365]$. Because most crop cycles begin at the end of the summer, the constant that appears in a clock constraint is the number of days since September 1. Both types of transitions make sense to describe the evolution of our system. Timed transitions are useful to express the time elapsed in a crop state, while discrete transitions express successions between landcover types. Cycles are defined by resetting the clocks.
Figure 4 illustrates a simple version of a plot evolution modeled using a timed automaton. The automaton is composed of several sub-automata, each of which refers to the evolution of one crop (corn, wheat, rape plant, etc.) or to invariant landcover types (water, forest, etc.). From the final location *end crop* of each sub-automaton, all possible transitions towards the successive crops begin. Two clocks are necessary. Clock x refers to the number of days in one cycle, from September 1 to August 31, such that $x \in [0, 365]$. Clock y represents the number of days since September 1 of the first cycle and enables us to count the years. The value of x corresponds to the value of y modulo 365 days. The data set used for this study was acquired from interviews with agronomists about the Rennes agricultural site located in Brittany (France). Our automaton contains 40 states and 62 transitions.

4 Specifying prediction and postdiction mechanisms with reachability properties

Given the evolution model of a plot defined using a timed automaton, this section explains how to specify the refinement problem in terms of the reachability properties of a timed system. Analyzing the behavior of a timed automaton means verifying whether the timing properties it is supposed to verify are satisfied. There are several verification problems and we have chosen to focus on reachability analysis. Reachability analysis consists in determining whether or not, given two states, an execution starting at one state reaches the other state.
TCTL [1] is a temporal logic usually used to specify properties where time is introduced explicitly into the syntax. The TCTL formulae are defined by the following grammar:

$$f ::= p \mid x \in I \mid \neg f \mid f_1 \vee f_2 \mid \exists \Diamond_I f \mid \forall \Diamond_I f$$

where $p \in \mathcal{P}$ is a basic predicate, $x \in \mathcal{X}$ is a clock and I is a time interval. Intuitively, $\exists \Diamond_I f$ means that there is an execution leading to a state satisfying f at time $t \in I$ and $\forall \Diamond_I f$ means that every run has a state where f holds at time $t \in I$.

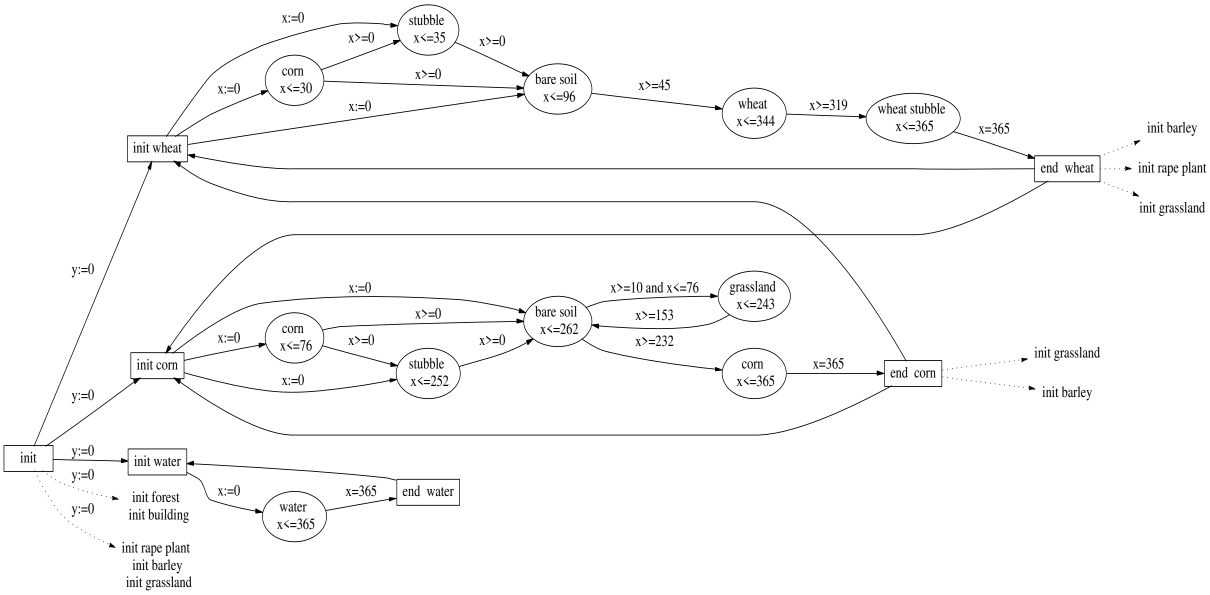

Figure 4. Part of the plot evolution modeling expressed as a timed automaton

Let t_i and d_i, be the date and the year of O_i. The refinement algorithm based on prediction and postdiction is now expressed using the TCTL formulae.

1- Prediction

Initialization:

$K'_1 = \{s \in O_1 \mid init \Rightarrow \exists\Diamond_{=t_1} s\}$

For all t_i, with $2 \leq i \leq n$,

$K'_i = \{s \in O_i \mid init \Rightarrow \exists\Diamond_{=t_i} s$ and $\exists s' \in K'_{i-1}$ such that: $(s' \wedge x = t_{i-1} \wedge y = d_{i-1}) \Rightarrow \exists\Diamond\,(s \wedge x = t_i \wedge y = d_i)\}$

2- Postdiction

Initialization: $K_n = K'_n$

For all t_i with $n-1 \geq i \geq 1$

$K_i = \{s \in K'_i \mid \exists s' \in K'_{i+1}$ such that: $(s \wedge x = t_i \wedge y = d_i) \Rightarrow \exists\Diamond\,(s' \wedge x = t_{i+1} \wedge y = d_{i+1})\}$

In the prediction step, the first reachability property verifies that each state s proposed by the observation is consistent, at time t_i, according to its invariant. The second one checks the successibility condition. Several algorithmic methods for solving this verification problem have been researched and are supported by tools [6, 10, 15]. To implement the verification of these properties, we have used Kronos [15].

5 Experimental results

Our approach has been applied to the images of an environmental project in which the landcover map is a fundamental feature regarding water pollution. A sequence of five images (one aerial, one LANDSAT and three SPOT) is at our disposal. They were taken at different dates (between April 18 1997 and August 7 1998) on a watershed near Rennes in France. Each image contains 2124 plots we aim to classify into ten classes known *a priori*.

5.1 Preliminary classification

The preliminary classification is a per-plot classification performed using a traditional classifier called Arkemie. The idea of preliminary statistical classification was proposed in knowledge-based imagery interpretation though it was applied to the pixel [7, 14]. The role of this initial classification is to apply low-level processes to the image. We have chosen a supervised classification which uses samples from the ground truth to identify the classes [9]. For each plot this classification provides a probability distribution over classes. In this approach, we do not consider the ranking value of classes but use a minimum threshold with a value of 0.1 in order to discard less representative classes from the set of plausible classes.

5.2 Classification assessment

It should be remembered that to assess the quality of a result set describing the plot, two criteria must be taken into account: the number of classes and the matching with the ground truth if it is available for the plot. With regard to the first criterion, we have defined three qualitative kinds of plot:

- *Clear* plot: the plot is identified by only one class.
- *Ambiguous* plot: the plot is identified by several classes.
- *Non-labeled* plot: the plot is not identified (there are classification difficulties in labeling too small plots).

With regard to the second one, we rely on a ground truth which is available over about 110 plots, for each image, meaning that for these plots we already know their real landcover type, thanks to human observation. We propose a classification performance criterion called *identification rate* performed on each image. Let $\mathcal{G}_i$ be, for the image I_i, the set of plots with a ground truth available. $|\mathcal{G}_i|$ is the cardinal of the set $\mathcal{G}_i$. Let $\mathcal{W}_i$ be, for the image I_i, the set of plots, elements of $\mathcal{G}_i$, which proposes the class given by the ground truth. $|\mathcal{W}_i|$ is the cardinal of the set $\mathcal{W}_i$. The identification rate τ_i for the image I_i is defined by $\tau_i = 100.\frac{|\mathcal{W}_i|}{|\mathcal{G}_i|}$ and is used to verify the reliability of the results. In our experiment we have a dual objective: to increase the number of clear plots between the results of the preliminary classification and the results provided by the refinement, and to obtain a reasonable identification rate.

5.3 Analysis of the results

Table 1 describes the results provided by the preliminary classification. The last three columns show the number and the percentage of

I_i	τ_i	clear plots		ambiguous plots		non-labeled plots	
I_1	90.91%	1788	84.2%	330	15.5%	6	0.3%
I_2	89.29%	1697	79.9%	386	18.2%	41	1.9%
I_3	75.68%	796	37.5%	1306	61.5%	22	1%
I_4	64.49%	958	45.1%	1161	54.7%	5	0.2%
I_5	63.55%	541	25.5%	1583	74.5%	0	0%

Table 1. Preliminary classification results

the three types of plots. Table 2 lists the results after the refinement of the classification has been applied to the 5 images of the series.

I_i	τ_i	clear plots		ambiguous plots		non-labeled plots	
I_1	90.91%	2030	95.6%	88	4.1%	6	0.3%
I_2	86.90%	1908	89.8%	175	8.2%	41	1.9%
I_3	81.08%	1948	91.8%	154	7.2%	22	1%
I_4	69.16%	1813	85.4%	306	14.4%	5	0.2%
I_5	75.70%	1594	75%	530	25%	0	0%

Table 2. Refinement of the classification results

The result clearly shows the progression of the number of clear plots on all images (up to 54.3% for image I_3). The identification rate remains reasonable. The interpretation of the results is detailed in [9]. We have conducted another experiment to show the respective contributions of prediction and postdiction. It leads us to make the length and the first image of the sequence vary. In Table 3, lines represent

	I_1	I_2	I_3	I_4	I_5
I_1	-	**242**	**0**	**0**	**0**
I_2	209	-	**2**	**0**	**0**
I_3	0	174	-	**806**	**172**
I_4	0	39	244	-	**572**
I_5	0	28	93	932	-

Table 3. Contribution of prediction and postdiction

the number of new clear plots resulting from the refinement for the images I_1 to I_5 and columns $I_1 - I_5$ represent the contribution of the images during the prediction and postdiction steps (shown in ordinary and bold type, resp.). If we take image I_3, image I_2 contributes 174 clear plots in the prediction step, and images I_4 and I_5 contribute 806 and 172 clear plots respectively in the postdiction step.

6 Conclusion

In this paper, we have proposed refinement of the classification of a remotely-sensed sequence of images. One distinctive feature of the method is that it considers the plot as a dynamic system and its evolution is modeled in order to improve the classification. Observations are derived from the images using a preliminary classification. The model simulation provides the expected state of the system, which is matched with the observation in order to propose a better set of classes for each plot at each date. The general framework is based on two mechanisms: prediction and postdiction. Timed automata appear to be the most convenient formalism to model the plot evolution by taking into account time constraints, uncertainty on transitions, and cycles. The experiments we have carried out show that the number of clear plots increases on all images and that the identification rate is globally better. We intend to validate the method by experimenting it on another area of study (the Vittel site, France).

Our current work focuses on the integration of probabilities in the modeling. A new matching operator has been defined to take into account the probabilistic distributions describing the observation and the expected data resulting from the simulation of a probabilistic timed automaton. Finally a decision rule has to be applied to choose the preferred class for ambiguous plots remaining at the end of the refinement process. First results obtained from this probabilistic approach do not seem to improve on the refinement of the classification proposed in this paper. A parallel may be drawn between this approach (simulation and matching) and the Generalized Update (GU) proposed by Boutilier [3] based on a two-step process (prediction and revision). This approach is also related to work developed to cope with uncertainty and imprecision of sensor data and based on the fusion of information from different sources [4, 8, 7]. These methods successfully discriminate between classes with similar spectral characteristics but they require a large amount of information and several images from various sensors at the same date. Our approach is sensor-independent and relies only on high-level knowledge that is easy to collect.

REFERENCES

[1] R. Alur, C. Courcoubetis, and D.L. Dill, 'Model-checking in dense real time', *Information and Computation*, **1**, 2–34, (1993).

[2] R. Alur and D.L. Dill, 'A theory of timed automata', *Theoretical Computer Science*, **126**(183), 235, (1994).

[3] C. Boutilier, 'A Unified Model of Qualitative Belief Change: a Dynamical Systems Perspective', *Artificial Intelligence*, **1-2**, 281–316, (1998).

[4] V. Clément, G. Giraudon, S. Houzelle, and F. Sandakly, 'Interpretation of Remotely Sensed Images in a Context of Multisensor Fusion Using a Multispecialist Architecture', *IEEE Transaction on Geosciences and Remote Sensing*, **31**(4), 779–791, (July 1993).

[5] C. Cossart and C. Tessier, 'Filtering vs Revision and Update: Let us Debate!', in *ECSQARU'99*, pp. 116–127, (1999).

[6] C. Daws, A. Olivero, S. Tripakis, and S. Yovine, 'The tool KRONOS', in *Hybrid Systems III*, ed., LNCS 1066, pp. 208–219. Springer-Verlag, (1996).

[7] J. Desachy, L. Roux, and E.H. Zahzah, 'Numeric and symbolic data fusion: A soft computing approach to remote sensing images analysis', *Pattern Recognition Letters*, **17**, 1361–1378, (1996).

[8] S. Le Hégarat-Mascle, I. Bloch, and D. Vidal-Madjar, 'Application of Dempster-Shafer Evidence Theory to Unsupervised Classification in Multisource Remote Sensing', *IEEE Transactions on Geoscience and Remote Sensing*, **35**(4), 1018–1031, (July 1997).

[9] C. Largouët and M-O. Cordier, 'Improving the Landcover Classification using Domain Knowledge', in *Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS 2000)*, Amsterdam, (July 2000).

[10] K. Larsen, P. Petterson, and W. Yi, 'UPPAAL in a nutshell', *Springer International Journal of Software Tools for Technology Transfer*, **1**, (1997).

[11] T. Matsuyama, 'Knowledge-based Aerial Image Understanding Systems and Expert Systems for Image Processing', *IEEE Transaction Geoscience of Remote Sensing*, **GE-25**, 305–316, (1987).

[12] D. M. McKeown, W.A. Harvey, and J. McDermott, 'Ruled-based interpretation of aerial imagery', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **7**(5), 570–585, (1985).

[13] H. Shimoda, S.Hong, K. Fukue, and T. Sakata, 'A Multi-temporal Classification Method using Dempster-Shafer Model', in *International Geoscience and Remote Sensing symposium IGARSS11*, Espoo Finland, (1991).

[14] J.K. Wu, D.S. Cheng, and W.T. Wang, 'Model based remotely-sensed imagery interpretation', *International Journal of remote-sensing*, **9**(8), 1347–1356, (1988).

[15] S. Yovine, 'Kronos : A verification tool for real-time systems', *International Journal of Software Tools for Technology Transfer*, **1**, (1997).