Adding probabilities to timed automata to improve landcover classification

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

[1] IRISA Campus de Beaulieu 35042 Rennes Cedex, FRANCE
{clargoue,cordier}@irisa.fr
[2] ENSAR 65, Route de St Brieuc 35042 Rennes Cedex, FRANCE

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 the results obtained by a traditional classifier, the observations given by a preliminary classification of images are combined with the expected states provided by simulation. The paper presents the general method, based on prediction and postdiction mechanisms, that have been adopted to improve the classification of a sequence of images. In a first experiment, plots are modelled using a timed automata formalism. The analysis of the results clearly demonstrates the interest of the method. A second experiment, in which plots are modelled using probabilistic timed automata, is described in a last section. The changes in the method are presented and the results are compared with the previous ones.

1 Introduction

The classification 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 [10] 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 [13, 11] 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 [11], 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 [12] 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 [4], and with the belief change/update domain [3]. We use a preliminary classification to define all plausible classes for each plot [1]. 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 image date.

The paper is organized as follows. Section 2 presents the refinement problem and the general mechanism adopted to improve the classification of a sequence of images. Section 3 defines the timed automata formalism chosen to model the evolution of the system. Section 4 explains how prediction and postdiction mechanisms are expressed as checking reachability properties which allows to take profit of the efficiency of model-checking techniques. Section 5 presents and discusses the experimental results. Section 6 is devoted to the second experimentation in which probabilities were added to the model. The results are commented. Lastly, section 7 concludes and gives further lines of research.

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 all the 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 the following:

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

Preclassification:
for each image I_i
    for each plot P
        classification $\longrightarrow O_i$;

Classification refinement:
for each plot P
    Plot refinement task
        $(\mathcal{A}, [O_1, \cdots, O_n]) \longrightarrow [K_1, \cdots, K_n]$
    Decision task
        $[K_1, \cdots, K_n] \longrightarrow [C_1, \cdots, C_n]$

Let $\mathcal{C}$ be the set of classes. We denote O_i the observation about a plot P at time t_i. O_i represents all plausible classes describing the plot and resulting from the classification of the image I_i.

The *plot refinement task* is applied on each plot considered independently along time. The idea is to combine the two sources of information, the observation sequence and the model, in order to get a "better" (more realistic) result than the observation sequence alone. The refinement task takes as input the pair $(\mathcal{A}, [O_1, \cdots, O_n])$ where $\mathcal{A}$ is the plot evolution model and $[O_1, \cdots, O_n]$ the sequence of observations resulting from the preliminary classification. It provides a sequence $[K_1, \cdots, K_n]$ where K_i follows the same representation as O_i. The objective of the refinement is to provide, for each plot, a sequence $[K_1, \cdots, K_n]$ where the "quality" of $[K_1, \cdots, K_n]$ is better than $[O_1, \cdots, O_n]$.

The *decision task* consists in selecting, from the sequence $[K_1, \cdots, K_n]$ provided by the refinement task, the final result. It depends on the application. In our case, the result will constitute one of the entry of a GIS, which means that exactly one class for each date and each plot must be proposed as a result of the algorithm. It is clear that the final quality of the refinement problem strongly depends on the quality of this choice. We come back on the quality criteria in a next paragraph.

Plot refinement task.
The refinement is realized in two steps. The evolution model of the plot $\mathcal{A}$ is used in simulation to determine the expected data at time t_i, denoted E_i. 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 $K_i = O_i \otimes E_i$. The matching operator used depends on the uncertainty formalism used to represent the data.

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

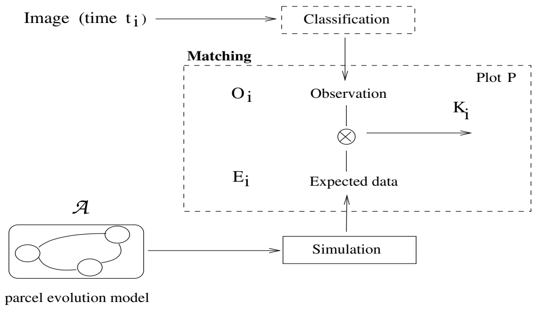


Fig. 1. Refinement of the classification: simulation and matching

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.

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 state resulting from the prevision step (resp. postdiction step) and K'_i (resp. K_i) the state resulting from the matching in the prediction step (resp. postdiction step). K'_1 represents the 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: $K_i = K'_i \otimes E_i$

Figure 2 shows the refinement of a classification on a sequence of 3 images, where the states of the system are represented as sets. In this case, the matching operator is the intersection.

Classification assessment and decision task.
The refinement task provides as a result the sequence $[K_1, \cdots, K_n]$ where K_i represents the plausible classes for a plot at date i. The decision task selects one class C_i among

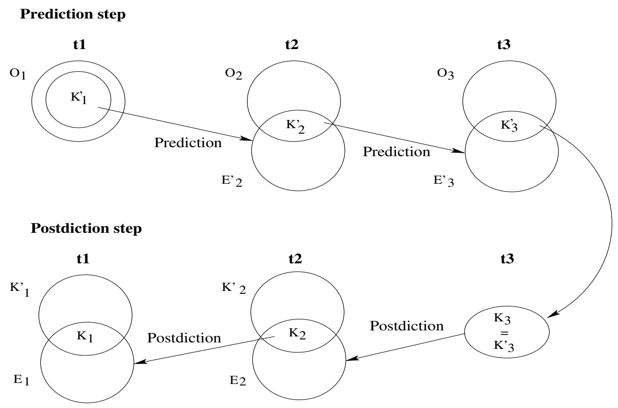


Fig. 2. Refinement task : prediction and postdiction on 3 dates

these plausible classes. The method relies on the hypothesis that the higher quality the sequence $[K_1, \cdots, K_n]$ is, the better selected class C_i will be.

The classification quality is given by the quality of the final sequence $[C_1, \cdots, C_n]$ and depends also on the quality of the intermediary sequence $[K_1, \cdots, K_n]$. It can be estimated in different ways. The more intuitive one is to compare each selected class C_i with the real class, called, in the remote-sensing domain, the ground truth. One of the criteria is the identification rate, i.e the percentage of correctly classified plots. The ideal case is clearly when there is only one plausible class after the refinement task and when this class corresponds to the one observed by the ground truth. The weakness of this criterion is that the ground truth is often difficult to get; it is thus available for only few plots at each date and it is even not always reliable. Another quality criterion, which does not require any ground truth, estimates the benefit of the refinement task by the number of classes of K_i with respect to that of O_i. For instance, in our experimentation, the non-ambiguity rate is the number of “clear” plots, i.e plots such that there is only one class left after the refinement process ($|K_i| = 1$).

The decision task depends on the uncertainty formalism used to represent the plausible classes for each plot. In this paper, we successively present two approaches. In the first one (section 3-5), we use a set-theoretic approach for the refinement task and the only uncertainty degree comes from the preclassification task. In the second one (section 6), we use a probabilitic model and the refinement task assigns to each plot a probability distribution over the set classes $\mathcal{C}$.

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.
- $\mathcal{I} : S \to \Phi(\mathcal{X})$ maps each location s with a clock constraint called an invariant.

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

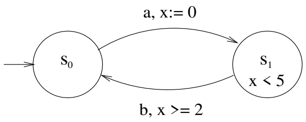


Fig. 3. Preliminary example of a timed automaton

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 $(s_1, x = 1) \xrightarrow{1} (s_1, x = 2)$.

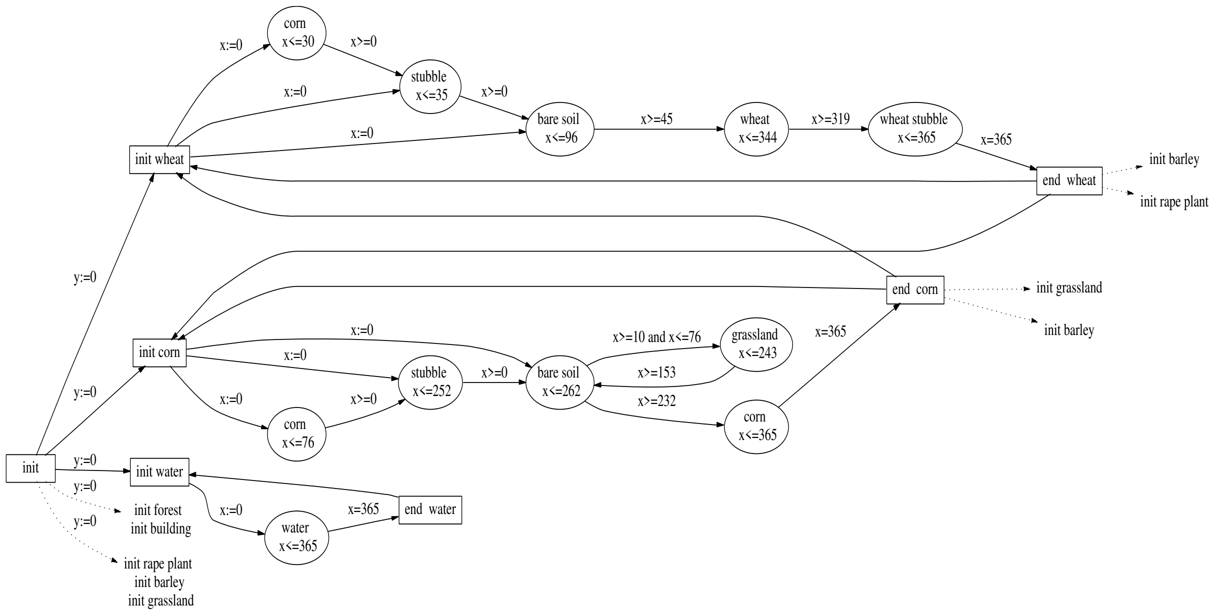


Fig. 4. Part of the plot evolution modeling expressed as a timed automaton

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 41 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. Model-checking techniques verify properties on states of an automaton whereas our problem deals with landcover classes. Reachability analysis is one of the symbolic verification methods usually dedicated to the validation of timed systems [7]. It is defined as follows: *given two locations of a timed automaton, the problem consists in verifying if there is a path leading from one location to the other location*. It is then quite adequate to implement the prediction and postdiction mechanisms. Reachability properties are expressed in the TCTL temporal logic [2].

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 $t \in I$.

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. A correspondence between classes and states is easily established by an *observation model*. It is used before the refinement task which is now applied on states of the automaton and after the refinement task in a reverse way so that the decision rule should be applied on the probability distributions over classes.

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 [5, 9, 14]. To implement the verification of these properties, we have used the tool Kronos [14].

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 [6, 13]. 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 [8]. For each plot this classification provides a probability distribution over classes. In this approach, we 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

In order to evaluate the method, it is necessary to judge the quality of the classification before and after the refinement. We define two criteria to assess the accuracy of the classification:

- non-ambiguity rate (γ_i): it describes the percentage of clear plots (i.e. the plots identified with only one candidate class) in an image. It is computed before having applied the decision rule.
- identification rate (τ_i): it describes the correctness of the classification with respect to the ground truth. It is the percentage of correctly classified plots (only 110 plots have a ground truth). It is calculated after the decision task.

With regard to the first criterion, we have defined three qualitative kinds of plot: (i) a *clear plot* is a plot identified by only one class, (ii) an *ambiguous plot* is a plot identified by several classes, (iii) a *non-labeled plot* plot is not identified (there are classification difficulties in labeling too small plots). We have a dual objective for each classified image: to increase the non-ambiguity rate and to obtain a reasonable identification rate.

Decision task. In this application and since the classification constitutes one of the entry of a GIS, we have to affect, for each plot and at each date, only one class. During the refinement of the classification, each time a class is discarded from the set of plausible classes, its probability is equitably added on the others having a value greater than 0. Then, after the refinement, the class chosen to compute the identification rate is the one, none discarded by the method, having the maximum probability.

Table 1 explains how the refinement and decision tasks are processed on a specific plot. The first line represents for the five images $I_1 \cdots I_5$ the classes given by the preliminary classification. The values into brackets are the probabilities associated to each class. The refinement task begins by the prediction step : prediction and matching at each date. Thus, given a grassland at t_1, the plot evolution model is used in simulation and propose for t_2 the set $\{$grassland, corn$\}$. This set is compared to the one given by the preliminary classification $\{$grassland, wheat$\}$ so that the grassland class is only retained for t_2 by the prediction step. This principle is applied for all the dates of the sequence. The postdiction begins when all the images have been treated by the prediction step. Then, the model is used in postdiction to determine the possible classes at t_4 from the resulting set given by the prediction at t_5: $\{$wheat, grassland$\}$. The model simulation proposes the classes $\{$wheat, grassland, corn$\}$ and the matching operator retains only the landcovers $\{$wheat, grassland$\}$. The postdiction mechanism is applied from t_5 to t_1. After the refinement task, the plot is still ambiguous for the last three dates. The decision task is then applied in order to select only one class for each date. Two scenarios are however consistent : *grassland, grassland, wheat, wheat, wheat* and *grassland, grassland, grassland, grassland, grassland*. In this approach, the decision task selects the class having the maximum of probability in the observation thus, wheat is chosen for I_1, I_2 and I_3.

	I_1	I_2	I_3	I_4	I_5
observation O_i	grassland (1)	grassland (0,88) wheat (0,12)	wheat (0,37) grassland (0,20) forest (0,43)	wheat (0,64) grassland (0,36)	wheat (0,70) grassland (0,11) forest (0,19)
prediction E'_i	$\rightarrow$ -	$\rightarrow$ grassland corn	$\rightarrow$ wheat grassland bare soil	$\rightarrow$ wheat grassland corn	$\rightarrow$ wheat grassland
matching K'_i	grassland	grassland	wheat grassland	wheat grassland	wheat grassland
postdiction E_i	$\leftarrow$ grassland	$\leftarrow$ wheat stubble grassland corn	$\leftarrow$ wheat grassland	$\leftarrow$ wheat grassland corn	$\leftarrow$ -
matching K_i	grassland	grassland	wheat grassland	wheat grassland	wheat grassland
choice	**grassland**	**grassland**	**wheat**	**wheat**	**wheat**

Table 1. Illustration of the refinement process

5.3 Analysis of the results

Table 2 describes the results provided by the preliminary classification. The results show, for each image I_i, the identification rate τ_i, the number and the percentage of clear plots, ambiguous plots and non labeled plots.

I_i	clear plots (number, γ_i)		ambiguous plots		non-labeled plot		id. rate τ_i
I_1	1788	84,2%	330	15,5%	6	0,3%	90,9%
I_2	1697	79,9%	386	18,2%	41	1,9%	89,3%
I_3	796	37,5%	1306	61,5%	22	1%	75,7%
I_4	958	45,1%	1161	54,7%	5	0,2%	64,5%
I_5	541	25,5%	1583	74,5%	0	0%	63,5%

Table 2. Preliminary classification results

Table 3 lists the results obtained after the refinement of the classification. The result clearly shows the progression of the number of clear plots on all images (up to 54% for image I_3). Since the number of sample plots is low, the identification rate is only used to assess the coherence of the results. In this experiment, the identification rate remains reasonable for all images and even increases for the last three images of the series.

I_i	clear plots (number, γ_i)		ambiguous plots		non-labeled plot		id. rate τ_i
I_1	2040	96,1%	84	3,9%	0	0%	90,9%
I_2	1947	91,7%	177	8,3%	0	0%	90,5%
I_3	1954	92,0%	170	8,0%	0	0%	81,1%
I_4	1745	82,1%	379	17,9%	0	0%	69,2%
I_5	1550	73,0%	574	27,0%	0	0%	74,8%

Table 3. Refinement of the classification results

We have conducted another experiment where the number of images and the first image of the sequence have varied. The analysis of the evolution of clear plots is illustrated in Table 4. It shows the respective contributions of prediction and postdiction.

	I_1	I_2	I_3	I_4	I_5
I_1	-	**248**	**2**	**3**	**-1**
I_2	209	-	**41**	**0**	**0**
I_3	0	355	-	**743**	**60**
I_4	0	22	284	-	**481**
I_5	0	11	66	932	-

Table 4. Contribution of prediction and postdiction

Lines represent 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 355 clear plots in the prediction step, and images I_4 and I_5 contribute 743 and 60 clear plots respectively in the postdiction step.

The misclassification remaining after the refinement is due to three kinds of problems: 1) the good class does not belong to the preliminary classification results, 2) the plot is still ambiguous after the refinement and the decision task does not propose the good class, 3) the initialization of the model is not reliable enough to supply good prediction.

It is clear that one crucial point is the way the decision task selects a class among other candidates when the plot is still ambiguous after the refinement. In our case, the only choice criterion is the probability distribution resulting from the preclassification. In the next section, we try to remedy this problem by adding probabilities to the plot model in order to take profit of this information.

6 Probabilistic approach

In this section, we propose to extend timed automata to probabilities, in order to represent uncertainty on the transitions, and we define *probabilistic timed automata*. this later being performed using a traditional model-checking method. The interest of computing the probabilities of expected states is to be able to take into account the weight of the simulation of the model during the decision task.

Probabilistic timed automata are classical timed automata in which the locations and the transitions are associated to discrete probability distributions. As in timed automata, timing constraints are associated to *locations* (vertices of the graph) and to transitions.

The next-location probability is defined as follows.

Next-location probability. Let $\mathcal{S}$ be the location space of the system. Let $\theta(S)$ be the set of next-location probability distributions over the set of locations $\mathcal{S}$. Each $p \in \theta(S)$ is a function $p : \mathcal{S} \times \mathcal{S} \times \Phi(\mathcal{X}) \to [0,1]$. We denote $p(s, s', \lambda)$, with $\lambda \in \Phi(\mathcal{X})$, the probability that the system moves from the location s to the location s' at a time t with t satisfying the timing constraint λ. We require, for each timing constraint λ, that $\sum_{s' \in \mathcal{S}} p(s, s', \lambda) = 1$.

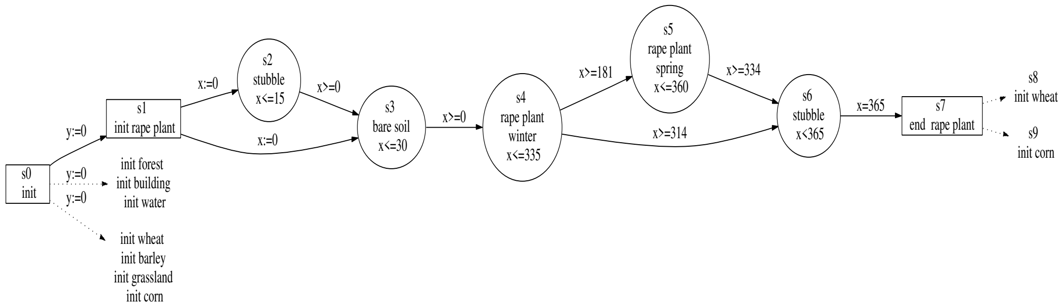
Probabilistic Timed Automata (PTA). A probabilistic timed automaton $\mathcal{A}$ is a tuple $< \mathcal{S}, \mathcal{X}, \mathcal{L}, \mathcal{E}, \mathcal{I}, \mathcal{P} >$ where:

- $\mathcal{S}$ is a finite set of locations and $s_o \in \mathcal{S}$ 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 $l \in \mathcal{L}$. The enabling condition is captured in φ and $\delta \subseteq \mathcal{X}$ gives the set of clocks to be reset when the edge is triggered.
- $\mathcal{I} : S \to \Phi(\mathcal{X})$ maps each location s with a timing constraint $I(s)$ called an invariant.
- $\mathcal{P} : S \to 2^{\theta(S)}$ associates to each $s \in \mathcal{S}$ a set of next-location probabilities.

The probability assigned to a timed transition is $p(s, s, \lambda)$ with $\lambda \in \Phi(\mathcal{X})$ satisfying $I(s)$ and the probability assigned to a discrete transition is $p(s, s', \lambda)$ with λ satisfying φ.

6.1 Plot evolution model

We now make the PTA framework more concrete in modeling the plot evolution. The timing constraints used in the next-location probability distributions correspond to time intervals inferred from the automaton invariants and guards. A next-location probability having as a timing constraint $true$ is independent of time. Figure 5 shows a fragment of the actual plot evolution model. It represents the rape plant cycle and part of the next-location probability distribution, only displayed for the location s_4. The entire PTA, used for our experimentation, contains 41 locations, 62 transitions and 100 next-location probabilities.



$\cdots \quad p(s_4, s_4, x < 181) = 1$	$p(s_4, s_4, x \geq 181 \wedge x < 314) = 0.7$	$p(s_4, s_5, x \geq 181 \wedge x < 314) = 0.3$
$p(s_4, s_4, x \geq 314 \wedge x \leq 335) = 0.3$	$p(s_4, s_5, x \geq 314 \wedge x \leq 335) = 0.1$	$p(s_4, s_5, x > 335) = 0$
$p(s_4, s_6, x \geq 314 \wedge x \leq 335) = 0.6$	$p(s_4, s_6, x > 335) = 1$	$\cdots$

Fig. 5. Part of the ATP modeling the plot evolution

6.2 Combining expectations and observations

This section explains how expected states and their respective probabilities are computed by making use of reachability analysis on a timed automaton. Lastly, we present the rule used to combine expectations and observations.

Probabilities of expected states. The objective of this step is to define the probabilistic distribution of the expected states, given the model and the set of possible scenarios. The probability of a scenario $K_{[s,t,s',t']}$ occurring from a location s at time t and leading to a location s' at time t' such that: $s \overset{e_1,t_1}{\rightarrow} s_1 \overset{e_2,t_2}{\rightarrow} \cdots s_{n-2} \overset{e_{n-1},t_{n-1}}{\rightarrow} s_{n-1} \overset{e_n,t_n}{\rightarrow} s'$, is defined as follows:

$$p(K_{[s,t,s',t']}) = p(s, s_1, \lambda_1)[\prod p(s_j, s_{j+1}, \lambda_{j+1})]p(s_{n-1}, s', \lambda_n)$$

with $j \in [1, n-2]$ and t_1 satisfying λ_1, t_{j+1} satisfying λ_{j+1} and t_n satisfying λ_n.
The probability to reach the location s' at time t' from the location s at time t is the sum of the probabilities of the v scenarios leading from the location s at time t to the location s' at time t'.

$$p(s', t'|s, t) = \sum p(K_{j_{[s,t,s',t']}}) \quad j \in [1, v]$$

We denote $p(s)$ the probability of the system to be in the location s at time t. The global probability of the location s' at time t' is then $p(s)p(s', t'|s, t)$. This value is renormalized in order to take into account all the possible successions leading to the location s.

$$p(s') = \frac{\sum_{s \in E_t^t} p(s)p(s', t'|s, t)}{\sum_{s'' \in \mathcal{S}} \sum_{s \in E_t^t} p(s)p(s'', t'|s, t)}$$

Combination rule . The method we use to combine the observations and the expectations is the renormalized product of observation and expectation probabilities. The probability distribution of observed states is easily derived from the probability distribution over classes resulting from the preliminary classification. The probability distribution of expected states is obtained as explained above.

Decision task. This section presents the decision rules used to choose the more realistic landcover type on ambiguous plots remaining after the refinement task. Three decision rules are considered:

maximum of probability The class C_{t_i} chosen for the plot P at time t_i is the one having the greatest probability on each image.

maximum of probability in prediction The class C_{t_i} chosen for the plot P at time t_i is, among the possible landcover successors of the class $C_{t_{i-1}}$ chosen for the plot P at time t_{i-1}, the one having the greatest probability.

maximum of probability in postdiction The class C_{t_i} chosen for the plot P at time t_i is, among the possible landcover predecessors of the class $C_{t_{i+1}}$ chosen for the plot P at time t_{i+1}, the one having the greatest probability.

6.3 Experimental results

This method has been applied on the same data as before. They concern a watershed supplying the city of of Rennes. The results, described in table 5, are obtained after the preliminary classification and after the refinement of the classification with the set-theoretical approach and with the probabilistic one. The uncertainty formalism daes not change the non-ambiguity rate τ_i since its effects concerns only the decision rule. After the preliminary classification, the class labeling the plot is the one having the maximum of probability in the observation. After the refinement with the probabilistic approach, the table present the three identification rates obtained according to the decision rule applied: *rule 1*) maximum of probability, *rule 2*) maximum of probability in prediction, *rule 3*) maximum of probability in postdiction.

	after the preliminary classification		after the refinement sets approach		after the refinement probabilistic approach			
I_i	γ_i	τ_i	γ_i	τ_i	γ_i	τ_i rule 1	τ_i rule 2	τ_i rule 3
I_1	84.2%	90.9%	96.1%	90.9%	96.1%	90.9%	90.9%	91.7%
I_2	79.9%	89.3%	91.7%	90.5%	91.7%	90.5%	90.5%	90.5%
I_3	37.5%	75.7%	92.0%	90.5%	92.0%	78.4%	79.7%	81.1%
I_4	45.1%	64.5%	82.1%	69.2%	82.1%	70.0%	68.2%	71.0%
I_5	25.5%	63.5%	73.0%	74.8%	73.0%	75.7%	74.8%	75.7%

Table 5. Comparison of results obtained with the different approaches

Although the low number of samples prevents us giving definite conclusions, the probabilistic approach using the decision *rule 3* appears to produce the better sequence.

7 Conclusion

This paper proposes a new approach to improve the quality of a sequence of dated observations. It takes profit of domain knowledge, here the dynamic evolution of a plot, and relies on combining expectations and observations. The formalism chosen to model the plot evolution, timed automata, defines in a explicit way uncertain temporal constraints on states. The evolution model is then undeterministic and imprecise. Because of noisy sensors and of the lack of precision of the preclassification, the observations are not necessary reliable. However, exploiting both allows us to discard some unrealistic hypotheses and to rank the remaining hypotheses more adequately. The experiment shows the interest of the method which increases the quality of each thematic map and ensures the global consistency of the sequence.

One original point is the cooperative use of prediction and postdiction mechanisms. An analysis of the results have shown that prediction and postdiction mechanisms contribute equally to the global improvement. We are currently experimenting the effects of a change in the initial states on the quality of the result. Another original point is to consider globally the sequence of observations in order to improve it, which is not the case in related domains as the Kalman filtering method in control theory or the belief change theory.

The choice of probabilities to express uncertainty is related to the fact that the preliminary classifier we are using provides us with probability distributions. However, we are currently studying the use of the Dempster-Shafer theory in order to be able to exploit the existing confusions known *a priori* between the classes of observations.

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. C. Cossart and C. Tessier. Filtering vs Revision and Update: Let us Debate! In *ECSQARU'99*, pages 116–127, 1999.
5. C. Daws, A. Olivero, S. Tripakis, and S. Yovine. The tool KRONOS. In LNCS 1066, editor, *Hybrid Systems III*, pages 208–219. Springer-Verlag, 1996.
6. 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.
7. T.A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994.
8. 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.

9. K. Larsen, P. Petterson, and W. Yi. UPPAAL in a nutshell. *Springer International Journal of Software Tools for Technology Transfer*, 1, 1997.
10. 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.
11. 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.
12. 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.
13. 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.
14. S. Yovine. Kronos : A verification tool for real-time systems. *International Journal of Software Tools for Technology Transfer*, 1, 1997.