Mining Pixel Evolutions in Satellite Image Time Series for Agricultural Monitoring

Andreea Julea[1], Nicolas Méger[2], Christophe Rigotti[3], Emmanuel Trouvé[2], Philippe Bolon[2], and Vasile Lăzărescu[4]

[1] Institute for Space Sciences, P.O. Box MG-23, Ro 077125, Bucharest-Măgurele, Romania
andreeamj@spacescience.ro
[2] Université de Savoie, Polytech'Savoie, LISTIC Laboratory, BP 80439, F-74944 Annecy-le-Vieux Cedex, France
{nicolas.meger,emmanuel.trouve,philippe.bolon}@univ-savoie.fr
[3] Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR 5205, F-69621, France
christophe.rigotti@insa-lyon.fr
[4] Politehnica University of Bucharest, Faculty for Electronics, Telecommunications and Information Technology, Applied Electronics and Information Engineering Department, Romania
vl@elia.pub.ro

Abstract. In this paper, we present a technique to help the experts in agricultural monitoring, by mining Satellite Image Time Series over cultivated areas. We use frequent sequential patterns extended to this spatiotemporal context in order to extract sets of connected pixels sharing a similar temporal evolution. We show that a pixel connectivity constraint can be partially pushed to prune the search space, in conjunction with a support threshold. Together with a simple maximality constraint, the method reveals meaningful patterns in real data.

Keywords: Satellite Image Time Series, Spatiotemporal Patterns, Constraints, Agricultural Monitoring.

1 Introduction

Current environmental and economic problems require better large scale agricultural monitoring. Continuous development of acquisition techniques of satellite images provides ever growing volumes of data containing precious information for environmental and agricultural remote sensing. It is now possible to gather series of images concerning a given geographical zone at a reasonable cost. This kind of datasets, termed as a Satellite Image Time Series (SITS), offers a great potential, but raises new analysis challenges as data volumes to be processed are large and noisy (e.g., atmospheric variations, presence of clouds), and as both the temporal and the spatial dimensions have to be taken into account.

We present an unsupervised technique to support SITS analysis in agricultural monitoring. The approach relies on frequent sequential pattern extraction [1]

P. Perner (Ed.): ICDM 2011, LNAI 6870, pp. 189–203, 2011.

along the temporal dimension, combined with a spatial connectivity criterion. It permits to exhibit sets of pixels that satisfy two properties of cultivated areas: being spatially connected/grouped and sharing similar temporal evolutions. The approach does not required prior knowledge of the objects (identified regions) to monitor and does not need user-supplied aggregate functions or distance definitions. It is based on the extraction of patterns, called *Grouped Frequent Sequential patterns* (GFS-patterns), satisfying a support constraint and a pixel connectivity constraint.

In this paper, we extend the general framework of GFS-patterns we proposed in [16], in two directions, when applied to agricultural monitoring.

Firstly, we show that, even though the connectivity constraint does not belong to a typical constraint family (e.g., monotonic, anti-monotonic), it can be pushed partially in the search space exploration, leading to significant reduction of execution times on real Satellite Image Time Series of cultivated areas.

Secondly, we show that a simple post-processing using a maximality constraint over the patterns is very effective, in the sense that it restricts the number of patterns to a human browsable collection, while still retaining highly meaningful patterns for agro-modelling, even on a poor quality input (rough image quantization, raw noisy images).

The new extended approach seems particularly appropriated in exploratory mining stages on this kind of data. Indeed, we show than the method can isolate cultivated fields vs. non-cultivated areas (city, path, field border), can find areas of homogeneous crop, and even highlight particular variety of a crop, and irrigation/fertilization differences. To our knowledge, no such coarse to fine grained results have been reported using a single other unsupervised method.

The technique does not aim to be exhaustive (e.g., identifying groups for all crops or varieties), but requires no domain knowledge (except the use of the well known Normalized Difference Vegetation Index [19]) and needs only a simple preprocessing of the SITS.

2 Grouped Frequent Sequential Patterns

In this section, the *grouped frequent sequential patterns* are introduced. They are dedicated to the extraction of groups of pixels, in which the pixels in a group share a common temporal pattern and satisfy a minimum average connectivity over space. Firstly, some preliminary definitions are given so as to view a SITS as a set of temporal sequences. Secondly, we recall and adapt in this context a common kind of local patterns, the so-called sequential patterns. Then, in the third part of this section, the connectivity measure used to define the grouped frequent sequential patterns is introduced.

2.1 Preliminary Definitions

Let us consider a SITS, i.e., a satellite image time series that covers the same area at different dates. Within each image, each pixel is associated to a value, e.g.,

the reflectance intensity of the geographical zone it represents. We transform these pixel values into values belonging to a discrete domain, using labels to encode pixel states. These labels can correspond to ranges obtained by image quantization or to pixel classes resulting from an unsupervised classification (e.g., using K-means or EM-based clustering).

Definition 1. ***(label and pixel state)*** Let $L = \{i_1, i_2, \ldots, i_s\}$ be a set containing s distinct symbols termed *labels*, and used to encode the values associated to the pixels. A *pixel state* is a pair (e, t) where $e \in L$ and $t \in \mathbb{N}$, and such that t is the occurrence date of e. The date t is simply the time stamp of the image from which the value e has been obtained.

Then, we define a *symbolic SITS* as a set of *pixel evolution sequences*, each sequence describing the states of a pixel over time.

Definition 2. ***(pixel evolution sequence and symbolic SITS)*** For a pixel p, the *pixel evolution sequence* of p is a pair $((x, y), seq)$, where (x, y) are the coordinates of p and seq is a tuple of pixel states $seq = \langle (e_1, t_1), (e_2, t_2), ..., (e_n, t_n) \rangle$ containing the states of p ordered by increasing dates of occurrences. A *symbolic SITS* (or SITS when clear from the context) is then a set of pixel evolution sequences.

For a typical symbolic SITS, we thus get a set of millions of pixel evolution sequences, each sequence containing the discrete descriptions of the values associated to a given pixel over the time.

2.2 Sequential Patterns

A typical base of sequences is a set of sequences of discrete events, in which each sequence has a unique sequence identifier. For SITS, if we take the pairs (x,y) of coordinates of the pixels as identifiers of their evolution sequences, then a symbolic SITS is a base of sequences, and the standard notions [1] of sequential patterns and sequential pattern occurrences can be easily defined as follows[1].

Definition 3. ***(sequential pattern)*** A *sequential pattern* α is a tuple $\langle \alpha_1, \alpha_2, \ldots, \alpha_m \rangle$ where $\alpha_1, \ldots, \alpha_m$ are labels in L and m is the *length* of α. Such a pattern is also denoted as $\alpha_1 \rightarrow \alpha_2 \rightarrow \ldots \rightarrow \alpha_m$.

Definition 4. ***(occurrence and support)*** Let $\mathcal{S}$ be a symbolic SITS, and $\alpha = \alpha_1 \rightarrow \alpha_2 \rightarrow \ldots \rightarrow \alpha_m$ be a sequential pattern. Then $((x, y), \langle (\alpha_1, t_1), (\alpha_2, t_2), \ldots, (\alpha_m, t_m) \rangle)$, where $t_1 < t_2 < \ldots < t_m$, is an *occurrence* of α in $\mathcal{S}$ if there exists $((x, y), seq) \in \mathcal{S}$ such that (α_i, t_i) appears in seq for all i in $\{1, \ldots, m\}$. Such a pixel evolution sequence $((x, y), seq)$ is said to support α. The *support* of α in $\mathcal{S}$, denoted by $support(\alpha)$, is simply the number of sequences in $\mathcal{S}$ that support α.

[1] Notice that in the original definitions several elements can occur at the same time in a sequence, while in our context a timestamp is associated to a single element.

Example 1. *A toy symbolic SITS containing the states of four pixels.*

$$((0,0),\langle(1,A),(2,B),(3,C),(4,B),(5,D)\rangle),$$
$$((0,1),\langle(1,B),(2,A),(3,C),(4,B),(5,B)\rangle),$$
$$((1,0),\langle(1,D),(2,B),(3,C),(4,B),(5,C)\rangle),$$
$$((1,1),\langle(1,C),(2,A),(3,C),(4,B),(5,A)\rangle)$$

This dataset describes the evolution of four pixels throughout five images with $L = \{A, B, C, D\}$. For example, the successive discrete labels associated to the values of the pixel located at $(0,0)$ are A, B, C, B and D. In this dataset, the sequential pattern $A \rightarrow C \rightarrow B$ has the four following occurrences (notice that the elements in an occurrence do not need to be contiguous in time):

$$((0,0),\langle(1,A),(3,C),(4,B)\rangle),$$
$$((0,1),\langle(2,A),(3,C),(4,B)\rangle),$$
$$((0,1),\langle(2,A),(3,C),(5,B)\rangle),$$
$$((1,1),\langle(2,A),(3,C),(4,B)\rangle)$$

The pattern has four occurrences, but appears in only three different pixel evolution sequences, and thus its support is $support(A \rightarrow C \rightarrow B) = 3$. Finally, it should be pointed out that a label can be repeated within a pattern, and for instance, pattern $C \rightarrow C$ has two occurrences, one in the third and one in the fourth sequence.

Definition 5. ***(frequent sequential pattern)*** Let σ be a strictly positive integer termed a *support threshold*. Let α be a sequential pattern, then α is a *frequent sequential pattern* if $support(\alpha) \geq \sigma$. The support threshold can also be specified as a relative threshold $\sigma_{rel} \in [0,1]$. Then a pattern α is frequent if $support(\alpha)/|\mathcal{S}| \geq \sigma_{rel}$, where $\mathcal{S}$ is the dataset and $|\mathcal{S}|$ is the number of sequences in $\mathcal{S}$.

Reusing the definitions of sequential patterns and of sequential patterns occurrences will enable to take advantage of the great research effort made in this domain to develop efficient extraction techniques (e.g., [1,28,21,10,31,30,25,27]).

2.3 Spatial Connectivity

The way sequential patterns are applied to SITS analysis leads to a natural interpretation of the notion of support. In fact, for a pattern α, the support of α is simply an area, i.e., the total number of pixels in the image having an evolution in which α occurs. These pixels are said to be *covered* by α.

Definition 6. ***(covered pixel)*** A pixel associated to the evolution sequence $((x,y), seq)$ is *covered* by a sequential pattern α if α has at least one occurrence in seq. The set of the coordinates of the pixels covered by α is denoted by $cov(\alpha)$. By definition, $|cov(\alpha)| = support(\alpha)$.

However, a threshold on the covered area is not sufficient, because, most of the time, interesting parts in images are made of pixels forming regions in space. Thus, an additional criterion, the *average connectivity* measure, based on the *8-nearest neighbors (8-NN)* convention [8], is introduced. This measure enables to select patterns that cover pixels having a tendency to form groups in space. It is defined as follows:

Definition 7. ***(local connectivity)*** For a symbolic SITS $\mathcal{S}$, let $occ((x,y),\alpha)$ be a function that, given the spatial coordinates (x,y) and a sequential pattern α, indicates whether α occurs in $\mathcal{S}$ at location (x,y). More precisely, $occ((x,y),\alpha)$ is equal to 1 if and only if there is a sequence seq in $\mathcal{S}$ at coordinates (x,y) and α occurs in $((x,y),seq)$. Otherwise $occ((x,y),\alpha)$ is equal to 0. If α occurs in $((x,y),seq)$, then its *local connectivity* at location (x,y) is $LC((x,y),\alpha) = [\sum_{i=-1}^{i=1}\sum_{j=-1}^{j=1} occ((x+i,y+j),\alpha)] - 1$.

The value $LC((x,y),\alpha)$ is the number of pixels in the 8-neighborhood of (x,y) that have an evolution supporting α. The reader should notice that the sum is decremented by one, so as not to count the occurrence of α at location (x,y) it-self. In Example 1, for sequential patterns $A \rightarrow C \rightarrow B$ and $C \rightarrow C$ we have:

$$\begin{gathered} LC((0,0), A \rightarrow C \rightarrow B) = 2 \\ LC((0,1), A \rightarrow C \rightarrow B) = 2 \\ LC((1,1), A \rightarrow C \rightarrow B) = 2 \\ LC((0,1), C \rightarrow C) = 1 \\ LC((1,1), C \rightarrow C) = 1 \end{gathered}$$

Definition 8. ***(average connectivity)*** The *average connectivity* of α is defined as:

$$AC(\alpha) = \frac{\sum_{(x,y)\in cov(\alpha)} LC((x,y),\alpha)}{|cov(\alpha)|}$$

This measure gives, for the pixels supporting α, the average number of neighbors in their 8-NN that also support α. In Example 1, $AC(A \rightarrow C \rightarrow B) = 6/3 = 2$ and $AC(C \rightarrow C) = 2/2 = 1$. Finally, we define the *grouped frequent sequential patterns* as follows.

Definition 9. ***(GFS-pattern)*** Let $\mathcal{S}$ be a symbolic SITS, given a sequential pattern α frequent in $\mathcal{S}$, and a positive real number κ termed *average connectivity threshold*, α is said to be a *Grouped Frequent Sequential pattern* (*GFS-pattern*) if $AC(\alpha) \geq \kappa$ in $\mathcal{S}$.

For instance, in Example 1, if $\sigma = 2$ and if $\kappa = 2$, then $A \rightarrow C \rightarrow B$ is a grouped frequent sequential pattern while $C \rightarrow C$ is not.

3 Grouped Frequent Sequential Pattern Extraction

As mentioned in Section 2, several efficient techniques are available to extract sequential patterns in a base of sequences and can be used in our context. A naive

solution is to extract frequent sequential patterns and then, in a post-processing step, to select among them the ones satisfying the average connectivity constraint $AC(\alpha) \geq \kappa$. In this section, we show that this constraint can be pushed partially in the extraction process to prune the search space and reduce the extraction time, as reported in the experiment presented in Section 4.2.

The average connectivity constraint does not correspond to a class of constraints that have been identified in sequential pattern mining, and for which pruning techniques have been proposed. The two main classes of constraints are the anti-monotonic constraints (if a pattern does not satisfy the constraint then its super-patterns cannot satisfy it) and monotonic constraints (if a pattern satisfies the constraint then all its super-patterns satisfy it).

For the simple form of sequential patterns used in this paper, the notion of *super-patterns* can be defined as follows.

Definition 10. ***(super-pattern)*** A sequential pattern $\beta = \beta_1 \rightarrow \beta_2 \rightarrow \ldots \rightarrow \beta_m$ is a *super-pattern* of a sequential pattern $\alpha = \alpha_1 \rightarrow \alpha_2 \rightarrow \ldots \rightarrow \alpha_n$ if there exist integers $1 \leq i_1 < i_2 < \ldots < i_n \leq m$ such that $\alpha_1 = \beta_{i_1}$, $\alpha_2 = \beta_{i_2}$, ..., $\alpha_n = \beta_{i_n}$.

It is straightforward that the average connectivity constraint is neither anti-monotonic, nor monotonic, and it is easy to show that it is neither prefix anti-monotonic, nor prefix monotonic [27]. Moreover it does not belong to classes of constraints used for frequent pattern mining in general, such as succinct [24], convertible [26] or loose anti-monotone [2].

The key hints to push partially the average connectivity constraint is to observe that for any frequent sequential pattern α since $|cov(\alpha)| \geq \sigma$, then

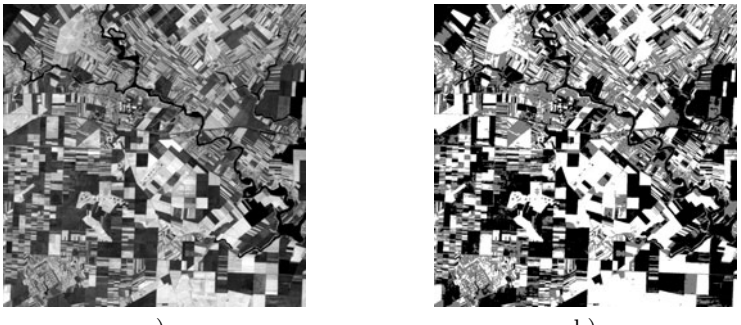
$$AC(\alpha) = \frac{\sum_{(x,y)\in cov(\alpha)} LC((x,y),\alpha)}{|cov(\alpha)|} \leq \frac{\sum_{(x,y)\in cov(\alpha)} LC((x,y),\alpha)}{\sigma}$$

Thus a frequent pattern α that does not satisfy $\frac{\sum_{(x,y)\in cov(\alpha)} LC((x,y),\alpha)}{\sigma} \geq \kappa$ cannot be a GFS-pattern. And, if we consider the conjunction of constraints $\mathcal{C} = support(\alpha) \geq \sigma \wedge \frac{\sum_{(x,y)\in cov(\alpha)} LC((x,y),\alpha)}{\sigma} \geq \kappa$, this conjunction is anti-monotonic, since the value $\sum_{(x,y)\in cov(\alpha)} LC((x,y),\alpha)$ cannot increase for super-patterns of α, and thus this conjunction can be used actively to prune the search space.

There is no real need for a new extraction algorithm, since many, if not all, of the sequential pattern mining algorithms can handle and push in the extraction process anti-monotonic constraints. We decided to integrate the anti-monotonic conjunction $\mathcal{C}$ into the *PrefixGrowth* algorithm [27], that is a recent and efficient algorithm for sequential pattern mining under constraints, and that can easily handle anti-monotonic constraints among others. Beside checking $\mathcal{C}$ to prune the search space, the only modification required is to verify before outputting a pattern α that $AC(\alpha) \geq \kappa$, since satisfying $\mathcal{C}$ does not imply satisfying the average connectivity constraint. The implementation of the whole algorithm has been done in C using our own data structures.

4 Experiments

We report experiments on the ADAM (Data Assimilation by Agro-Modeling) SITS [6], a SITS dedicated to the assessment of spatial data assimilation techniques within agronomic models. This dataset and its preprocessing are presented in Section 4.1, and result in a set of one million of sequences of size 20. In Section 4.2, we show that pushing the average connectivity measure constraint, during GFS-pattern extraction, is effective to reduce the search space. Then, in Section 4.3, we show that together with a maximality constraint, the approach is useful to find meaningful patterns in real data. All experiments have been run on a standard PC (Intel Core 2 @3GHz, 4 GB RAM, Linux kernel 2.6), using our own extractor engine developed in C (see Section 3).



a) b)

Fig. 1. Satellite NDVI images examples a) original image b) quantization of the image with 3 intervals

4.1 The ADAM SITS: Presentation, Selection and Preprocessing

We build a dataset of one million of sequences of size 20, using 20 images (1000 $\times$ 1000) of the ADAM SITS taken between October 2000 and July 2001, so as to make sure that enough data is available to observe agricultural cycles, from autumn ploughing and seeding to harvest. These images have been acquired with three bands by SPOT satellites: B1 in green (0.5 - 0.59 μm), B2 in red (0.61 - 0.68 μm) and B3 in near infrared (NIR 0.78 - 0.89 μm). The spatial resolution is 20m$\times$20m and the observed scene is a rural area located in East Bucharest, Romania.

A sub-scene (containing 1000 $\times$ 1000 pixels) depicting a given area, namely Fundulea, has been selected. The resulting dataset contains noise (mainly atmospheric perturbations), and has a size (20 images 1000 $\times$ 1000) typical in the domain of per-pixel SITS analysis. This sub-scene mainly shows agricultural fields whose dimensions are larger than the spatial resolution. Various types of crops such as wheat, corn, barley, chickpea, soya, sunflower, pea, millet, oats or lucerne are present. Other objects can be categorized into 'roads', 'rivers', 'forests' and 'towns'. The topography of this region is generally flat with a very

limited fraction of the area corresponding to slopes bordering a river and to several micro-depressions. A ground truth is available for the period 2000-2001 for the fields that belong to the Romanian National Agricultural Research and Development Institute. It represents 5.9% of the scene, and can be used to evaluate our results.

For each pixel, and for each date, we compute a synthetic band B4 corresponding to the *Normalized Difference Vegetation Index* (NDVI) [19] and defined as $B4 = \frac{B3-B2}{B3+B2}$. The NDVI index is widely used to detect live green plant canopies in multispectral remote sensing data. An example of an original image of the ADAM SITS encoded in the B4 band is presented in Figure 6a. The image quantization is performed by splitting the B4 value domain in 3 intervals that are equally populated. In order to minimize the influence of possible calibration defaults, quantization is separately done for each image. For a given acquisition date, a pixel is described by a single label that indicates which interval this pixel value belongs to. Label 1 relates to low NDVI values, label 2 represents mid NDVI values and label 3 denotes high NDVI values. The result of the quantization of the image of Figure 6a is shown in Figure 6b. When encoded as sequences, we obtain a set of one million of sequences of size 20 over an alphabet of 3 symbols.

4.2 Quantitative Results

The two parameters that can be set by the user are σ, the minimum support and κ, the minimum average connectivity. The values of the minimum support are taken in the range $[0.25\%, 2\%]$ so as to ask for minimum areas covering from 2500 pixels (1 km^2) to 20000 pixels (8 km^2). Those values allow us either to consider all fields including the smallest ones (low σ values) or to extract quite large fields (high σ values). In order to assess the impact of κ, values between 0 and 7 are considered. As the definition of the average connectivity measure relies on the 8-nearest neighbors convention, and makes no distinction between pixels on image borders and the other ones, the average connectivity measure indeed belongs to $[0, 8)$.

The experiments show that the number of frequent sequential patterns that are discarded thanks to the minimum average connectivity constraint is important, and that pushing partially this constraint leads to a significant reduction of the execution times (from 10 to 20%).

The number of output patterns N_p can be several orders of magnitude lesser than the total number of frequent patterns. This is represented in Figure 2a. If no minimum average connectivity constraint is applied ($\kappa = 0$), then all frequent sequential patterns are extracted, and N_p rises up to 78885 patterns, and as expected, the higher κ is, the lower is N_p. In the worst case scenario, i.e., for $\sigma = 0.25\%$, if $\kappa = 4$, then $N_p = 7623$ while if $\kappa = 7$ then $N_p = 21$. The minimum average connectivity constraint is a very selective one, as it can be observed, for a given value of κ such that $\kappa \neq 0$, N_p has rather limited variations with respect to σ. For example, for $\kappa = 4$, N_p rises from 4042 ($\sigma = 2\%$) to 7623 GFS-patterns ($\sigma = 0.25\%$) while for $\kappa = 6$, N_p rises from 454 ($\sigma = 2\%$) to 479 GFS-patterns ($\sigma = 0.25\%$). N_p is even stable for $\kappa = 7$ with 21 GFS-patterns.

As presented in Figure 2b, the extraction times are the same for all values of κ if the average connectivity constraint is not pushed (one single curve). If the constraint is pushed, then extraction times are reduced for all settings, from 10% up to 20%. For example, for $\sigma = 0.75\%$ and $\kappa = 7$, it takes 756 seconds to perform an extraction without constraint pushing while it only takes 599 seconds with constraint pushing.

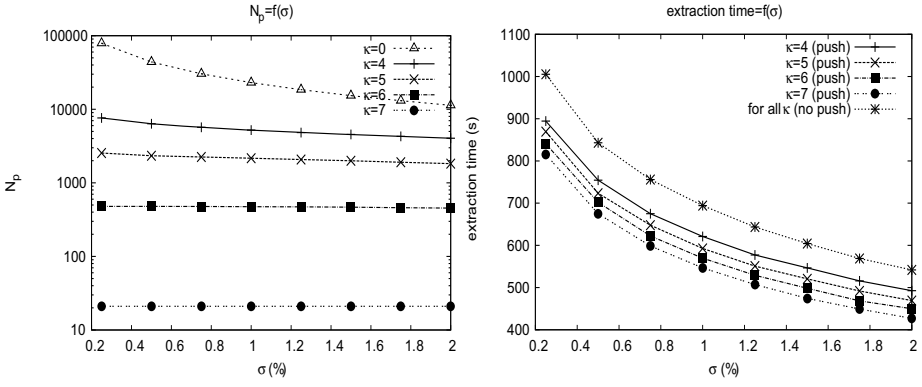


Fig. 2. For different values of κ a) N_p vs. σ b) Extraction times vs. σ, with and without constraint pushing

The corresponding pruning can be quantified using the number $N_{checked}$ of frequent sequential patterns that are considered during the extraction and for which the average connectivity constraint is checked. The values obtained for $N_{checked}$ are given Figure 3a. If no constraint pushing is performed, then, for example, $N_{checked}$ rises up to 78885 patterns for $\sigma = 0.25$ (whatever κ might be). At the same support threshold, if the constraint pushing is performed, then, for instance, with $\kappa = 7$, $N_{checked}$ goes down to 50227. For a given σ and a given κ, when the constraint is pushed, $N_{checked}$ is reduced in all settings. This reduction (in %) is depicted in Figure 3b. It varies between 7.7% ($\sigma = 2\%$,$\kappa = 4$) and 36.3% ($\sigma = 0.25\%$,$\kappa = 7$). The pruning is more effective (large relative reduction) in the most difficult extraction settings (low values of σ).

4.3 Qualitative Results

In this section, σ is set to 1% in order to ask for GFS-patterns relating to areas covering at least 4 km^2 (the whole image covers 400 km^2). Main crops are thus focused on, which will help us in characterizing our results. The ground truth that has been made available by the experts and that covers 5.9% of the image indeed contains representative crops of that region.

We show that using a typical maximality constraints on these patterns is a very effective way to focus on a small number of meaningful GFS-patterns, still carrying key information for agro-modelling experts.

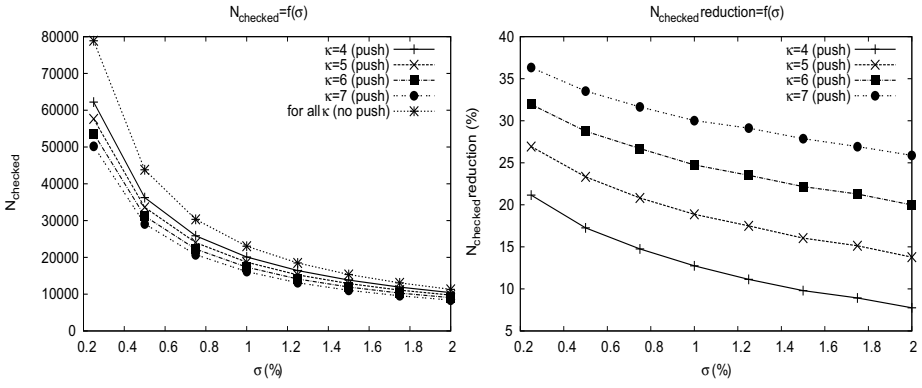


Fig. 3. For different values of κ a) $N_{checked}$ vs. σ b) $N_{checked}$ reduction vs. σ

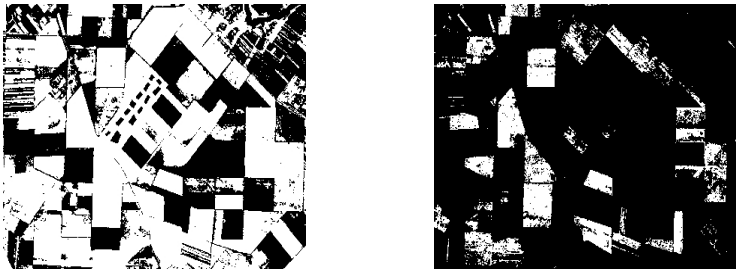
The maximality constraint used is very simple, it consist in selecting the patterns in the output having no super-pattern also present in the output. These patterns are in some sense the most specific.

To visualize the result, for each of these maximal patterns we draw an image where the pixels covered by the pattern are highlighted. Since we obtain only a few tens of such images, the visual inspection can be quickly done by the expert. Notice that if we extract all frequent sequential patterns (without taking into account the spatial connectivity of the pixels) at $\sigma = 1\%$, then 23038 patterns are obtained, among which 4684 are maximal, forming a collection that cannot reasonably be handle by the expert.

It should also be pointed out, that in these experiments, the image quantization does not seem to be a critical issue, as well as the presence of intrinsic noise in SITS (mainly atmospheric variations and clouds). Indeed, though the image quantization in 3 levels leads to patterns built over a small alphabet of 3 labels, and though no dedicated noise preprocessing is performed, the joint use of the spatial and temporal information still allows to find meaningful patterns. So the technique is likely to be applicable to poor quality image series (e.g., due to limitations of the measuring device) and to require little preprocessing.

For the first experiment we set κ to 7, that is a very selective value of the threshold. In this case, 21 GFS-patterns are obtained. They relate to general evolutions as their length does not exceed 12. Only 7 are maximal, and among them we have for example, pattern $3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3$. The pixels covered by that pattern are depicted in white in Figure 4a over the area for which the ground truth is available. It covers 96.2% of the pixels of the ground truth that correspond to cultivated fields, and 98.3% of the pixels it covers in this area correspond to cultivated fields.

In order to get more specific evolutions, i.e., longer patterns, we set κ to a less selective value and use $\kappa = 6$. We obtain 31 maximal patterns out of the 474 GFS-patterns that are extracted. One of these maximal patterns is



a) b)

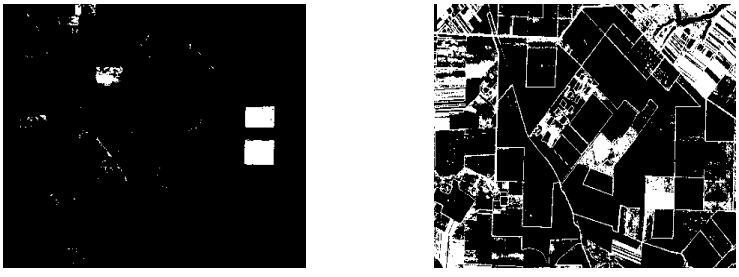
Fig. 4. a) Localization of pattern $3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3$ b) Localization of pattern $2 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1$

$2 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1$. The pixels covered by that pattern are represented in Figure 4b. According to the ground truth, it covers 61.4% of the pixels of the ground truth that relate to wheat crop, and 96.3% of the pixels it covers in the area where the ground truth is available, correspond to wheat crop.

Interesting information can be drawn from such patterns. For instance, as it can be observed, some *holes* (small black areas) appear within the fields (large polygon almost completely filled in white) in Figure 4a and in Figure 4b. The pixels of those holes are not covered by the pattern covering the ones in the white areas. Their temporal behavior is thus different from their surrounding pixels though they should be related to the same crops. Some of those holes match pedological differences that have been reported by the experts while other holes are likely to be due to different fertilization and/or irrigation conditions. Such information is particularly interesting as it can be used to adapt locally soil fertilization or irrigation.

Furthermore, it is possible to extract patterns corresponding to a single variety of a given crop. For example, with $\kappa = 5.5$ we have 1074 GFS-patterns, and 66 of them are maximal. Among these maximal ones, we have pattern $3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1$. Figure 5a gives its localization. While the previous pattern relates to wheat crop in general, that one relates to a particular variety. Indeed, 98.8% of the pixels it covers in the ground truth area are all of a same variety of wheat. Two rectangular fields are clearly identified (right part of the picture), the upper one corresponds to an area partially covered by the previous pattern, while this is not the case for the other rectangle, that exhibits another field of wheat. Both rectangles are covered by the general pattern corresponding to cultivated fields and shown Figure 4a.

The pixels covered by the patterns do not always correspond to cultivated areas, for instance, for $\kappa = 6$ we also obtained as a maximal GFS-pattern $2 \rightarrow 2 \rightarrow 2 \rightarrow 2 \rightarrow 2 \rightarrow 2 \rightarrow 2$ that corresponds to paths, fallows, cities and field borders. Its localization is depicted in Figure 5b.



a) b)

Fig. 5. a) Localization of pattern $3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1$ b) Localization of pattern $2 \rightarrow 2 \rightarrow 2 \rightarrow 2 \rightarrow 2 \rightarrow 2 \rightarrow 2 \rightarrow 2$

5 Related Work

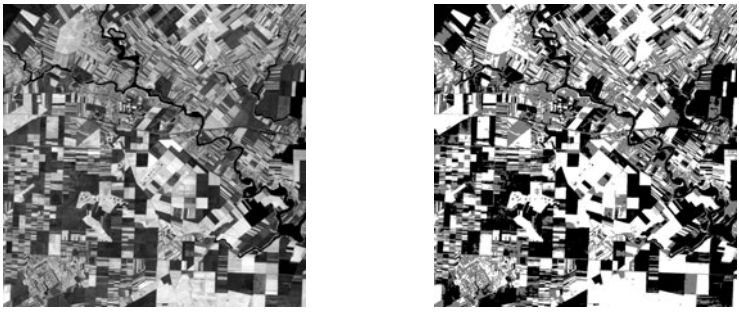
SITS can be processed at a higher level than the pixel one, after having identified objects or groups of pixels forming regions of interest (e.g., [12,13]). This family of approaches, needs as input identified objets/regions. If not known, objects/regions are hard to select in SITS since groups of pixels do not always form objects in a single image[2] (e.g., because of atmospheric perturbations, shading phenomenon).

Per-pixel analysis of SITS have also retained attention as they do not require prior object identification. These techniques are essentially clustering techniques to form clusters of pixels (e.g., [23,9,17] These approaches are the closest to the one presented in this paper, in the sense that they perform per-pixel analysis without prior knowledge of the objects (identified regions) to monitor. However, they required to incorporate domain knowledge in the form of feature/aggregation/distance definitions and and they do not find overlapping areas, and areas that refine other areas, such as the ones presented in Section 4.

Other approaches, based on change detection, generate a single image in which changes are plotted (e.g., [15,29]). They require prior information about the type of changes and are targeted to a specific phenomenon, e.g., earthquakes or biomass accumulation. They can be applied at the pixel level (e.g., [7], [20]), at the texture level [18] or at the object level (e.g., [3]).

Other works (e.g., [4,5,22,14,11]) rely on local patterns for analyzing trajectories and neighborhoods in spatio-temporal datasets. Nevertheless, to our knowledge, they reported no application to satellite image time series.

[2] This cannot be easily overcome, for instance, by averaging pixel values over consecutive images, since the aspect of an object is likely to change from a image to the next one.



a) b)

Fig. 6. Satellite NDVI images examples a) original image b) quantization of the image with 3 intervals

6 Conclusion

In this paper, we applied the GFS-patterns to extract sets of pixels sharing similar evolution from Satellite Image Time Series over cultivated areas. Beside having a common temporal evolution, such a set of pixels must be populated enough (support constraint) and connected enough (average connectivity constraint). We showed that the connectivity constraint can be partially pushed to prune the search space and that using a simple maximality constraint allows to focus on small collections of patterns that are easy to browse and interpret.

The experiments also showed that, even on poor quality input (i.e., noisy images, rough quantization), the method can exhibit various level of details of primary interest in agro-modelling (i.e., cultivated vs. non-cultivated areas, types of crops, varieties).

Acknowledgments. The authors thank the French Research Agency (ANR) for supporting this work through the EFIDIR project (ANR-2007-MCDC0-04, www.efidir.fr) and FOSTER project (ANR-2010-COSI-012-02, foster.univ-nc.nc). They also thank the ADAM project and the CNES agency for making the data available. Finally, the authors express their gratitude to Roxana Vintila (Research Institute for Soil Science and Agrochemistry - Bucharest, Romania) and to Gheorghe Petcu (National Agricultural Research and Development Institute Fundulea, Romania) for supplying the ground truth of the regions that we studied through acquisitions of the ADAM project.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proc. of the 11th International Conference on Data Engineering (ICDE 1995), Taipei, Taiwan, pp. 3–14. IEEE Computer Society Press, Los Alamitos (1995)

2. Bonchi, F., Lucchese, C.: Pushing tougher constraints in frequent pattern mining. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 114–124. Springer, Heidelberg (2005)
3. Bontemps, S., Bogaert, P., Titeux, N., Defourny, P.: An object-based change detection method accounting for temporal dependences in time series with medium to coarse spatial resolution. Remote Sensing of Environment 112, 3181–3191 (2008)
4. Cao, H., Mamoulis, N., Cheung, D.W.: Mining frequent spatio-temporal sequential patterns. In: Proc. of the Fifth IEEE International Conference on Data Mining (ICDM 2005), Houston, Texas, USA, pp. 82–89 (2005)
5. Cao, H., Mamoulis, N., Cheung, D.W.: Discovery of periodic patterns in spatiotemporal sequences. IEEE Transaction on Knowledge and Data Engineering 19(4), 453–467 (2007)
6. Centre National d'Etudes Spatiales. Database for the Data Assimilation for Agro-Modeling (ADAM) project, `http://www.kalideos.cnes.fr/index.php?id=accueil-adam`
7. Coppin, P., Jonckheere, I., Nackaerts, K., Muys, B., Lambin, E.: Digital change detection methods in ecosystem monitoring: a review. International Journal of Remote Sensing 25(9), 1565–1596 (2004)
8. Fisher, R., Dawson-Howe, K., Fitzgibbon, A., Robertson, C., Trucco, E.: Dictionary of Computer Vision and Image Processing. John Wiley and Sons, New York (2005)
9. Gallucio, L., Michel, O., Comon, P.: Unsupervised clustering on multi-components datasets: Applications on images and astrophysics data. In: Proc. of the 16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, pp. 25–29 (2008)
10. Garofalakis, M., Rastogi, R., Shim, K.: Spirit: Sequential pattern mining with regular expression constraints. In: Proc. of the 25th International Conference on Very Large Databases (VLDB 1999), Edinburgh, United Kingdom, pp. 223–234 (1999)
11. Gudmundsson, J., Kreveld, M., Speckmann, B.: Efficient detection of patterns in 2D trajectories of moving points. Geoinformatica 11(2), 195–215 (2007)
12. Héas, P., Datcu, M.: Modeling trajectory of dynamic clusters in image time-series for spatio-temporal reasoning. IEEE Transactions on Geoscience and Remote Sensing 43(7), 1635–1647 (2005)
13. Honda, R., Konishi, O.: Temporal rule discovery for time-series satellite images and integration with RDB. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 204–215. Springer, Heidelberg (2001)
14. Huang, Y., Zhang, L., Zhang, P.: A framework for mining sequential patterns from spatio-temporal event data sets. IEEE Transactions on Knowledge and Data Engineering 20(4), 433–448 (2008)
15. Inglada, J., Favard, J.-C., Yesou, H., Clandillon, S., Bestault, C.: Lava flow mapping during the Nyiragongo January, 2002 eruption over the city of Goma (D.R. Congo) in the frame of the international charter space and major disasters. In: Proc. of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2003), Toulouse, France, vol. 3, pp. 1540–1542 (2003)
16. Julea, A., Meger, N., Bolon, P., Rigotti, C., Doin, M.-P., Lasserre, C., Trouve, E., Lazarescu, V.: Unsupervised spatiotemporal mining of satellite image time series using grouped frequent sequential patterns. IEEE Transactions on Geoscience and Remote Sensing 49(4), 1417–1430 (2011)
17. Ketterlin, A., Gançarski, P.: Sequence similarity and multi-date image segmentation. In: Proc. of the 4th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MULTITEMP 2007), Leuven, Belgium, pp. 1–4 (2007)

18. Li, L., Leung, M.K.H.: Robust change detection by fusing intensity and texture differences. In: Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Kauai Marriott, Hawaii, pp. 777–784 (2001)
19. Lillesand, T.M., Kiefer, R.W.: Remote Sensing and Image Interpretation, 4th edn. John Wiley and Sons, New York (2000)
20. Lu, D., Mausel, P., Brondizio, E., Moran, E.: Change detection techniques. International Journal of Remote Sensing 25(12), 2365–2407 (2004)
21. Masseglia, F., Cathala, F., Poncelet, P.: The PSP approach for mining sequential patterns. In: Żytkow, J.M. (ed.) PKDD 1998. LNCS (LNAI), vol. 1510, pp. 176–184. Springer, Heidelberg (1998)
22. Nanni, M., Pedreschi, D.: Time-focused clustering of trajectories of moving objects. Journal of Intelligent Information Systems 27(3), 267–289 (2006)
23. Nezry, E., Genovese, G., Solaas, G., Rémondière, S.: ERS - Based early estimation of crop areas in Europe during winter 1994-1995. In: Proc. of the 2nd International Workshop on ERS Application, London (1995)
24. Ng, R.T., Lakshmanan, L.V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. In: Proc. of the ACM SIGMOD International Conference on Management of Data (SIGMOD 1998), Seattle, Washington, USA, pp. 13–24 (1998)
25. Pei, J., Han, B., Mortazavi-Asl, B., Pinto, H.: Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: Proc. of the 17th International Conference on Data Engineering (ICDE 2001), Heidelberg, Germany, pp. 215–226 (2001)
26. Pei, J., Han, J., Lakshmanan, L.V.S.: Mining frequent itemsets with convertible constraints. In: Proc. of the 17th International Conference on Data Engineering (ICDE 2001), Heidelberg, Germany, pp. 433–442 (2001)
27. Pei, J., Han, J., Wang, W.: Constraint-based sequential pattern mining: the pattern-growth methods. Journal of Intelligent Information Systems 28(2), 133–160 (2007)
28. Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057, pp. 3–17. Springer, Heidelberg (1996)
29. Vina, A., Echavarria, F.R., Rundquist, D.C.: Satellite change detection analysis of deforestation rates and patterns along the colombia-ecuador border. AMBIO: A Journal of the Human Environment 33, 118–125 (2004)
30. Zaki, M.J.: Sequence mining in categorical domains: incorporating constraints. In: Proc. of the 9th International Conference on Information and Knowledge Management (CIKM 2000), Washington, DC, USA, pp. 422–429 (2000)
31. Zaki, M.J.: Spade: an efficient algorithm for mining frequent sequences. Machine Learning 42(1/2), 31–60 (2001)