6
Implementation of the proposed multitemporal classification method in the multitemporal framework

During the course of this research the InterIMAGE framework was extended with the multitemporal functionalities proposed in the last chapter. As a way to validate the extended framework, the method introduced in Chapter 3 was implemented on it, with the aid of these new functionalities.

It should be noted that the framework is not bound to any particular multitemporal method, and in fact it was designed to accept a variety of such methods. Having that stated, the objective of this chapter is to show how the different variants of the proposed multitemporal method can be structured through different knowledge models, considering temporal knowledge representation, external structural and temporal operators, and interpretation strategies.

It is also important to observe that the purpose of this chapter is not to further evaluate the devised multitemporal method, but rather to prove that given the same conditions, with the same transition matrixes and monotemporal classifiers trained in the same fashion, the outcome of the method implemented on the framework is similar to that obtained in the environment it was originally implemented. Moreover, in order to produce comparable results, the same segmentation and monotemporal classification methods, described respectively in sections 4.1.1 and 4.2, were used in the implementation of the structural top-down operators described in the next section.

6.1.6.1. Description of the operators

Initially, let's consider a land cover interpretation knowledge model represented by the semantic network in Figure 38, where t and $t+1$ are temporal nodes, and the ω_i nodes represent land cover classes at both times.

The semantic network depicted in Figure 38 can be used in the design of different knowledge models, characterized by the particular node attributes, top-down and bottom-up operators. In the next sections different choices of structural and temporal operators will be presented.

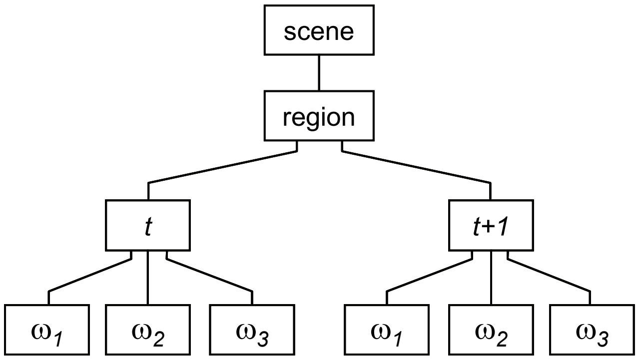

Figure 38. Land-cover multitemporal interpretation problem, ω_i correspond to land cover classes.

6.1.1. Structural operators

The first issue regarding knowledge models based on the semantic network in Figure 38 is how the geographic regions associated to the *Region* concept are defined. There are two basic alternatives.

In the fist alternative, segments[14] associated to the *Region* object hypotheses are defined by a structural top-down operator (see Section 5.2.2) attached to the *Region* node, and each segment is later classified by the top-down operators attached to the ω_i nodes. In this case the segments classified at each date will coincide spatially, in other words, the exact same geographic region would be classified by structural top-down operators attached to nodes *t* and *t+1*. The structural top-down operators attached to the ω_i nodes would have the sole task of assigning to the object hypotheses confidence values, which are in fact the grade of membership to the ω_i classes at each date. In this alternative, therefore, for each segment generated in the top-down phase at the *Region* node, one instance for each ω_i node is created.

[14] The term *segment* is used here with the same meaning of the term *region*, used to denote a geographical region associated to an object hypothesis or instance in previous sections. The use of the term *segment* in this section is meant to avoid confusion with the semantic network node named *Region*.

In the second alternative a dummy structural top-down operator (see Section 5.1.2) is attached to the *Region* node, and (structural) holistic top-down operators are attached to the ω_i nodes. Those holistic operators are responsible for segmenting the images and assigning to the corresponding object hypotheses confidence values that are equal to the membership to the ω_i classes.

Considering that the same segmentation procedure described in Section 4.1.1 was used in the experiments reported in Section 6.2, and that such segmentation procedure generated more than 18.000 segments, the second alternative was chosen, mainly for performance reasons – in the first alternative, a processing thread would have to be executed for each of the thousands of segments. Therefore it was decided to attach a dummy top-down operator to the *Region* node.

In a way consistent to that choice, considering the monotemporal classifiers ${}^{E}\mathbf{C}$ and ${}^{L}\mathbf{C}$ defined in Section 3.3, two multi-class top-down operators were implemented and attached to all ω_i nodes of each temporal branch. These operators segment the images using the same procedure described in Section 4.1.1, actually generating coincident segments for each time. For each segment, at each time, the operator generates one hypothesis for each land-cover class, assigning to those hypotheses confidence values that are equal to the membership values calculated through the monotemporal classification procedures described in Section 4.2.

Actually a single parametric structural top-down operator was implemented. The operator takes as input the images of all times, a digital terrain model and a drainage map (used to calculate the distance from the image objects to water bodies), and also a time-tag that tells it from which image it should take the feature values and calculate class membership values. Further inputs of the operator are the vector of spectral mean values and covariance matrixes, respectively $\bar{\mathbf{x}}_{\varpi i}$ and $\mathbf{\Sigma}_{\varpi i}$ (see Section 4.2), that correspond to each of the ω_i land cover classes.

The operator initially produces image segment using the procedure described in Section 4.1.1 and then, for each segment, creates one hypothesis per land cover class. Then it assigns for each hypothesis a confidence value that is equal to the membership produced by the classification procedure described in

Section 4.2. This operator is therefore equivalent to the coupling of the segmentation procedure to the monotemporal classifiers ${}^{E}\mathbf{C}$ and ${}^{L}\mathbf{C}$. Such operator will hereafter be denoted either as *monotemporal structural top-down operator* ${}^{E}\mathbf{C}$ or as *monotemporal structural top-down operator* ${}^{L}\mathbf{C}$ (indicating that it took as the time-tag parameter either the earlier or later date, which will vary for each experiment).

Figure 39 indicates schematically the placement of the monotemporal structural top-down operators in the multitemporal land-cover classification knowledge models designed for the validation experiments.

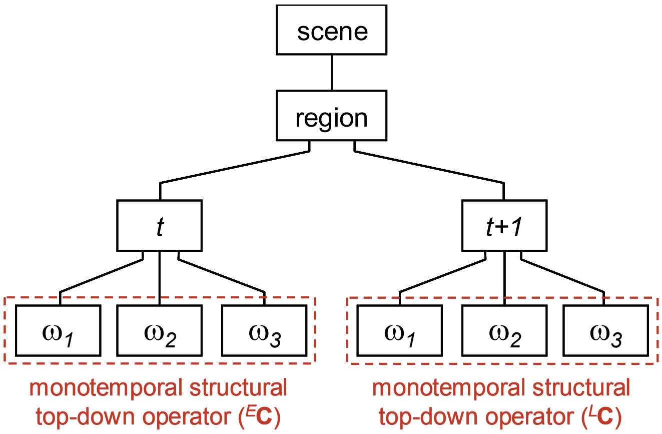

Figure 39. Placement of the structural monotemporal top-down operators in the land-cover classification knowledge models.

6.1.2. Temporal operators

Before discussing the temporal operators that will take part in the implemented knowledge models, an alternative description of the multitemporal classification model proposed in Chapter 3 will be presented, in order to ease the description of those operators.

Let's first consider the formulation of the forward model given by Equation (12), where the inputs are the feature vectors calculated for the same geographic region respectively over images ${}^{t+1}\mathbf{I}$ and ${}^{t}\mathbf{I}$. Replacing ${}^{E}\mathbf{C}({}^{t}\mathbf{x})$ and ${}^{L}\mathbf{C}({}^{t+1}\mathbf{x})$ respectively by ${}^{t}\boldsymbol{\alpha}$ and ${}^{t+1}\boldsymbol{\alpha}$; for a given $\mathbf{T}$, ${}^{t+1}\mathbf{w}$ in Equation (12) may be expressed as a function ${}^{F}\mathbf{M}_{\alpha}\,({}^{t+1}\boldsymbol{\alpha}, {}^{t}\boldsymbol{\alpha})$, as below.

$$ {}^{t+1}\mathbf{w} = {}^{F}\mathbf{M}_{\alpha}\,({}^{t+1}\alpha,\, {}^{t}\alpha) = \mathbf{H}\{\, \mathbf{F}[{}^{t+1}\alpha,\, {}^{t}\alpha \circ \mathbf{T}]\,\} \tag{35} $$

The same reasoning can be used for the backward (${}^{B}\mathbf{M}_{\alpha}$) and for the general classification function ($\mathbf{M}_{\alpha}$), respectively given by Equation (36) and Equation (37).

$$ {}^{t}\mathbf{w} = {}^{B}\mathbf{M}_{\alpha}\,({}^{t+1}\alpha,\, {}^{t}\alpha) = \mathbf{H}\{\, \mathbf{F}[{}^{t+1}\alpha \circ \mathbf{T}^{-1},\, {}^{t}\alpha]\,\} \tag{36} $$

$$ ({}^{t}\mathbf{w}, {}^{t+1}\mathbf{w}) = \mathbf{M}_{\alpha}\,({}^{t+1}\alpha,\, {}^{t}\alpha) \tag{37} $$

Considering the functions ${}^{F}\mathbf{M}_{\alpha}$, ${}^{B}\mathbf{M}_{\alpha}$ and $\mathbf{M}_{\alpha}$ and the two control strategies introduced in Section 5.2.2, namely synchronous and sequential interpretation strategies, and in view of the actual interpretation goal, different knowledge models can be designed, each model requiring a specific temporal operator.

If the objective is the implementation of the synchronous interpretation strategy, a temporal bottom-up operator based on the general multitemporal classification function $\mathbf{M}_{\alpha}$ should be attached to the temporal nodes. The input fuzzy label vectors ${}^{t+1}\alpha$ and ${}^{t}\alpha$ correspond to the confidence values of the hypotheses created by the monotemporal classifier operators – ${}^{E}\mathbf{C}$ or ${}^{L}\mathbf{C}$ – attached to the ω_i nodes, respectively at each of the two temporal branches. The class transition matrix $\mathbf{T}$, defined or estimated prior to the execution of the interpretation is a parameter of the temporal operator.

Alternatively, if the goal is the implementation of the sequential interpretation strategy, temporal top-down operators based respectively on the forward ${}^{F}\mathbf{M}_{\alpha}$ or the backward ${}^{B}\mathbf{M}_{\alpha}$ functions should be attached to the temporal nodes. Such operators implement those functions only partially – they create for the descendent ω_i nodes hypotheses that have as confidence values temporally updated membership values, i.e. ${}^{t+1}\beta = {}^{t}\alpha \circ \mathbf{T}$ or ${}^{t}\beta = {}^{t+1}\alpha \circ \mathbf{T}^{-1}$ (see Section 3.1), with respect to the last processed temporal branch. In reality, only the hypotheses for which the confidence values (${}^{t+n}\beta$) are greater than zero need to be created. A threshold greater than zero could eventually be used in the decision of the hypotheses to be created, since, if the confidence value is very low, the chance of a hypothesis to be discarded later would be high. This could be a way of sparing processing time; however, in the implementation of the temporal top-down operator in this work, such threshold was set to zero. Anyhow, the confidence values (${}^{t+n}\beta$) are stored as a property of the object hypotheses and are made available after the subsequent top-down and bottom-up processing in the respective temporal branch is completed.

Furthermore, a temporal bottom-up operator attached to the temporal nodes must perform the fusion and hardening operations (respectively functions **F** and **H** in equations (35) and (36)) over the stored confidence values (${}^{t+n}\beta$) and those obtained through the complete structural (top-down and bottom-up) processing of the corresponding temporal branch (${}^{t+n}\alpha$), in order to attain the final crisp classification at the corresponding date.

From the temporal top-down and bottom-up operators mentioned in this section different knowledge models can be built: *synchronous multitemporal interpretation knowledge models*, using the temporal bottom-up operator based on $\mathbf{M}_\alpha$; and *sequential multitemporal interpretation knowledge models*, using the temporal operators based on ${}^F\mathbf{M}_\alpha$ and ${}^B\mathbf{M}_\alpha$. Examples of those two types of knowledge models were implemented in the experiments reported in Section 6.2. In order to facilitate later discussion, those operators will be denoted as follows.

a) *synchronous temporal bottom-up operator*

Temporal bottom-up operator that evaluates the hypotheses of the children of all temporal nodes based on the general multitemporal classification function $\mathbf{M}_\alpha$ as described above.

b) *sequential temporal top-down operator*

Temporal top-down operator that creates hypotheses for the children of the temporal nodes, based on the forward (${}^F\mathbf{M}_\alpha$) or backward (${}^B\mathbf{M}_\alpha$) classification functions, assigning to those hypotheses confidence values that are equal to the temporally updated membership values, i.e. ${}^{t+1}\beta$ or ${}^{t}\beta$, as described above.

c) *sequential temporal bottom-up operator*

Temporal bottom-up operator that implements the fusion **F** and hardening **H** operations of the functions (${}^F\mathbf{M}_\alpha$ or ${}^B\mathbf{M}_\alpha$), as described above.

Figure 40 indicates schematically the placement of the temporal operators in the multitemporal land-cover classification knowledge models designed for the validation experiments.

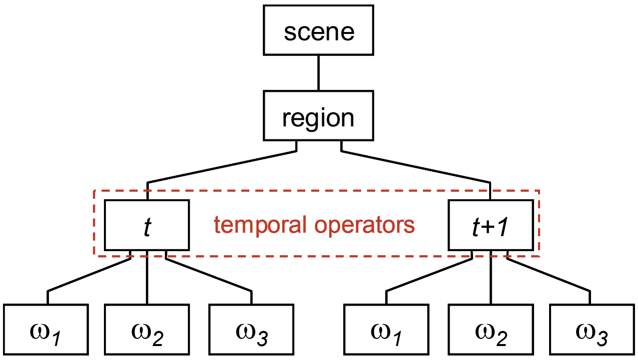

Figure 40. Placement of the temporal operators in the land-cover classification knowledge models.

6.1.3.
Implementation of the operators

In this section the main aspects of the implementation of the operators (structural and temporal, top-down and bottom-up) used in the experiments are revealed.

The structural multiclass top-down operators denoted as *monotemporal structural top-down operators* ${}^{E}\mathbf{C}$ and ${}^{L}\mathbf{C}$ in Section 6.1.1, were coded into a single parametric operator. Part of the operator was implemented through routines coded in the MATLAB software (Mathworks, 2009). The outcome of the MATLAB routine, namely the segmentation and the mean spectral values of the generated segments, were imported during the interpretation process by a program written in C++, which was called by the interpretation control process when it reached the semantic network nodes associated to the land-cover classes.

Instead of implementing new temporal bottom-up operators from scratch, the routines necessary for temporal processing were coded into the *generic bottom-up operator*. The generic bottom-up operator is a program written in C++ that has many embedded functions to process and evaluate the input region hypothesis. Its functions where designed to help the user build different decision rules for the rejection/validation of the hypotheses generated by the top-down operators. Such operator accepts as input a set of commands that can be defined through the system's GUI in a reverse polish notation (RPN) based language[15]. The temporal processing routines can then be evoked through commands of this proprietary language, and used in combination with the previously existing ones. The most important routines implemented in this work were:

a) *multitemporalMerge*

This routine takes as input a list of hypothesis from the children of the temporal nodes and a class transition matrix. It re-segments the original regions associated to the input hypotheses, partitioning them in such a way that each spatial overlap among the original regions will become a partial region of the original hypotheses. The spatially coincident partial regions inherit the confidence values of the original hypotheses, which are regarded as the memberships ${}^{t+1}\alpha_i$ and ${}^{t}\alpha_i$ in the decision rule described in Section 3.4 (Figure 11). All but the winner partial region hypotheses for each time are then erased from the original input hypotheses. This can be clarified with the aid of Figure 41, assuming that the regions of different colors are the original regions associated to hypotheses of land-cover classes ω_0, ω_1 at times t and $t+1$ (denoted as ${}^{t}\omega_0$, ${}^{t}\omega_1$, ${}^{t+1}\omega_0$ and ${}^{t+1}\omega_1$). All numbered sub regions (1 to 11) will become partial regions of the original hypotheses.

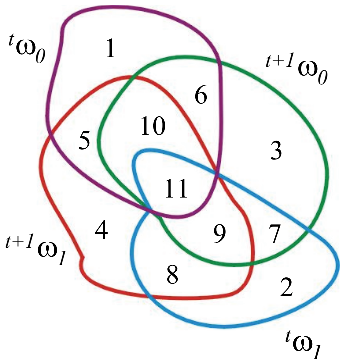

Figure 41. Overlapping regions of hypotheses from different times.

One possible outcome of the *multitemoralMerge*, depending on the hypotheses' confidence values and the transition matrix possibility values, can be that shown in Figure 42.

[15] Currently a visual language interface is under development. The user can define through the visual language rules that will be internally translated into the RPN language.

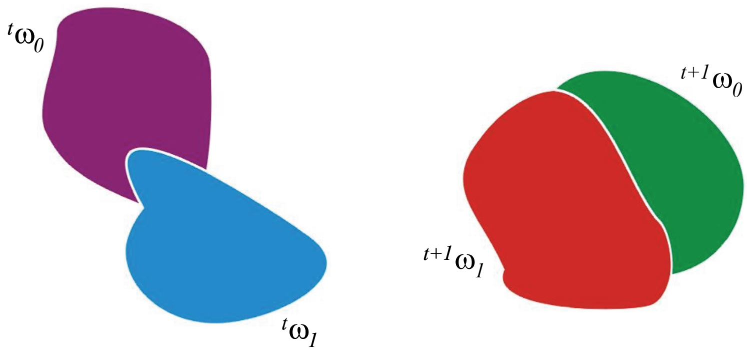

Figure 42. Possible outcome of the multitemporalMerge routine.

After the resolution of the spatial conflicts among the hypotheses of the children of the temporal nodes, this routine updates the temporal nodes' hypotheses, so that their corresponding regions will be the union of the respective children's regions (see Section 5.1.3). This routine was used in the implementation of the *synchronous temporal bottom-up operator* mentioned in the last section. Note that in the experiments reported in this chapter, no re-segmentation was needed, since the segments from the different times coincide spatially, nevertheless, the regions associated to the different hypotheses from a temporal branch do overlap spatially and this routine was used to select the winner hypotheses of each branch.

b) *multiDmerge*

This routine performs a multi-dimensional merge. It takes as input a list of hypothesis and the name of an attribute common to those hypotheses. It then separates the hypotheses which have the same values for that attribute into different sets and resolves the spatial conflicts among those hypotheses solely based on the hypothesis confidence values in such a way that the overlapping region will be erased from the loser hypotheses. This routine was used in the implementation of the *sequential temporal bottom-up operator* introduced in the last section.

The temporal top-down operator used in the *sequential multitemporal interpretation knowledge model* was also coded within a previously existing operator, called *top-down symbolize operator*. This operator is similar to the generic bottom-up operator; in fact it uses the same C++ classes and also takes as input a set of commands in the same RPN-based language. The main difference is that instead of delivering a set of *verified* hypotheses (or instances) its output is a set of *new* region hypothesis.

A routine called *projectMultitemporal* was specially implemented to be used in such operator. It takes as input a list of previously validated hypothesis (instances) associated to the children of one temporal node and creates from each of those instances hypothesis for the children of the next temporal node (considering the nodes' temporal order, see Section 5.2.3).

6.2. Experimental results

Two sets of experiments were designed to show how the multitemporal classification method proposed in Chapter 3 can be implemented on the framework. The experiments' results and the corresponding knowledge models will be presented in the next sections.

It is important to note that these experiments were devised to investigate if, given the same conditions, with the same transition matrixes and monotemporal classifiers trained in the same fashion, the outcome of the method structured in the framework is similar to that obtained in the environment it was originally implemented. Moreover, the same images used in the experiments reported in Chapter 4 were interpreted in the experiments and, in order to produce comparable results, the same segmentation and monotemporal classification techniques, described respectively in sections 4.1.1 and 4.2, were used in the implementation of the structural top-down operators used in the experiments.

6.2.1. Sequential interpretation experiments

In this sequence of experiments, the classification of the reference segments over the 1999 image using the sequential interpretation strategy (Section 5.5.2) was performed.

In these experiments the training set was built exactly as in Section 4.5.1. The reference image segments were first separated into groups according to the class transitions they undergone between 1999 and 2000. To estimate the parameters of the structural monotemporal operators (the mean spectral values and covariance matrixes corresponding to each of the ω_i land cover classes, in each year), as well as the transition possibility matrix (a parameter of the temporal operators), around 50% of the segments in each group were randomly selected to form the training set. The remaining segments were used to evaluate the interpretation outcome. The transition matrixes used in this set of experiments were estimated using the GA-based estimation technique (Section 3.5), using average class as the accuracy metric.

The computation of the operators' parameter values was done previous to the run of the interpretation process in the framework.

The semantic network used in these experiments, as depicted by the framework's graphical user interface, is shown in Figure 43.

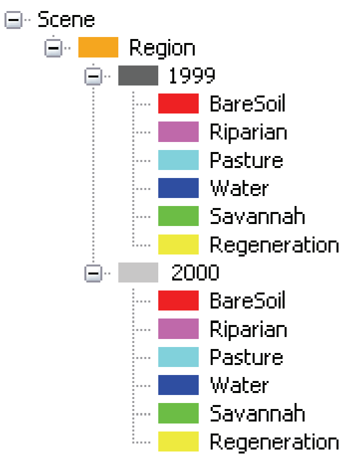

Figure 43. Semantic network for sequential classification of the image from 1999.

The temporal order of the *2000* temporal node in the semantic network was set to 1 and the temporal order of the *1999* temporal node, to 2, indicating that the 2000 temporal branch would be classified first, in order to create the initial hypotheses for the children of the temporal node *1999*. The operators attached to the nodes of the network are indicated bellow.

Structural operators:

a) *structural dummy top-down operator* at nodes *Scene* and *Region.*

b) *monotemporal structural top-down operator* at the children of temporal node *1999* – performs the segmentation and monotemporal classification of the 1999 image.

c) *monotemporal structural top-down operator* at the children of temporal node *2000* – performs the segmentation[16] and monotemporal classification of the 2000 image.

Temporal operators:

a) *sequential temporal top-down operator* at temporal node *2000* – creates the initial hypotheses of the child nodes of the *1999* temporal node.

b) *sequential temporal bottom-up operator* at temporal nodes *1999* and *2000* – evaluates the hypotheses of the child nodes of the *1999* temporal node.

Two sets of experiments were carried out, in each of them, 5 interpretations were performed, each time with a different random selection of training and testing reference segments. In the fist set, the correct classification of the reference segments in 2000 was not used; in the second one such information was used.

In the first set of experiments the average of the attained class accuracies was 64.4%. Table 18 shows the confusion matrix generated (over the testing set of segments) from the outcome of the best experiment, which reached an average class accuracy of 69.6%. Figure 45 shows a thematic map produced in that experiment. Figure 44 shows the reference classification for 1999.

Classes	**Bare Soil**	**Riparian**	**Pasture**	**Water**	**Savannah**	**Regeneration**	***Omission***
Bare Soil	27	0	8	0	1	0	25.0
Riparian	0	8	1	0	0	2	27.3
Pasture	62	2	63	0	0	1	50.8
Water	0	0	0	7	0	0	0.0
Savannah	0	4	0	0	24	6	29.4
Regeneration	0	0	0	0	1	1	50.0
Commission	69.7	42.9	12.5	0.0	7.7	90.0	
Accuracy	75.0	72.7	49.2	100.0	70.6	50.0	

Table 18. Confusion matrix for the sequential classification of the image from 1999 – classification from 2000 not known.

[16] As the two segmentations would actually produce the same results, a special feature of the

Classes	Bare Soil	Riparian	Pasture	Water	Savannah	Regeneration	*Omission*
Bare Soil	31	0	5	0	0	0	13.9
Riparian	0	11	0	0	0	0	0.0
Pasture	10	0	118	0	0	0	7.8
Water	0	0	0	7	0	0	0.0
Savannah	0	0	0	0	34	0	0.0
Regeneration	0	0	0	0	0	2	0.0
Commission	24.4	0.0	4.1	0.0	0.0	0.0	
Accuracy	86.1	100.0	92.2	100.0	100.0	100.0	

Table 19. Confusion matrix for the sequential classification of the image from 1999 – classification from 2000 known.

In the second set of experiments, the reference classification from 2000 was used for the interpretation of the 1999 image. The average of the attained class accuracies was 94.6%. Table 19 shows the confusion matrix created from the results of the best experiment in this set, which reached an average class accuracy of 96.4%. Figure 45 shows the thematic map created in the experiment.

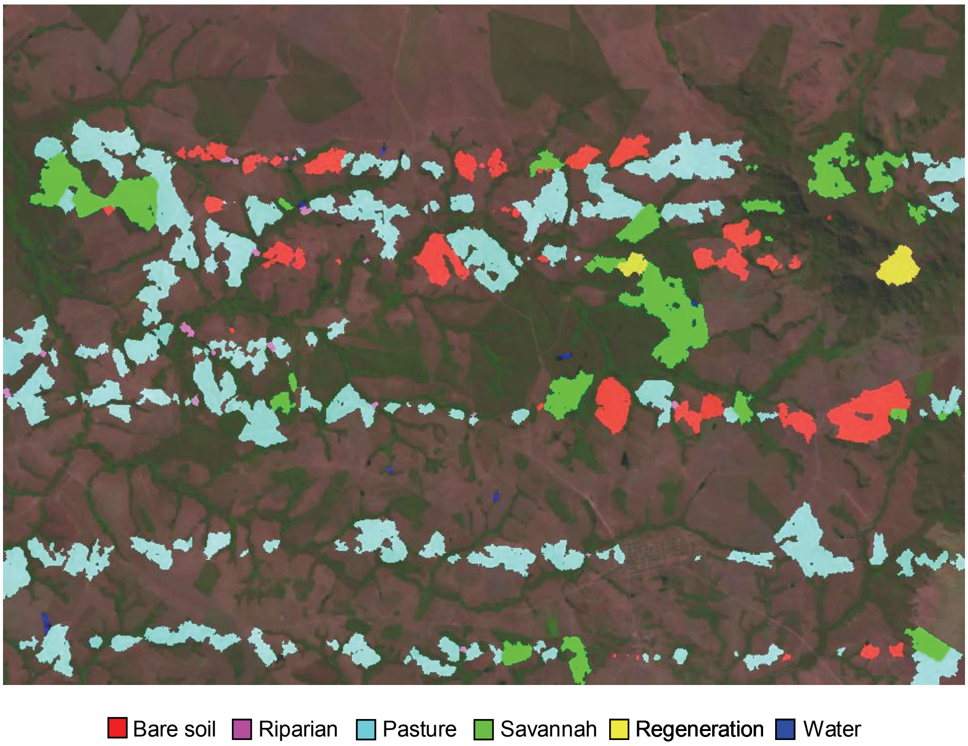

Figure 44. Reference classification for the 1999 image.

framework was used so that the segments were generated only once.

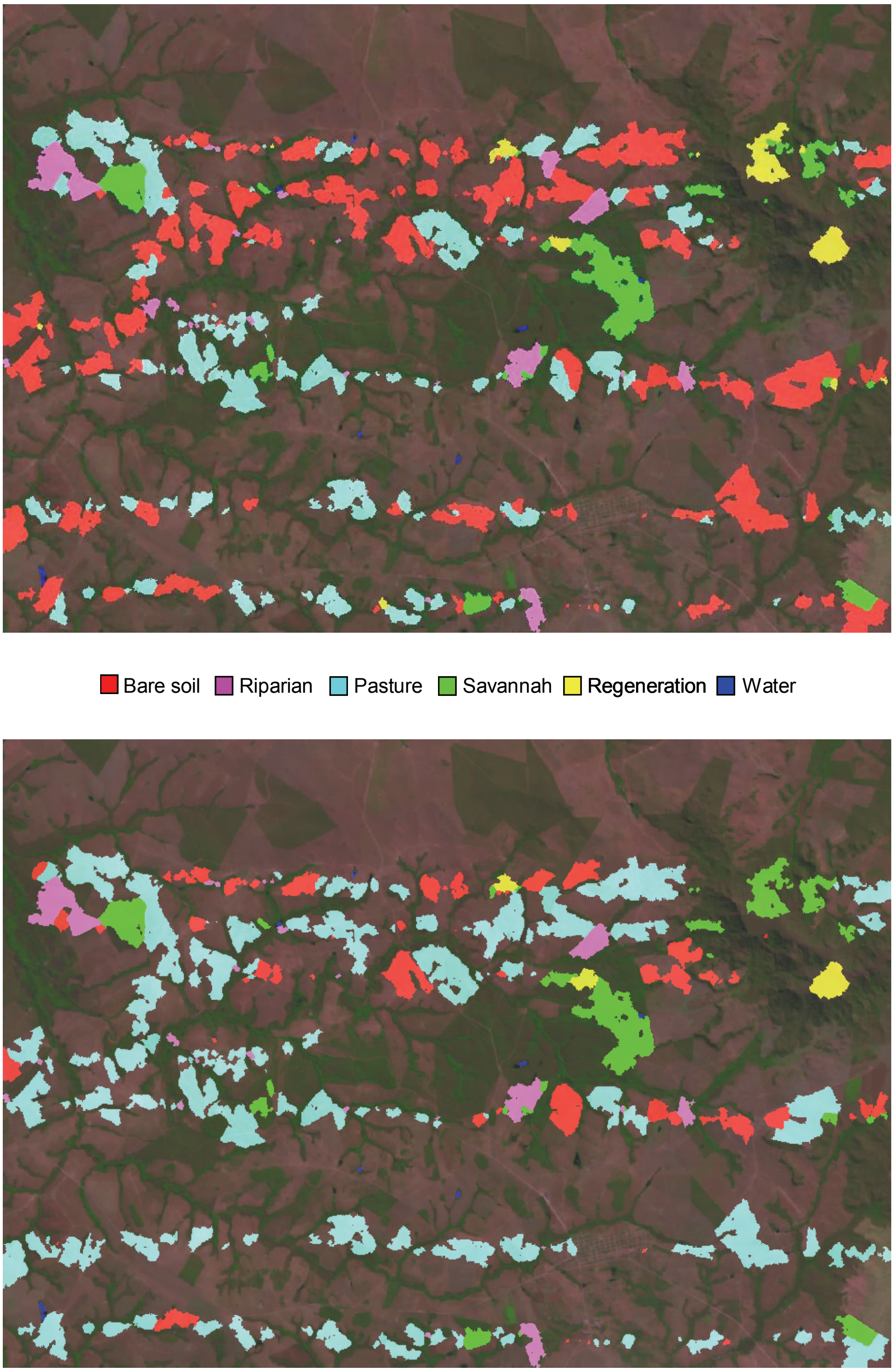

Figure 45. Classification of the reference segments over the 1999 image: top, sequential classification – true classification for 2000 not known; bottom, sequential classification – true classification for 2000 known (average class accuracies: 69.6 % and 96.4%, respectively).

6.2.2. Synchronous interpretation experiments

In this set of experiments, the classification of the reference segments over the images from 2000 and 2001, using the synchronous interpretation strategy (Section 5.5.2) was performed.

The training set was built as reported in the previous section. The reference segments were separated into groups according to their class transitions between 2000 and 2001. To estimate the parameters of the structural monotemporal operators, as well as the transition possibility matrixes, approximately 50% of the segments in each group were randomly selected to form the training set. The remaining segments were used to evaluate the interpretation outcome. Again, the computation of the operators' parameter values was done previous to the run of the interpretation process in the framework.

The semantic network used in these experiments is shown in Figure 46.

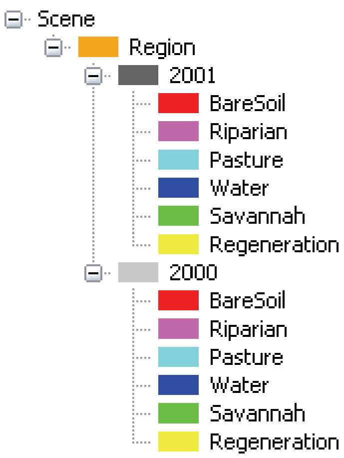

Figure 46. Semantic network for synchronous classification of the images from 2000 and 2001.

The temporal order of both temporal nodes (*2000* and *2001*) was set to 1, indicating that the interpretation strategy was synchronous. The operators attached to the nodes of the network are indicated bellow.

Structural operators:

a) *structural dummy top-down* at nodes *Scene* and *Region.*

b) *monotemporal structural top-down operator* at the children of temporal node *2000* – performs the segmentation and monotemporal classification of the 2000 image.

c) *monotemporal structural top-down operator* at the children of temporal node *2001* – performs the segmentation and monotemporal classification of the 2001 image.

Temporal operator:

a) s*ynchronous temporal bottom-up operator* at temporal nodes *2000* and *2001* – evaluates the hypotheses of the child nodes of the *2000* and *2001* temporal nodes.

Two sets of experiments were carried out. In the first one, information about the correct classification of the reference segments from 2000 and 2001 was not used, in the second set of experiments, the information about the correct classification of the segments in 2000 was used for the interpretation of the 2001 image. In each set of experiments 5 interpretations were executed, each time with a different random selection of training and testing sets.

In the first set of experiments, transition matrixes were estimated using the least-squares estimation technique; and in the second, using the GA-based one (Section 3.5). Average class was the accuracy metric for both estimation techniques.

In the first set of experiments the average class accuracy attained for the 2000 image was 66.8%, and for the 2001 image, 67.2%. In the best experiment from this set, the accuracy attained by the automatic interpretation of the 2000 image was of 75.6%, and for the 2001 image, of 73.4%.

Tables 20 and 21 show the confusion matrixes generated from the outcome of the best experiment. Figure 47 shows a thematic map produced in the experiment for the 2000 image. In Figure 48, the thematic map produced for the 2001 image is showed.

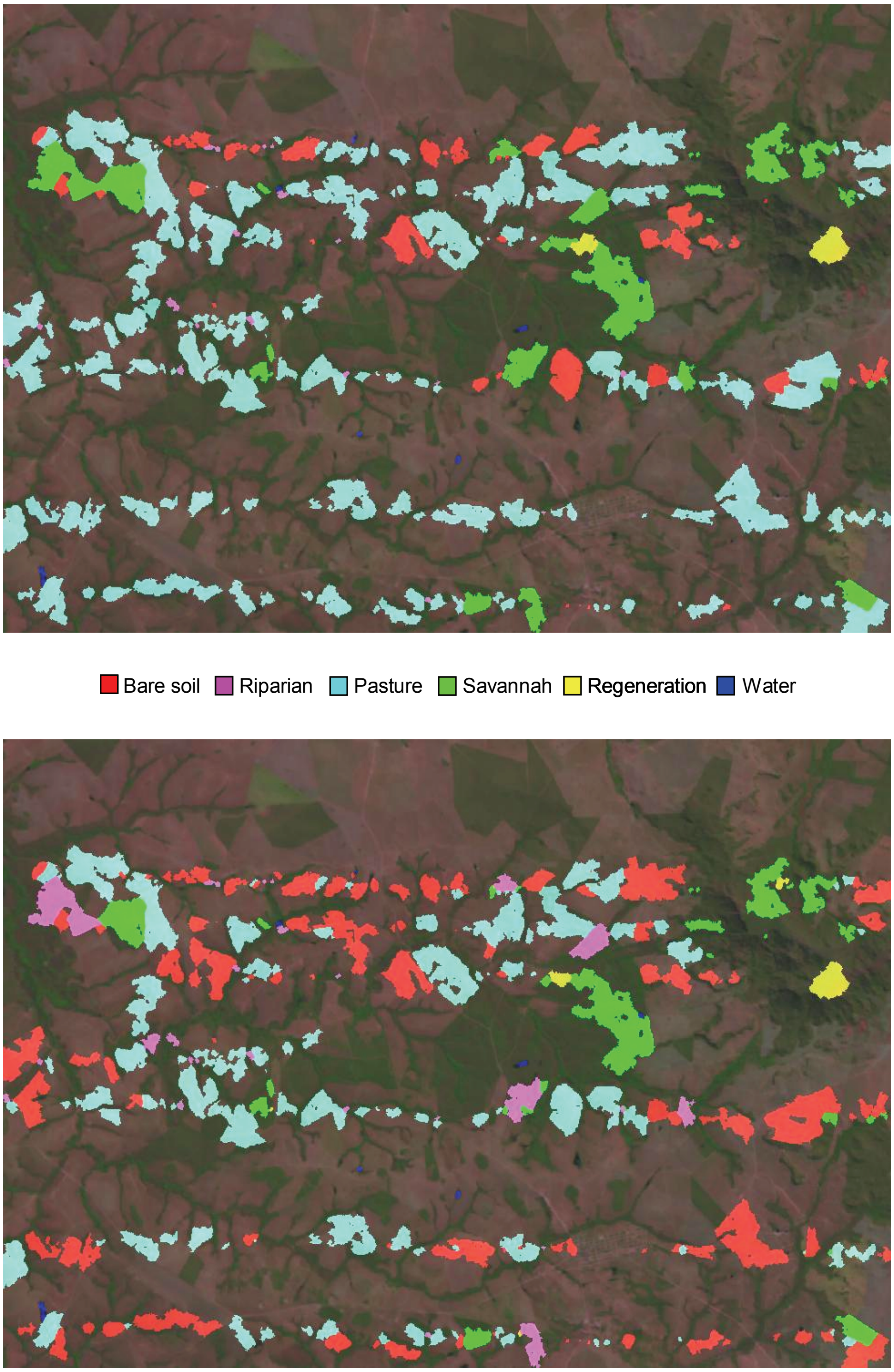

Figure 47. Classification of the reference segments over the 2000 image: top, reference classification; bottom, synchronous classification – true classification for 2001 not known (average class accuracy of 75.6%).

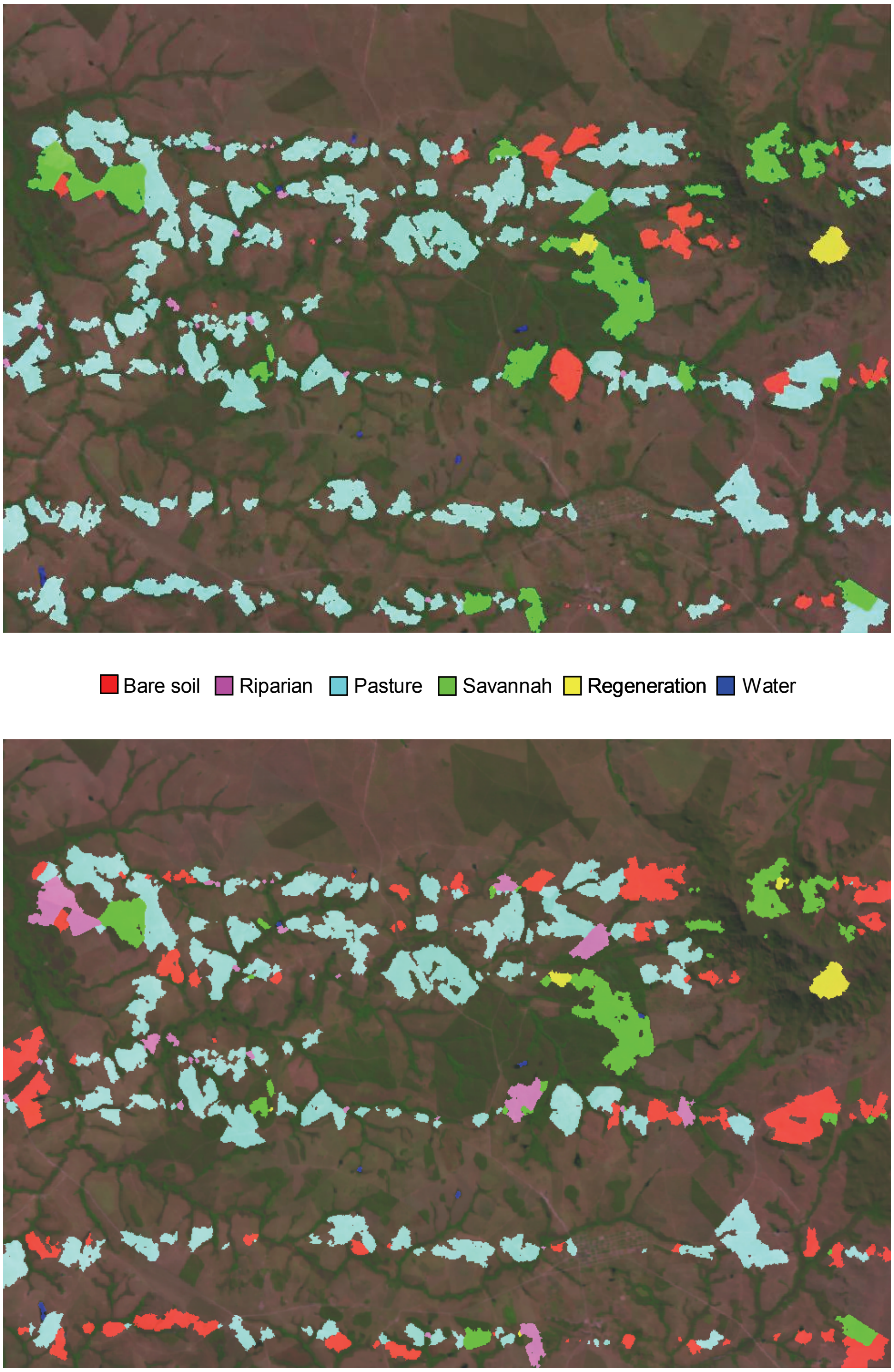

Figure 48. Classification of the reference segments over the 2001 image: top, reference classification; bottom, synchronous classification – true classification for 2000 not known (average class accuracy of 73.4%).

Classes	Bare Soil	Riparian	Pasture	Water	Savannah	Regeneration	*Omission*
Bare Soil	25	0	6	0	0	0	19.4
Riparian	0	11	0	0	0	0	0.0
Pasture	51	1	81	0	0	0	39.1
Water	1	0	0	6	0	0	14.3
Savannah	0	4	0	0	26	4	23.5
Regeneration	0	0	0	0	1	1	50.0
Commission	67.5	31.3	6.9	0.0	3.7	80.0	
Accuracy	80.6	100.0	60.9	85.7	76.5	50.0	

Table 20. Confusion matrix for the synchronous classification of the image from 2000 – classification from 2001 not known.

Classes	Bare Soil	Riparian	Pasture	Water	Savannah	Regeneration	*Omission*
Bare Soil	12	0	9	0	0	0	42.9
Riparian	0	11	0	0	0	0	0.0
Pasture	39	1	102	0	1	0	28.7
Water	1	0	0	6	0	0	14.3
Savannah	0	4	0	0	26	4	23.5
Regeneration	0	0	0	0	1	1	50.0
Commission	76.9	31.3	8.1	0.0	7.1	80.0	
Accuracy	57.1	100.0	71.3	85.7	76.5	50.0	

Table 21. Confusion matrix for the synchronous classification of the image from 2001 – classification from 2000 not known.

In the second set of experiments, the correct classification of the reference segments in 2000 was used for the interpretation of the 2001 image. Once again, 5 interpretations were performed, each time with a different random selection of training and testing sets to train both the monotemporal classifiers and to estimate the transition matrixes. Transition matrixes were estimated using the GA-based estimation technique (Section 3.5), with average class was the accuracy metric.

The average accuracy achieved in this set of experiments, for the 2001 testing segment set was 93.6%. In the best experiment, the accuracy attained was of 96.1%. The respective confusion matrix is shown in Table 22. The classification of the reference segments from 2001 is shown on the top of Figure 49. On the bottom of the figure, the classification of the whole 2001 image is shown. Note that only the correct classification of the reference segments are known, such information is not available for the other image segments.

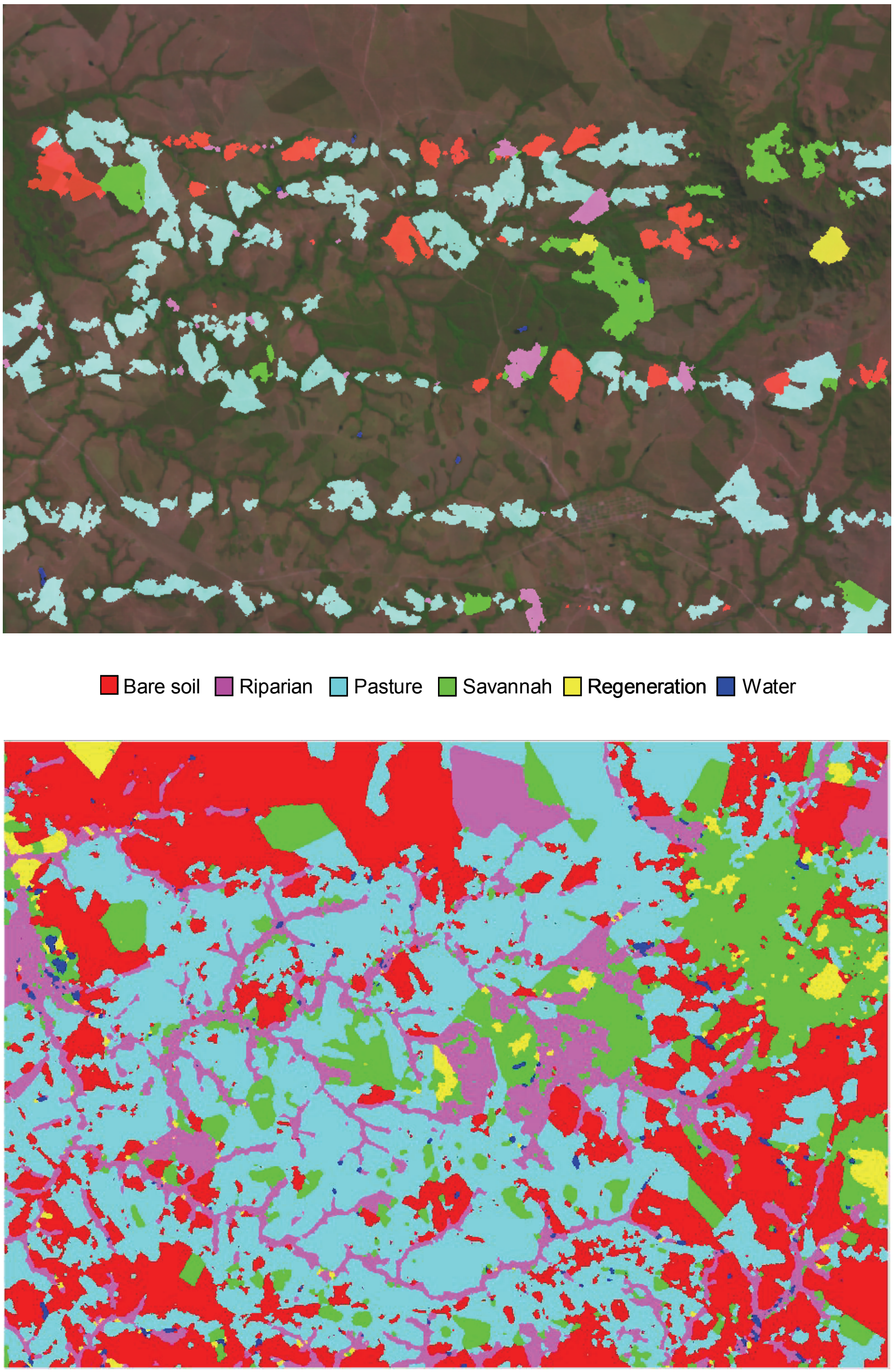

Figure 49. Top, synchronous classification for the reference segments of the 2001 image – true classification for 2000 known (average class accuracy of 96.1%); bottom, complete classification of the 2001 image – true classification of the reference segments in 2000 known.

Classes	**Bare Soil**	**Riparian**	**Pasture**	**Water**	**Savannah**	**Regeneration**	***Omission***
Bare Soil	18	0	3	0	0	0	14.3
Riparian	0	11	0	0	0	0	0.0
Pasture	13	0	130	0	0	0	9.1
Water	0	0	0	7	0	0	0.0
Savannah	0	0	0	0	34	0	0.0
Regeneration	0	0	0	0	0	2	0.0
Commission	41.9	0.0	2.3	0.0	0.0	0.0	
Accuracy	85.7	100.0	90.9	100.0	100.0	100.0	

Table 22. Confusion matrix for the synchronous classification of the image from 2001 – classification from 2000 known.

All results presented in this section are consistent with those reported on Section 4.5.1., what shows that the fuzzy Markov chain method presented in Chapter 3 was successfully implemented through the proposed multitemporal framework. In order to further attest that, experiments were performed using solely the original environment in which the method was implemented – the MATLAB software – using the same transition and covariance matrixes and mean vectors, as the ones used in the experiments that attained the best performances. The results of those experiments were exactly the same as the ones presented in this section.