2
Previous works

This chapter presents the most relevant works in the two particular fields this research has concentrated on – knowledge based image interpretation systems and multitemporal cascade-classification methods – focusing on those initiatives applied to the interpretation of remote sensing image data. It must be noted that the following text is not intended to be an extensive survey; instead it aims at offering an overview of the key aspects of the most important systems and methods related in some way to what is proposed in this Thesis.

2.1.
Knowledge-based image interpretation

The goal of knowledge based automated image interpretation consists in the generation of symbolic descriptions of images, automatically identifying its contents to the extent of what is relevant to a particular application. The symbolic descriptions can be used for different purposes: as inputs of decision making processes; in the search of images with a specific content; in image compression; and so on.

According to Shapiro and Stockman (2001), the most significant results in the area have been restricted to specific domains, for which the properties of the objects expected to be found in the image are easily observable.

Part of the problem has to do with the computational power currently available, especially as compared to the ultimate benchmark of this field: the human vision system (Crevier and Lapage, 1997). Automatic image interpretation systems are slow, process images in relatively restrict range of resolutions and produce incomplete results, while our visual system operates in real-time, process images of variable resolutions (from very high to very low) and produce excellent results for problems of diverse natures (of unrestricted scope).

Another problem relates to the processes performed by our brains when interpreting images. As we start the strive for making sense of what we see, as

PUC-Rio - Certificação Digital Nº 0510501/CA

soon as we first open our eyes outside our mothers' wombs, long before any language is learned, those mental processes are carried out almost completely bellow our consciousness threshold. That's why image interpreters are not capable of mapping seamlessly the processes that support their observations into rules or other forms of knowledge representation, as experts in some other domains do (Forsyth and Ponce, 2003).

This difficulty in mapping the human visual interpretation mechanisms, however, reinforces the need of making explicit the current knowledge about image interpretation problems. There are many advantages in structuring and organizing knowledge of image interpretation in an explicit form. When explicitly represented, such knowledge can be studied, questioned and validated by experts, favoring interactive approaches for the solution of the problem and providing subsidies for the exploration and comparison of alternative interpretation strategies.

Another advantage of making use of explicitly represented knowledge has to do with the possibility of integrating, at the same time differentiating knowledge associated to different domains: generic knowledge about image processing; about the software tools used; about the particular characteristics of the images and of the objects of interest; and knowledge about the physical world (Crevier and Lapage, 1997).

Knowledge based systems enhance, through providing support, that which a human agent does (Graham and Jones, 1997). The computer support usually consists of some representation of the problem solving knowledge, an inference mechanism and some uncertainty handling features. Such systems store specific knowledge about a particular application – generally regarded as a knowledge model – in an explicit way, and process this knowledge through a particular problem solving strategy.

Every image classifier incorporates some form of knowledge representation, but by lacking an explicit high-level knowledge representation such classifiers are seldom labeled as knowledge-based (Mota et al., 2007).

The knowledge representation structures and the interpretation process control logics are the key aspects that differentiate the knowledge-based image interpretation systems devised thus far. In the following sections the most relevant of such systems will be briefly described and classified according basically to

PUC-Rio - Certificação Digital Nº 0510501/CA

their knowledge representation structures. Before that, a definition of what is understood as knowledge in this work is given.

2.1.1. Definition of knowledge

In artificial intelligence terms, *knowledge* stands for *facts* and *heuristics*. Facts constitute the body of information available and heuristics are rules of good judgment, of plausible reasoning, that characterize expert-level decision making (Harmon and King, 1985).

A number of different classifications of knowledge have been proposed, some of which are particularly relevant in the context of this research. The first one differentiates between *implicit knowledge* and *explicit knowledge* (Gottlob et al., 1990; Pahl, 2003). Implicit knowledge is stated directly in a computer program code and describes a rigid procedural plan. It is expressed by an algorithm which is not intended to support alternative problem solving methods, but to perform a specific set of actions known to be useful at a particular time, for a particular problem, and for input data with particular characteristics. The problem solving knowledge is encoded in the program, and cannot be easily read or extracted from it.

Explicit knowledge is stored independently from the system components responsible for its processing. It is represented through a formal language, with a well defined syntax and semantics, and can be easily read, edited or updated.

Explicit knowledge can be further discriminated into *declarative knowledge* and *procedural knowledge*. These concepts are intimately related to what has been denoted by facts and heuristics. Declarative knowledge describes the characteristics of the relevant concepts – which represent the real-word entities involved in the problem to be solved – as well as the relationships among those entities. Procedural knowledge describes instead processing steps or inference rules. Sagerer and Niemann (1997) add to those types of explicit knowledge a third one – *a posteriori knowledge* – which represents the results of the problem solving process, a solution according to the system (that might not be the correct one).

PUC-Rio - Certificação Digital Nº 0510501/CA

In (Mota et al., 2007) an alternative classification for explicit knowledge is proposed in the context of remote sensing image interpretation: *spectral knowledge*, *spatial knowledge* and *temporal knowledge*. Spectral knowledge is related to the spectral characteristics of the individual objects, it is local knowledge in the sense that it describes an object by itself.

Spatial knowledge is used to describe the spatial or geographical contexts in which the objects can occur; this may include a description of their parts as well as their spatial relations to other objects. Both special and spectral knowledge refer to object features on an image, and not take the history of the object into account.

Temporal knowledge describes the temporal dynamics of the classes of objects in a specific geographical area. It refers particularly to the knowledge about possible class changes of the recognizable visual objects present in a set of multitemporal image data. In this work, temporal knowledge stands for the possibility of an object belong to a particular class in one point in time, e.g. agricultural field, and to any other class, e.g. urban area or forest, in another time (past or future).

Knowledge representation addresses the problem of capturing in a formal language the knowledge required for solving a particular problem. Many different representation forms of explicit knowledge have been used in image interpretation systems, such as attribute tables (Haralick and Shapiro, 1999), production rules (Clement et al., 1993; McKeown et al., 1985; Nieman et al., 1990), frames (Brooks, 1983; Matsuyama and Hwang, 1990), semantic networks (Hanson and Riseman, 1988; Liedtke et al., 1997; Liedtke et al., 2001; Nieman et al., 1990; Schiewe, 2001), and agents (Crevier and Lepage, 1997; Draper et al., 2000; Draper et al., 2004). In the field of knowledge-based image interpretation, semantic networks, frames and rules have been the most often used knowledge representation.

PUC-Rio - Certificação Digital Nº 0510501/CA

2.1.2.
Semantic network-based systems

A semantic network is basically a directed graph, where the nodes represent concepts, and the arcs represent binary relationships between concepts. It is an example of a declarative knowledge representation form.

Actually, semantic nets can be used both for expressing what is expected to be found in an image, and for representing the objects found (instantiated) on the process of interpreting the image (a posteriori knowledge). Therefore, according to the particular kind of knowledge represented, they can be regarded either as a *conceptual network* – which represents knowledge of the characteristics of classes of objects and the interrelationship among classes – or as an *instance network* – which identifies and describes the properties of the actual objects found in the image. There is an implicit relationship among the nodes of both types of networks, an instance node, which is associated to one image object, is always related to the conceptual node that represents the class of the object.

It is interesting to note that the image interpretation systems that use semantic networks usually consider alternative hypotheses for the same visual object, i.e. for the one particular geographical region. This means that the instance network will change during the interpretation process: with the addition of new nodes, when a new object hypothesis is found; with the modification of the information stored in its nodes; or the deletion of nodes when the hypotheses they represent are discarded.

In most knowledge-based image interpretation systems, the conceptual network does not change during the execution of the interpretation process. A notable exception is ERNEST (Niemann et al., 1990), a system used in speech recognition and image analysis, in which nodes of a special type called *modified concept* can be added by the control process to the concept network. During the interpretation process, after instances of the subclasses of a particular concept have been found, the system can automatically define additional restrictions (over the range of attributes) for the respective superclass, creating a new modified concept node, hence refining the knowledge model through information produced during the interpretation process. Modified concept nodes are, however, not

PUC-Rio - Certificação Digital Nº 0510501/CA

included definitely in the conceptual network, as they are a function of the particular inputs of an interpretation problem.

The links of the semantic network state both hierarchical and other types of relationships among concepts, and usually hold an explicitly defined semantics. They typically represent specialization (*is-a*) or parthood (*part-of*) relationships, but many other types of relations can be expressed. For instance, in ERNEST and AIDA (Liedtke et al., 1997), a relation type called *concrete-of* is also defined. It relates concepts of different abstraction levels, indicating the primitive geometric class associated to a specific abstract class.

Figure 1 shows a semantic network in the ERNEST system. It should be noted that the depicted network links are named and their specific semantics are identified: *c – concrete-of*; *sp – specialization*; e *p – part-of*.

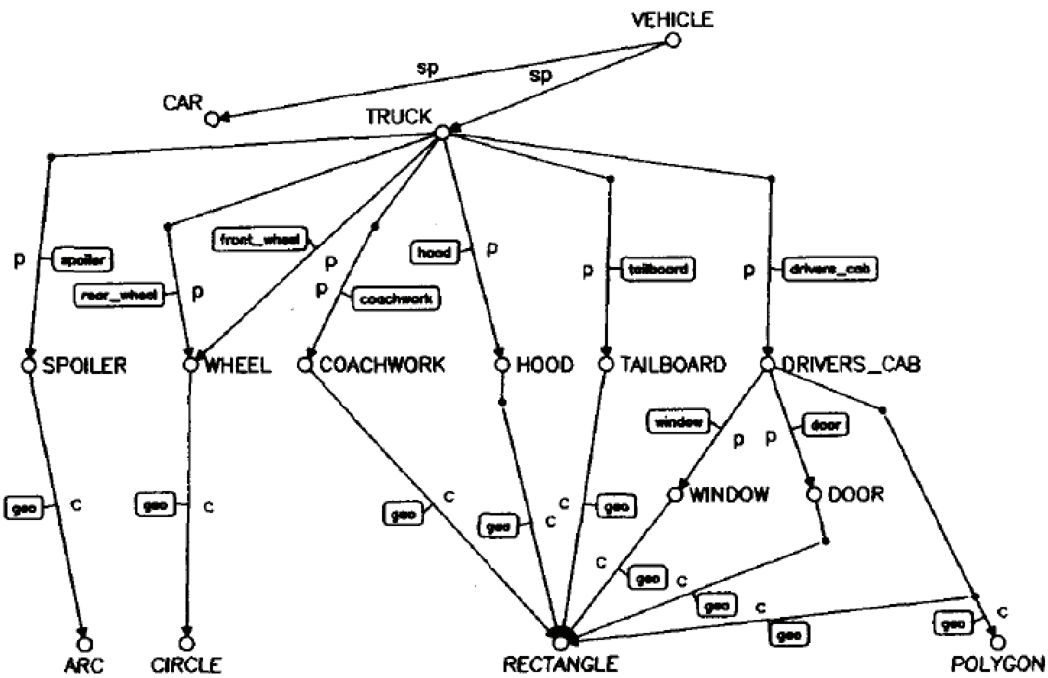

Figure 1. Semantic network in the ERNEST system (Niemann et al., 1990).

The semantic network presented in Figure 2 was designed for an interpretation project in the AIDA system. It has links of types *part-of* (untagged solid links), *concrete-of* (dashed links), *is-a* and *close-to*, this last type expressing distance relations among concepts. Another interesting characteristic of AIDA's semantic networks is the presence of explicitly defined abstraction layers (*scene*, *GIS*, *material*, *geometry* e *sensor*). Note that links between classes in different

PUC-Rio - Certificação Digital Nº 0510501/CA

abstraction layers are of type *concrete-of*. Also explicitly defined through the links of the network are cardinality constraints associated to parthood relations.

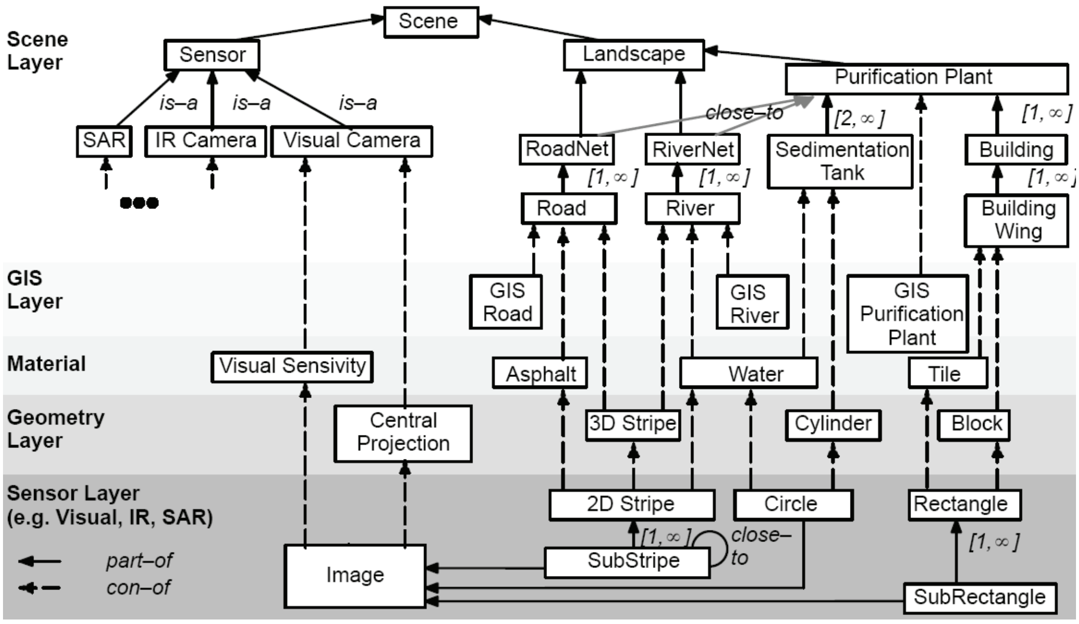

Figure 2. Semantic network in the AIDA system (Liedtke et al., 1997).

Figure 3 shows a semantic network defined for an interpretation project in the GeoAIDA system (Pahl, 2003). In this system the links express only hierarchical relations between concepts – the semantics of the links are defined through specific attributes associated to the nodes of the network. Among those attributes are procedures related both to image processing operations and to the judgment of object hypotheses.

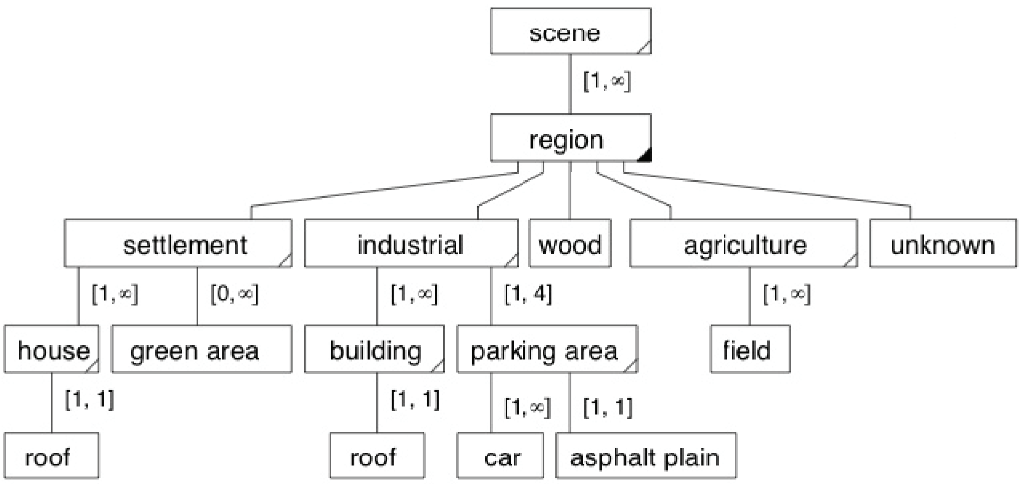

Figure 3. Semantic network in the GEOAIDA system (Liedtke et al., 2001).

PUC-Rio - Certificação Digital Nº 0510501/CA

GeoAIDA introduces a concept denoted as *holistic operator*, whose function is to create instances of a class before the concepts associated to the descendent nodes are instantiated. Accordingly, all leaf nodes must be attached to a holistic operator. For the judgment of image object hypotheses GeoAIDA relies on procedures, which are executed by a specialized operator, defined at the nodes of the conceptual semantic network. Such procedures are stated in a high-level programming language that provides a number of functions for selecting, validating or discarding object hypotheses, and for resolving eventual spatial conflicts among them. A procedure acts upon the hypotheses associated to the child nodes of the concept node for which it was defined. Figure 4 shows an example of a judgment procedure in GeoAIDA.

```
nodelist
'Grassland' selectClass
'grassland_membership_function' node.push
'brightness' node.get
node.fuzzyp
'p' node.set
drop
nodelist
'Forest' selectClass
'forest_membership_function' node.push
'brightness' node.get
node.fuzzyp
'1.0 -' node.run
'p' node.set
drop
nodelist
'NDVI' node.get
'dup 0.27 > swap 1.0 < *' node.run
select
merge
```

Figure 4. A judgment procedure in GeoAIDA.

Although it can be categorized as *procedural system* (see Section 2.1.5), the Definiens Developer image analysis software package (Definiens, 2007), also provides the means for declaring knowledge through semantic networks. Classes of objects of interest for a particular application can be organized in a so-called *class hierarchy*, which is associated to two distinct networks: the *inheritance network*, which defines the inheritance relationships among the classes; and the *groups network*, which defines their semantic grouping. Figure 5 shows an example of the two networks associated to the class hierarchy of an interpretation application in Definiens Developer.

PUC-Rio - Certificação Digital Nº 0510501/CA

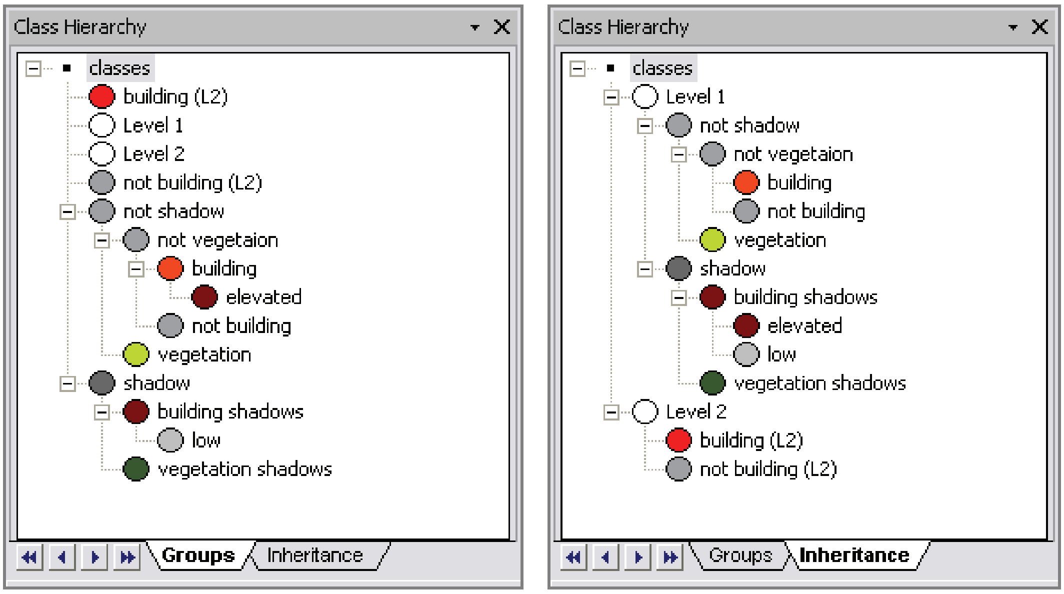

Figure 5. A class hierarchy in the Definiens Developer software: inheritance network (left) and groups network (right).

Descriptions based on membership functions can be created for each class. A number of membership functions – which are defined over the image segments' spectral, morphological or topological attributes – can be combined in a class description through fuzzy logics *t-norms* or *s-norm* operators (Zadeh, 1978), providing a rule for calculating the membership of an image segment to the respective class.

Through the inheritance relationships, derived from the inheritance network, the class description associated to a node is inherited by its descendents and combined to the particular membership functions defined at those nodes. Classification of the image segments based on such class descriptions is implemented by a process called *hierarchical classification*. Figure 6 shows an example of a class description in Definiens Developer and of a membership function that takes part in it.

Beside ERNEST, AIDA, GeoAIDA and Definiens Developer, the knowledge-based image interpretation systems MOSES (Quint, 1997) and VISIONS (Hanson and Riseman, 1988) are examples of systems that use semantic networks for knowledge representation.

An extensive discussion on the application of semantic networks for image interpretation can be found in (Sagerer and Niemann, 1997).

PUC-Rio - Certificação Digital Nº 0510501/CA

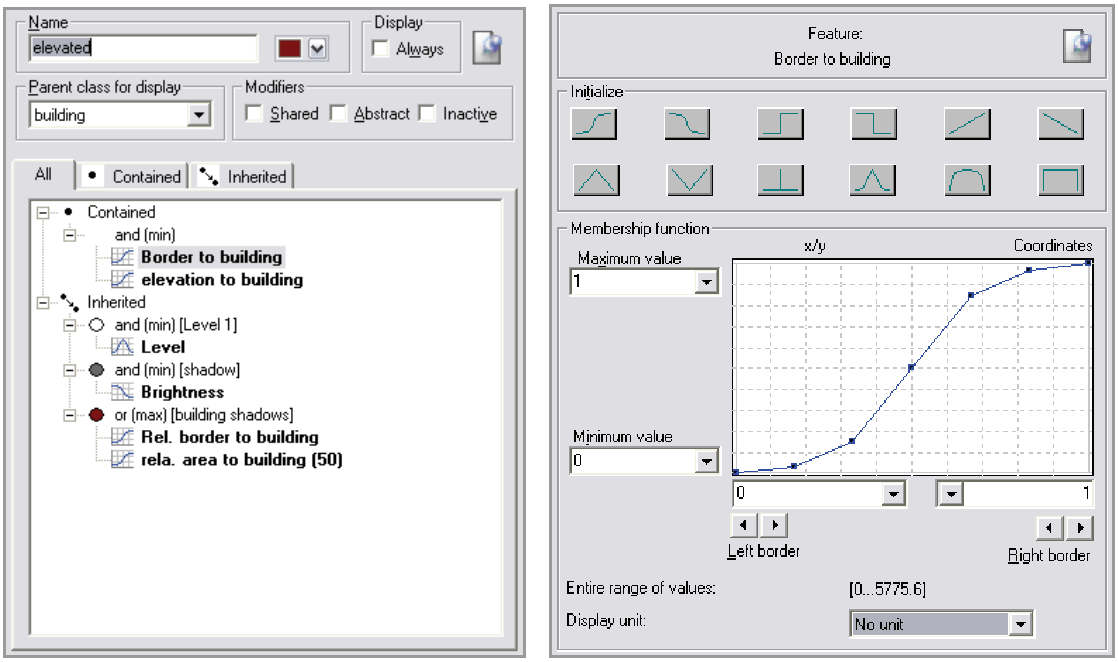

Figure 6. A class description in the Definiens Developer software (left) and of a membership function (right).

2.1.3. Systems based on frames

Frames were introduced by Minsky (1975) as a data structure for representing a stereotyped situation. He envisioned a relational structure whose terminal nodes consist of *slots* and *fillers*.

Slots can be considered as attributes of an object and fillers as the values of the attributes. A filler can be atomic or can reference another frame, it can be empty and waiting for a value or can contain a default value until the slot is filled. A filler may be a scalar value, a literal expression or a procedure that must be executed to produce a value.

Frames are very flexible structures and represent data models that can be easily mapped into commonly used data structures, such as the ones present in relational databases or in any general purpose programming language. It is also true that many knowledge representation forms may be mapped into frames, including semantic networks.

A *schema* in database terminology is a model or prototype. A database schema describes entities and the relationships among them. In artificial intelligence, the term schema is either used as a synonym for frame or it can have

PUC-Rio - Certificação Digital Nº 0510501/CA

a meaning somewhat similar to its database homonym, that is, schemas can be used to represent a meta-frame, a description of the slots of a frame, including domain restrictions on the respective fillers.

A *schemata* is a data model that represents the relationships of a set of concepts within a domain, it is closely related to the concept of *ontology* in computer science (Gruber, 1993). In the knowledge-based image interpretation systems that use such form of knowledge representation, each image object class is described by a schema. The instances of the object classes are represented by frames associated to the respective class schemas. In such systems, threads of the interpretation process, associated to each schema can be executed in parallel, using *blackboards* as communication mechanisms (Draper et al., 1989).

The ACRONYM (Brooks, 1983), SIGMA (Matsuyama and Hwang, 1990) and MESSIE (Clement et al., 1993) are examples of systems that use schematas and frames as knowledge representation structures.

PUC-Rio - Certificação Digital Nº 0510501/CA

2.1.4. Rule-based systems

Many of the first knowledge based image interpretation systems can be categorized as *production systems* (or production rule systems). The flexible and modular structure that resulted from the use of *production rules* became a trend during the eighties (Crevier and Lapage, 1997).

A production system consists primarily of a set of rules, termed production rules, which describe the behavior or characteristics of objects in a particular domain. Production systems provide the mechanisms necessary to execute production rules in order to achieve some goal for the system, in this context – the recognition of objects in digital images.

The general form of a production is given by Equation (1), it consist of two parts: the *antecedents* (a set of preconditions) and the *consequents* (a set of actions). If a production rule's preconditions are fulfilled, then the rule is said to be *triggered*. If a rule's action is executed, it is said to have *fired*. A production system also contains a database, usually called *working memory*, which maintains data about the knowledge acquired or inferred by the system, and a rule interpreter. The rule interpreter must provide a mechanism for prioritizing productions when more than one is triggered.

$$< \text{antecedents} > \rightarrow < \text{consequents} > \quad (1)$$

Production rules can be used to represent both declarative and procedural knowledge. The characteristics of airplanes depicted in images are represented as a combination of attribute tables and production rules in SPAM (McKeown et al., 1985). In ERNEST (Niemann et al., 1990), the knowledge engineer (or knowledge designer) can define particular interpretation control strategies through production rules.

Among other image interpretation systems that use production rules either for description of image object patterns or for analysis control, the following can be mentioned: ACRONYM (Brooks, 1983), AIDA (Liedtke et al., 1997), BPI (Stilla and Michaelsen, 1997), ICARE (Desachy, 1991), MESSIE (Clement et al., 1993), MOSES (Quint, 1997), SIGMA (Matsuyama and Hwang, 1990) and ENVI’s Feature Extraction module (ITT, 2007).

It is important to observe that rule bases for image interpretation problems can contain large numbers of rules, especially considering the complexity and variability of the image objects present in remote sensing image data, and such large rule bases are hard to read and to maintain (Pahl, 2003, 2008).

PUC-Rio - Certificação Digital Nº 0510501/CA

2.1.5. Procedural interpretation

In general terms a *procedure* is a specified series of actions or operations, which have to be executed in the same manner in order to produce the same result, under the same circumstances. Procedures are, therefore, a form of procedural knowledge representation.

The whole image interpretation process can be described by procedures designed to find the objects of concern on a scene. Any computer program code can be understood as a set of procedures, however, the interest here in explicit knowledge representation.

The Definiens Developer software (Definiens, 2007) provides a high level visual language for defining procedures for image interpretation. The basic units of these procedures are the so-called *processes* that can be organized in sequence in a *process tree* (Figure 7).

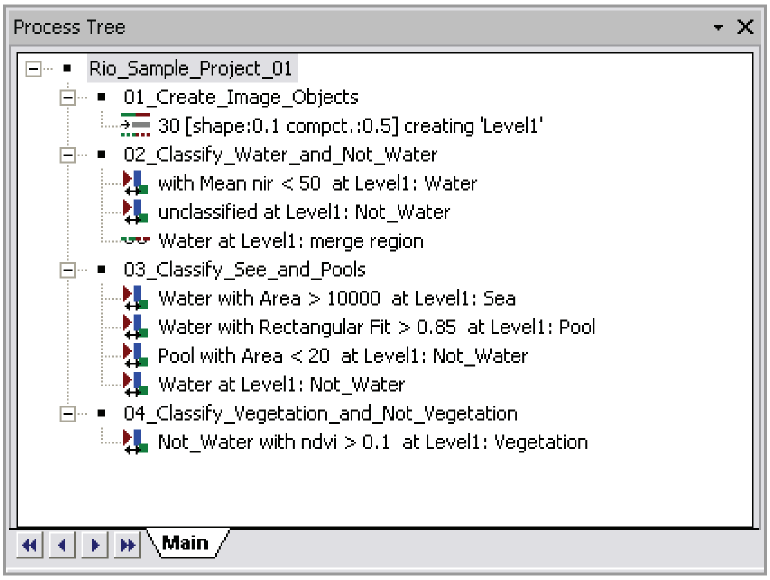

Figure 7. A process tree in the Definiens Developer software.

The basic functional parts of a single process are an *algorithm* and an *image object domain* (a set of object classes). In the system's terminology, an algorithm defines the operation a process will perform, and the operations provided by the system have two main functions: generating or modifying image objects; and classifying image objects (one of the algorithms available is the hierarchical classification mentioned in Section 2.1.2). The image object domain describes the region of interest where the algorithm will be executed in the image object hierarchy (see Section 2.1.2).

2.1.6. Multitemporal interpretation

Relatively few works can be found in the literature concerning knowledge-based image interpretation systems for the interpretation of multitemporal remote sensing data, the most important examples are the methodologies proposed in (Liedke and Growe, 2001) and (Pakzad, 2002). Both methods are based on the AIDA system (Liedke et al., 2001).

PUC-Rio - Certificação Digital Nº 0510501/CA

Liedke and Growe (2001) propose a multitemporal extension of AIDA, in which temporal knowledge is explicitly represented by temporal relations associated to links in the semantic network (Figure 8).

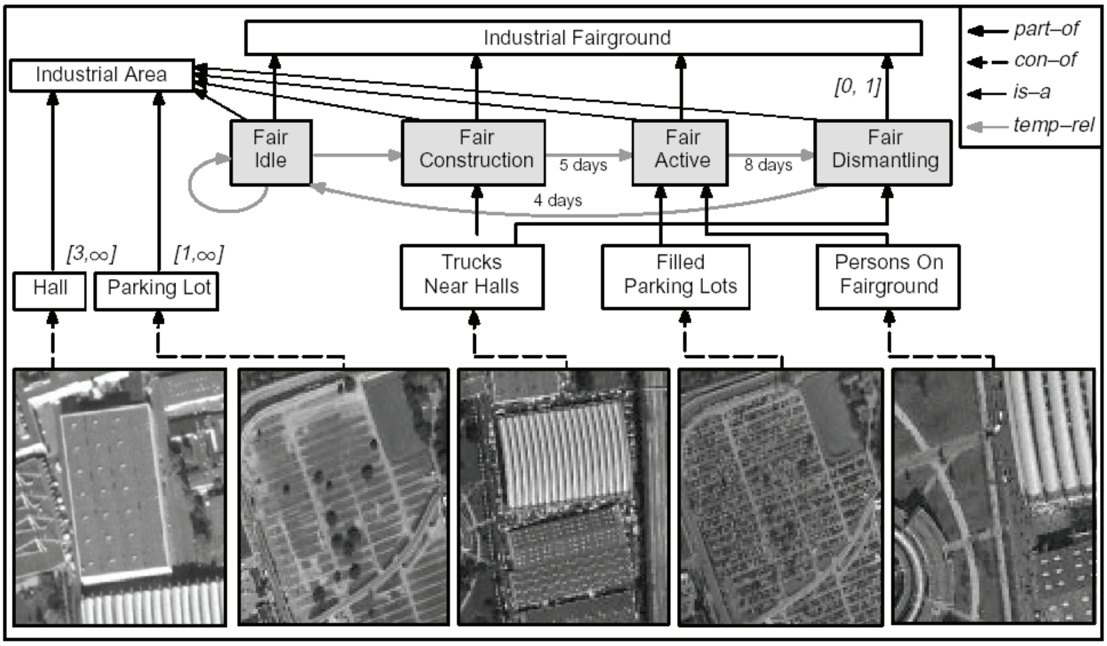

Figure 8. Semantic network with temporal relations in the AIDA system (Liedke and Growe, 2001).

Attached to each temporal relation (represented by the gray links in Figure 8) there is information about the respective transition probability – the probability of a geographical region to change from a particular class in one epoch (represented by the link's starting concept node) to another class in a subsequent epoch (represented by the link's ending node) – and about the temporal intervals in which the transitions are supposed to happen (transition times).

Analysis starts with the first image of the given sequence marked with time stamp t_1 (first epoch). If a concept subjected to temporal change (nodes with gray fill in Figure 8) can be instantiated, the temporal knowledge is used to create new hypotheses for the concepts connected by a temporal link for the next image in the chronological order (time stamp t_2). The system creates hypotheses for all connected concepts within the elapsed time $t_2 - t_1$, according to the transition time associated to the temporal relation. All hypotheses are treated as competing alternatives and stored in a search tree with a corresponding probability value.

PUC-Rio - Certificação Digital Nº 0510501/CA

Starting with the alternative with the highest probability, the hypotheses for the successor epoch are either verified or falsified.

Pakzad (2002) introduces a hybrid system, in which AIDA is responsible for the interpretation in individual epochs, and intermediate processing steps are responsible for generating classification hypotheses for geographical regions (or image segments) from one epoch to another. Temporal knowledge is represented through a class transition graph, as the one in Figure 9.

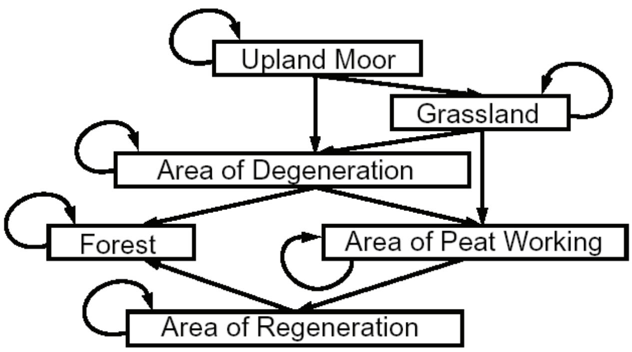

Figure 9. Class transition diagram in the method proposed in (Pakzad, 2002).

Each class in the transition graph correspond to a concept present in a semantic network in AIDA. Attached to the links in the class transition graph there are transition probabilities (defined by a human specialist), which are used in the process of predicting the classes of the image segments for the subsequent epoch, thus generating the initial hypotheses associated to the corresponding semantic network concept nodes. Before submitting the new hypotheses to AIDA, a special module may split the respective image segments in a resegmentation step taking into account the image data from the next epoch.

2.1.7. Control strategies

As well as there are different ways to organize and represent knowledge, there are different strategies for the processing of knowledge in a knowledge base, some which have been mentioned in the previous sections.

PUC-Rio - Certificação Digital Nº 0510501/CA

There are two main groups of control strategies: *hierarchical* and *heterarchical*, in basically four variants: *hierarchical top-down*, *bottom-up* and *hybrid control*; and *heterachical control* (Haralick and Shapiro, 1993).

Hierarchical control refers to a predefined (herarchical) ordering of the procedures that perform a specific task. Regarding image analysis, the most obvious sequence of events would be the extraction of features from the image, possibly after some pre-processing or preparation of the image; the construction of a symbolic description of the image objects, describing their respective attributes and interrelationships; and a decision making procedure for the determination of classes of the objects in the image, the conclusions of the automatic analysis.

Such a sequence: pre-processing; feature extraction; symbolic description; and decision making, constitute what is called as the bottom-up hierarchical control, a *data-driven* approach. An analogy between bottom-up hierarchical control and *forward chaining reasoning* can be made. Forward chaining reasoning start with the initial facts, and keep using rules to draw new conclusions (or take certain actions) given those facts. This is a common strategy in traditional remote sensed image classification software (including non knowledge-based ones). The hierarchical classification procedure in the Definiens Developer system (Definiens, 2007) is an example of bottom-up hierarchical control.

As bottom-up control can be associated to forward chaining, hierarchical top-down control strategy can be associated to backward chaining reasoning. It is a *model-driven* approach: it starts with the formulation of hypotheses about the occurrence of complex objects, which leads to new hypothesis about simpler objects and eventually of specific primitives.

Since neither strict bottom-up control, nor top-down control are flexible enough for the analysis of complex images, a hybrid form of control is more commonly used. This strategy usually starts with an initial segmentation of the image and identification of a preliminary set of image objects. After this first (bottom-up) step, depending on the objects found and their spatial arrangement, hypotheses of more complex objects are made, and a top-down run is executed to check for their pertinence. As more features, and then objects are recognized, more information can be deduced, helping to direct the recognition process. In GeoAIDA, as it will be described in more detail in Chapter 5, the interpretation process has an initial top-down step that is followed by a bottom-up step.

PUC-Rio - Certificação Digital Nº 0510501/CA

ERNEST (Niemann et al., 1990), offers the possibility for the knowledge engineer to define and implement its own strategy, using any particular combination of top-down and bottom-up control.

The control mechanisms used by most computer vision knowledge-based systems fit in this category. AIDA (Lietdke et al., 1997), GeoAIDA (Bükner et al., 2001), SCERPO (Lowe, 1987), SPAM (McKeown et al., 1985), and SIGMA (Matsuyama and Hwang, 1990) are some examples.

Heterarchical control differs from hierarchical in the sense that control is dictated not by a strict strategy, but by the data itself. This is the strategy implemented by *backboard systems*. Knowledge is segmented and packed inside *knowledge sources*.

Knowledge sources can be understood as information resources that also contain control information. They are supposed to work cooperatively, for example, finding two corners in an image could activate a knowledge source for the detection of buildings, finding a building would in turn trigger the activation of a knowledge source for finding streets. Knowledge sources access a global working memory – a blackboard – in which they state eventual conclusions and the intention of executing a specific procedure. As some sort of ordering is needed for this multiple activations, a scheduler controls the access of the blackboard and the execution of the pending procedures.

In the BPI system (Stilla and Michaelsen, 1997) a net of production rules is used for knowledge representation and a blackboard is used for process communication. Another blackboard-based architecture is proposed in (Mees and Perneel, 1998), sensor-independent knowledge is represented through fuzzy production rules, and sensor-dependent knowledge is encoded in image processing operators called *local detectors*.

2.2.
Multitemporal interpretation methods

Many different multitemporal approaches for of remote sensing image data analysis have been proposed, most of which concerned with change detection. Rather then providing a classification of the objects present in an image in terms of land cover/land use (LC/LU) classes, their main concern is to identify the

PUC-Rio - Certificação Digital Nº 0510501/CA

regions where change occurred, not necessarily attributing a LC/LU class to the region. A survey on the most relevant change detection techniques was presented in (Lu et al. 2004).

An interesting object-based (Blaschke and Strobl, 2001) change detection method, developed in the context of the DeCOVER Project (Büscher and Buck, 2007), is described in (Hofmann et al., 2008). The method compares two images from the same area taken at different points in time at the pixel level in order to create change objects within the boundaries of the LU/LC objects as defined in a GIS database. Change indicators are associated to each object and the most probable classes for a changed object at the later epoch are determined by taking into consideration the indicator values, the object's prior class assignment (given by the GIS database) and a manually defined class transition probability matrix. Information about the probable new class assignments can then be used to aid visual classification or to help select the automatic classification techniques to which the change objects will be later submitted.

Most multi-date image interpretation methods proposed so far can be regarded as *post-classification* approaches. These methods are based on separate single-date classifications whose results are subsequently compared (Weismiller et al., 1977). More powerful alternatives, called *cascade-classification* approaches (Swain, 1978), use all the information contained in the image sequence, trying to exploit the temporal correlation between images.

Various cascade classification techniques have been proposed, including Bayesian methods (Serpico and Melgani, 2000), neural networks (Bruzzone et al., 1999), as well as multi-classifier approaches (e.g. Bruzzone et al., 2004). In (Mota et al., 2007) a fuzzy multitemporal method is proposed for land-cover updating applications. The method relies on class transition possibilities that are estimated upon training data by a Genetic algorithm (GA) (Davis, 1990). The method is restricted to applications where the true class of the object being classified at an earlier time is known.

The following sections provide some notable examples of cascade multitemporal classification methods applied for remote sensing data.

PUC-Rio - Certificação Digital Nº 0510501/CA

2.2.1.
Cascade multitemporal approaches

Most of the cascade approaches found in the literature treat multitemporal classification interpretation as a data fusion problem, having evolved from earlier research on multisensor classification (Khazenie and Crawford, 1990; Jeon and Landgrebe, 1992; Aach and Kaup, 1995; Solberg et al., 1996; Bruzzone et al., 1999; Bruzzone and Cossu, 2002; Bruzzone et al., 2004). One of the early reports on such methods is found in (Jeon and Landgrebe, 1992), which proposes a contextual classifier that considers both spatial and temporal interpixel class dependencies, being the former modeled by class transition probabilities.

Another data fusion method that incorporates both the multisensory and temporal aspects of multidate image data was proposed in (Solberg et al., 1994; Jeon and Landgrebe, 1999). In this method, the a priori information on probabilities of class changes between image acquisition epochs is incorporated into a single-time model.

In (Tavakkoli-Sabour et al., 2008) a method for classifying agricultural crops in a series of SAR images through a statistical approach is proposed. The mean and standard deviation of pixel values covering the extent of a set of sample fields are used as signatures of each crop class for each image. The values in the distance vector for each field, comprising the distances computed for all epochs, are merged into an absolute distance value, and the field is assigned to the closest crop.

In (Bruzzone and Cossu, 2002) and (Bruzzone and Prieto, 2002), a cascade multitemporal method is proposed for land-cover map updating. Maximum likelihood classifiers and radial basis function neural networks compose the classification system. While the target application in both works is land-cover map updating, their central issue is how to explore data from an earlier image for training the classifiers when no ground truth information for the image being classified is available, the so called *partially unsupervised* training strategy.

Most cascade multitemporal approaches assume class conditional independence in the time domain (Bruzzone et al., 1999; Bruzzone and Prieto, 2002). Other works present methods that do not assume independence and try to

PUC-Rio - Certificação Digital Nº 0510501/CA

capture inter-source (spectral, spatial and temporal) correlations by means of neural networks (Melgani et al. 2001, 2003).

2.2.2. Semantic approaches

Another group of related approaches compares images of different dates at the semantic level; some of them have already been mentioned in Section 2.1.6. In these approaches different conditions for possible changes between objects from one date to another are described by means of class transition diagrams, which constitute the temporal part of prior knowledge (Pakzad, 2002; Pakzad et al., 2003).

Class transition diagrams can be used to identify possible class changes and to restrict the number of classes being considered for a given image segment (Bückner et al., 1999; Growe et al., 2000).

Pakzad (2001) and Growe (2001) associated each class transition to a value that expresses the probability that it might occur within a given time period. In those works, class transition probabilities merely establish the search order for a solution through a semantic network. Transition probabilities do not take part in the computation of any discriminant function and the class transition probabilities are defined empirically by specialists on the specific object classes and geographic area under analysis.

In (Leite et al., 2008) a method for crop class classification is proposed and tested over a set of LANDSAT images. In this method a hidden Markov model is generated for each of the crop classes considered. The individual models are built using expert knowledge about the respective crop's phenological cycle: each phenological stage is modeled as a hidden state, and the mean spectral values and the NDVI of the pixel values inside each image segment (at each of the 12 epochs considered) are considered as symbols emitted by the hidden states.

In (Mota et al., 2007), a possibilistic approach is introduced for modeling land cover class transitions. Class transition possibilities are estimated by means of a Genetic algorithm (Davis, 1990), whose objective function is the average class accuracy based on training samples. From an earlier crisp classification of the region of interest and the class transition possibility values, a fuzzy

PUC-Rio - Certificação Digital Nº 0510501/CA

classification is obtained and then fused with the output of the classification of the region in an image from a later date.

The need for a complete classification from a prior date is an important limitation of the approach proposed in (Mota et al., 2007), essentially because that information is often not available. The method described in the next chapter overcomes this limitation in such a way that the needed information about the prior date is simply an image from the same geographic area and some training samples for which the true classes are known at both dates (in the general case) or only at one date (in a particular case). Further innovations proposed here are a conceptual structure based on fuzzy Markov chains (Avrachenkov and Sanchez, 2002), and an analytical method for the estimation of transition possibilities.

PUC-Rio - Certificação Digital Nº 0510501/CA